Oracle® E-Business Suite

Integrated SOA Gateway Implementation Guide

Release 12.2

Part No. E20925-08

November 2013

ORACLE®

Oracle E-Business Suite Integrated SOA Gateway Implementation Guide, Release 12.2

Part No. E20925-08

Primary Author: Melody Yang

Contributor: Rekha Ayothi, Sudipto Chakraborty, Bhaskar Ghosh, Vardhan Kale, Jackie Lichtenstein, Rajeev Kumar, Megha Mathpal, Sai Munnalur, Aditya Rao, Anil Kemisetti, Nadakuditi Ravindra, Dilbaghsingh Sardar, Vijayakumar Shanmugam, Shivdas Tomar, Abhishek Verma, Sarah Zhu

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.

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.

This software or hardware and documentation may provide access to or information on 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. 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.

Contents

Send Us Your Comments

Preface

1 Oracle E-Business Suite Integrated SOA Gateway Overview

2 Setting Up Oracle E-Business Suite Integrated SOA Gateway

3 Administering Native Integration Interfaces and Services

4 Administering Composite Services - BPEL

5 Administering Custom Integration Interfaces and Services

6 Securing Web Services

7 Logging for Web Services

8 Monitoring and Managing SOAP Messages Using Service Monitor

9 Implementing Service Invocation Framework

A Oracle E-Business Suite Integrated SOA Gateway Diagnostic Tests

B Synchronous and Asynchronous Web Services

C Error Messages

Glossary

Index

Send Us Your Comments

Oracle E-Business Suite Integrated SOA Gateway Implementation Guide, Release 12.2

Part No. E20925-08

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Oracle E-Business Suite Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Intended Audience

Welcome to Release 12.2 of the *Oracle E-Business Suite Integrated SOA Gateway Implementation Guide.*

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.

- Computer desktop application usage and terminology.

- Oracle E-Business Suite integration interfaces.

- B2B, A2A and BP integrations.

This documentation assumes familiarity with Oracle E-Business Suite. It is written for the technical consultants, implementers and system integration consultants who oversee the functional requirements of these applications and deploy the functionality to their users.

If you have never used Oracle E-Business Suite, we suggest you attend one or more of the Oracle E-Business Suite training classes available through Oracle University.

See Related Information Sources on page x for more Oracle E-Business Suite product information.

Documentation Accessibility

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

Access to Oracle Support

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

Structure

Related Information Sources

This book is included in the Oracle E-Business Suite Documentation Library, which is supplied in the Release 12.2 Media Pack. If this guide refers you to other Oracle E-Business Suite documentation, use only the latest Release 12.2 versions of those guides.

Online Documentation

All Oracle E-Business Suite documentation is available online (HTML or PDF).

- **Online Help** - Online help patches (HTML) are available on My Oracle Support.

- **PDF Documentation** - See the Oracle E-Business Suite Documentation Library for current PDF documentation for your product with each release.

- **Release Notes** - For information about changes in this release, including new features, known issues, and other details, see the release notes for the relevant product, available on My Oracle Support.

- **Oracle Electronic Technical Reference Manual -** The Oracle Electronic Technical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for each Oracle E-Business Suite product. This information helps you convert data from your existing applications and integrate Oracle E-Business Suite data with non-Oracle applications, and write custom reports for Oracle E-Business Suite products. The Oracle eTRM is available on My Oracle Support.

Related Guides

You should have the following related books on hand. Depending on the requirements of your particular installation, you may also need additional manuals or guides.

Oracle E-Business Suite Concepts

This book is intended for all those planning to deploy Oracle E-Business Suite Release 12.2, or contemplating significant changes to a configuration. After describing the Oracle E-Business Suite architecture and technology stack, it focuses on strategic topics, giving a broad outline of the actions needed to achieve a particular goal, plus the installation and configuration choices that may be available.

Oracle Application Framework Personalization Guide

This guide covers the design-time and run-time aspects of personalizing applications built with Oracle Application Framework.

Oracle E-Business Suite Installation Guide: Using Rapid Install

This book is intended for use by anyone who is responsible for installing or upgrading Oracle E-Business Suite. It provides instructions for running Rapid Install either to carry out a fresh installation of Oracle E-Business Suite Release 12.2, or as part of an upgrade to Release 12.2.

Oracle E-Business Suite Maintenance Guide

This guide explains how to patch an Oracle E-Business Suite system, describing the adop patching utility and providing guidelines and tips for performing typical patching operations. It also describes maintenance strategies and tools that can help keep a system running smoothly.

Oracle E-Business Suite Security Guide

This guide contains information on a comprehensive range of security-related topics, including access control, user management, function security, data security, and auditing. It also describes how Oracle E-Business Suite can be integrated into a single sign-on environment.

Oracle E-Business Suite Setup Guide

This guide contains information on system configuration tasks that are carried out either after installation or whenever there is a significant change to the system. The activities described include defining concurrent programs and managers, enabling Oracle Applications Manager features, and setting up printers and online help.

Oracle Fusion Middleware Adapter for Oracle Applications User's Guide

This guide covers the use of Adapter for Oracle Applications in developing integrations between Oracle E-Business Suite and trading partners.

This book is available in the Oracle Fusion Middleware 11*g* Documentation Library.

Oracle Fusion Middleware Introduction to Oracle WebLogic Server

This book provides an overview of Oracle WebLogic Server features and describes how you can use them to create enterprise-ready solutions. This book is available in the Oracle Fusion Middleware 11*g* Documentation Library.

Oracle E-Business Suite User's Guide

This guide explains how to navigate, enter and query data, and run concurrent requests using the user interface (UI) of Oracle E-Business Suite. This guide also includes information on setting user profiles and customizing the UI.

Oracle Diagnostics Framework User's Guide

This manual contains information on implementing and administering diagnostics tests for Oracle E-Business Suite using the Oracle Diagnostics Framework.

Oracle E-Business Suite Integrated SOA Gateway User's Guide

This guide describes the high level service enablement process, explaining how users can browse and view the integration interface definitions and services residing in Oracle Integration Repository.

Oracle E-Business Suite Integrated SOA Gateway Developer's Guide

This guide describes how system integration developers can perform end-to-end service integration activities. These include orchestrating discrete Web services into meaningful end-to-end business processes using business process execution language (BPEL), and deploying BPEL processes at run time.

This guide also explains how to invoke Web services using the Service Invocation Framework. This includes defining Web service invocation metadata, invoking Web services, and testing the Web service invocation.

Oracle e-Commerce Gateway Implementation Guide

This guide describes implementation details, highlighting additional setup steps needed for trading partners, code conversion, and Oracle E-Business Suite. It also provides architecture guidelines for transaction interface files, troubleshooting information, and a description of how to customize EDI transactions.

Oracle iSetup User's Guide

This guide describes how to use Oracle iSetup to migrate data between different instances of the Oracle E-Business Suite and generate reports. It also includes configuration information, instance mapping, and seeded templates used for data migration.

Oracle Workflow Administrator's Guide

This guide explains how to complete the setup steps necessary for any product that includes workflow-enabled processes. It also describes how to manage workflow processes and business events using Oracle Applications Manager, how to monitor the progress of runtime workflow processes, and how to administer notifications sent to workflow users.

Oracle Workflow User's Guide

This guide describes how users can view and respond to workflow notifications and monitor the progress of their workflow processes.

Oracle Workflow API Reference

This guide describes the APIs provided for developers and administrators to access Oracle Workflow.

Oracle XML Gateway User's Guide

This guide describes Oracle XML Gateway functionality and each component of the Oracle XML Gateway architecture, including Message Designer, Oracle XML Gateway Setup, Execution Engine, Message Queues, and Oracle Transport Agent. It also explains how to use Collaboration History that records all business transactions and messages exchanged with trading partners.

The integrations with Oracle Workflow Business Event System, and the Business-to-Business transactions are also addressed in this guide.

Integration Repository

The Oracle Integration Repository is a compilation of information about the service endpoints exposed by the Oracle E-Business Suite of applications. It provides a complete catalog of Oracle E-Business Suite's business service interfaces. The tool lets users easily discover and deploy the appropriate business service interface for integration with any system, application, or business partner.

The Oracle Integration Repository is shipped as part of the E-Business Suite. As your instance is patched, the repository is automatically updated with content appropriate for the precise revisions of interfaces in your environment.

You can navigate to the Oracle Integration Repository through Oracle E-Business Suite Integrated SOA Gateway.

Do Not Use Database Tools to Modify Oracle E-Business Suite Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle E-Business Suite data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle E-Business Suite data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle E-Business Suite tables are interrelated, any change you make using an Oracle E-Business Suite form can update many tables at once. But when you modify Oracle E-Business Suite data using anything other than Oracle E-Business Suite, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle E-Business Suite.

When you use Oracle E-Business Suite to modify your data, Oracle E-Business Suite automatically checks that your changes are valid. Oracle E-Business Suite also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

1

Oracle E-Business Suite Integrated SOA Gateway Overview

Oracle E-Business Suite Integrated SOA Gateway Overview

Oracle E-Business Suite Integrated SOA Gateway (ISG) is enhanced to leverage Oracle SOA Suite 11*g* running on Oracle WebLogic Server to provide greater capabilities and infrastructure for exposing various integration interfaces within Oracle E-Business Suite as Web services.

With service enablement feature, integration interfaces published in the Oracle Integration Repository can be transformed into Web services with the supported type SOAP and REST.

SOAP services use Oracle SOA Suite. Once the services are generated with WSDL descriptions, they are deployed to Oracle SOA Suite for service consumption.

Unlike SOAP services, REST services, without the dependency on Oracle SOA Suite, are developed with the infrastructure of Oracle E-Business Suite. REST services described in WADL URLs are directly deployed to an Oracle E-Business Suite WebLogic environment. They can be used for user-driven applications such as Oracle E-Business Suite mobile applications.

> **Note:** All service-enabled interfaces can be generated as standard SOAP services. However, only PL/SQL APIs can be exposed as REST services in this release.

- At development phase, users with the System Integration Developer role can create custom interfaces, and annotate custom interface's definitions. Users with the Integration Repository Administrator role can validate and upload annotated custom interfaces to the Integration Repository where all the registered interfaces, regardless of custom or Oracle packaged ones, can be viewed and accessed by all users.

- At design time, users with the Integration Repository Administrator role can generate SOAP services with desired operation patterns, and deploy them to Oracle SOA Suite by attaching an appropriate security policy. For interfaces can be exposed as REST services, the administrator can select desired service operations before deploying them to Oracle E-Business Suite.

- At run time, Web service clients send request messages to invoke Oracle E-Business Suite services enabled through ISG's Service Provider. After authenticating and authorizing the users who request the services, services can be invoked.

  Users with the Integration Repository Administrator role are responsible for monitoring and managing the entire service deployment life cycle.

Major Features

Oracle E-Business Suite Integrated SOA Gateway can do the following:

- Display all Oracle E-Business Suite integration interface definitions through Oracle Integration Repository

- Support custom integration interfaces from Oracle Integration Repository

- Provide service enablement capability (SOAP and REST services) for seeded and custom integration interfaces within Oracle E-Business Suite

- Use the Integration Repository user interface to perform design-time activities such as generate and deploy Oracle E-Business Suite Web services

- Support synchronous and asynchronous (callback without acknowledgement only) interaction patterns for SOAP-based Web services

  > **Note:** In this release, only PL/SQL APIs can be enabled with the support for asynchronous service pattern.

- Support synchronous interaction pattern for REST-based Web services

  > **Note:** In this release, only PL/SQL APIs can be exposed as REST services.

- Support multiple authentication types for inbound service requests in securing Web service content

- Enforce function security and role-based access control security to allow only authorized users to execute administrative functions

- Provide centralized, user-friendly logging configuration for Web services generated through Oracle E-Business Suite Integrated SOA Gateway's service provider

- Audit and monitor Oracle E-Business Suite inbound service operations from Service Monitor

- Leverage Oracle Workflow Business Event System to enable Web service invocation from Oracle E-Business Suite

Major Components Features and Definitions

Oracle E-Business Suite Integrated SOA Gateway provides two major service offerings:

- Providing Services

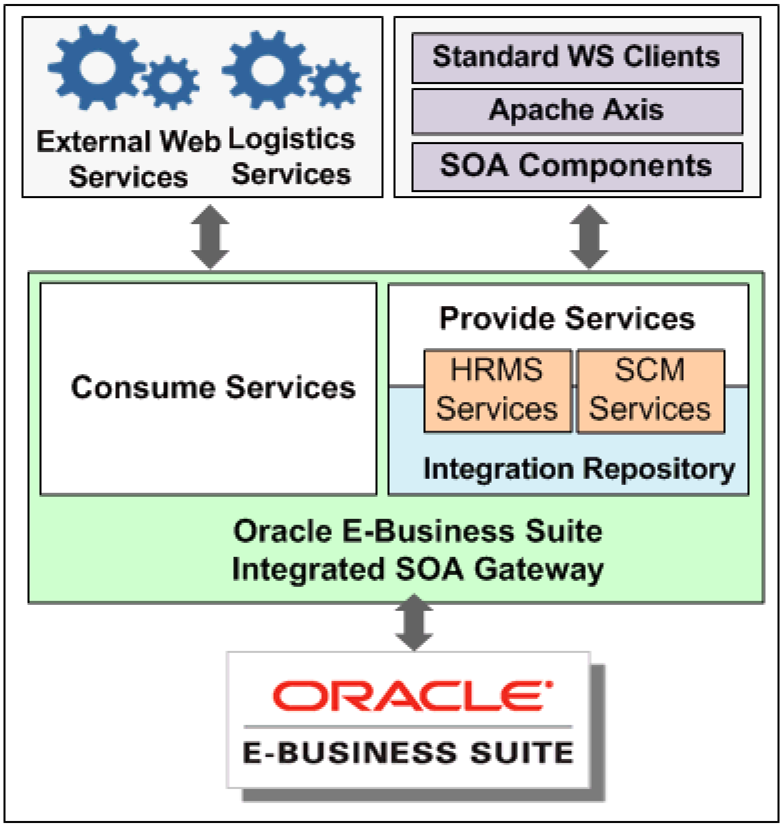
  Oracle E-Business Suite interfaces resided in Oracle Integration Repository can be service enabled through service provider. The service enablement is the key feature within the Oracle E-Business Suite Integrated SOA Gateway.

  Once services are deployed, Web service clients send request messages and invoke Oracle E-Business Suite services. All SOAP requests and responses are monitored and audited through Service Monitor.

- Consuming Services

  In addition to providing services, Oracle E-Business Suite Integrated SOA Gateway can consume external Web services through Service Invocation Framework.

Providing and Consuming Web Services

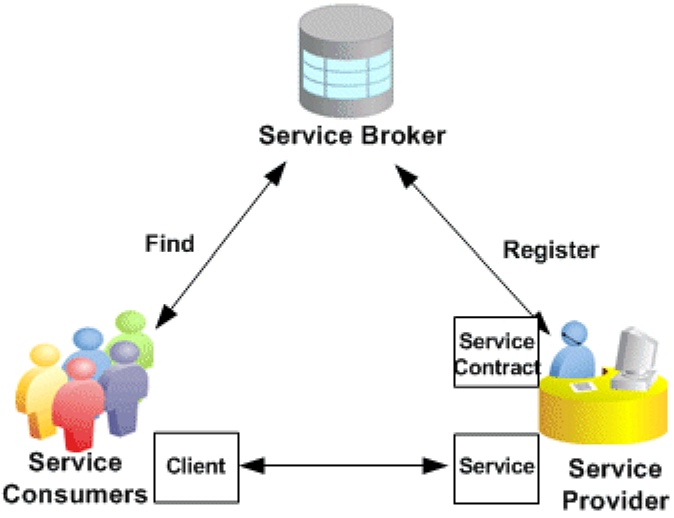


To better understand Oracle E-Business Suite Integrated SOA Gateway, the next sections explain essential components and how each component is used.

Enabling Oracle E-Business Suite Web Services

Service enablement is the key feature within Oracle E-Business Suite Integrated SOA Gateway. It provides a mechanism that allows native packaged integration interface definitions resided in Oracle Integration Repository to be transformed into Web services. SOAP services are deployed from the Integration Repository to Oracle SOA Suite allowing more consumptions over the Web. REST services are deployed to Oracle E-Business Suite.

The basic concept of Web service components is illustrated in the following diagram:



- Service Provider is the primary engine underlying the Web services. It acts as a bridge between Oracle E-Business Suite and Oracle SOA Suite to facilitate the service enablement for various types of Oracle E-Business Suite interfaces.

  Note: In earlier Oracle E-Business Suite Releases, SOA Provider and Web Service Provider were used in enabling Oracle E-Business Suite Web services. In the Release 12.2, Service Provider is the engine for service enablement.

  Please note that Service Provider leverages Oracle SOA Suite for provisioning Oracle E-Business Suite SOAP-based services. It is the engine that performs the actual service generation and deployment behind the scene.

- Service Consumer (Web service client) is the party that uses or consumes the services provided by the Service Provider.

- Service Broker (Service Registry) describes the service's location and contract to ensure service information is available to potential service consumers.

Oracle Integration Repository and Service Enablement

Oracle Integration Repository, an integral part of Oracle E-Business Suite, is the centralized repository that contains numerous interface endpoints exposed by applications within the Oracle E-Business Suite. It supports the following interface types:

- PL/SQL

- XML Gateway

- Concurrent Programs
- Business Events
- Interface Tables/Views
- EDI
- Business Service Object (Service Beans)
- Java
  - Java APIs for Forms

    > **Note:** Java APIs for Forms are XML document-based integration points wrapped in Java classes for executing business logic in Oracle Forms. These specialized Java classes are categorized as a subtype of Java interface.

  - Security Services

    > **Note:** Security Services are a set of predefined and predeployed REST services from Oracle Application Object Library. This type of services provides Authentication and Authorization services for mobile applications. These services are built on Java; therefore, they are categorized as a subtype of Java interface.

- Composite Interfaces

Oracle E-Business Suite Integrated SOA Gateway leverages Oracle Integration Repository to provide the capabilities of Web service generation and deployment, as well as service life cycle management.

> **Note:** Please note that not all the interface types resided in the Integration Repository can be service enabled. The supported interface types for service enablement are XML Gateway, PL/SQL, Current Program, Business Events, Business Service Object, and Java API for Forms.
>
> As mentioned earlier, security services are pregenerated REST services from Oracle Application Object Library. Therefore, there is no need to enable the security services from the repository as required by other supported interface types.

Web Service Security

To protect application data from unauthorized access, Oracle E-Business Suite integrated SOA Gateway enforces the security rules through subject authentication and authorization:

- To authenticate users who request Oracle E-Business Suite Web services, request messages must be checked based on the selected authentication type:
  - The SOAP messages must be authenticated using UsernameToken or SAML Token based security. The identified authentication information is embedded in the `wsse:security` Web Security headers.
  - The REST messages are authenticated using HTTP Basic Authentication security (either username/password or security token) at HTTP transport level.
- To authorize users on specific services or operations, the access permissions must be explicitly given to the users through security grants. Multiple organization access control (MOAC) security rule is also implemented for authorizing interface execution related to multiple organizations.

Additionally, input message header (such as SOAHeader and RESTHeader) is used to pass application contexts needed in invoking Oracle E-Business Suite services as part of the subject authorization.

Service Monitor

Service Monitor known as SOA Monitor in earlier releases is a centralized, light-weight service execution monitoring and management tool.

It fetches data and statistics for each instance of a Web service request and response from the underlying Oracle SOA Suite infrastructure to let you monitor Oracle E-Business Suite Web services. You can use the Service Monitor user interface in Oracle E-Business Suite to view the runtime request and response data received and sent from Oracle SOA Suite.

Please note that only SOAP services are monitored and audited through Service Monitor. Runtime REST service monitoring and auditing features are not supported in this release.

Service Invocation Framework

Service Invocation Framework (SIF) leverages Oracle Workflow Java Business Event System (JBES) and a seeded Java rule function to invoke services within Oracle E-Business Suite.

It provides an infrastructure allowing developers to interact with Web services through WSDL descriptions. For detailed implementation information, see Implementing Service Invocation Framework, page 9-1.

Native Service Enablement Architecture Overview

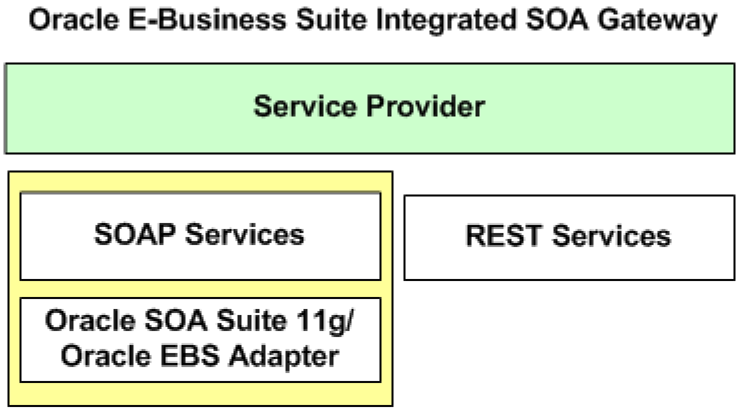
Oracle E-Business Suite Integrated SOA Gateway employs essential components that enable service integration at design time and run time, and ease the service management throughout the entire service deployment life cycle.

Service Provider is the primary engine enabling the Oracle E-Business Suite services. It is the engine that performs the actual service generation and deployment behind the scene for both SOAP and REST services.

- In SOAP-based service enablement, it leverages Oracle SOA Suite 11*g* and Oracle Applications Adapter (also called Oracle E-Business Suite Adapter) for provisioning standard Web services for business integration.

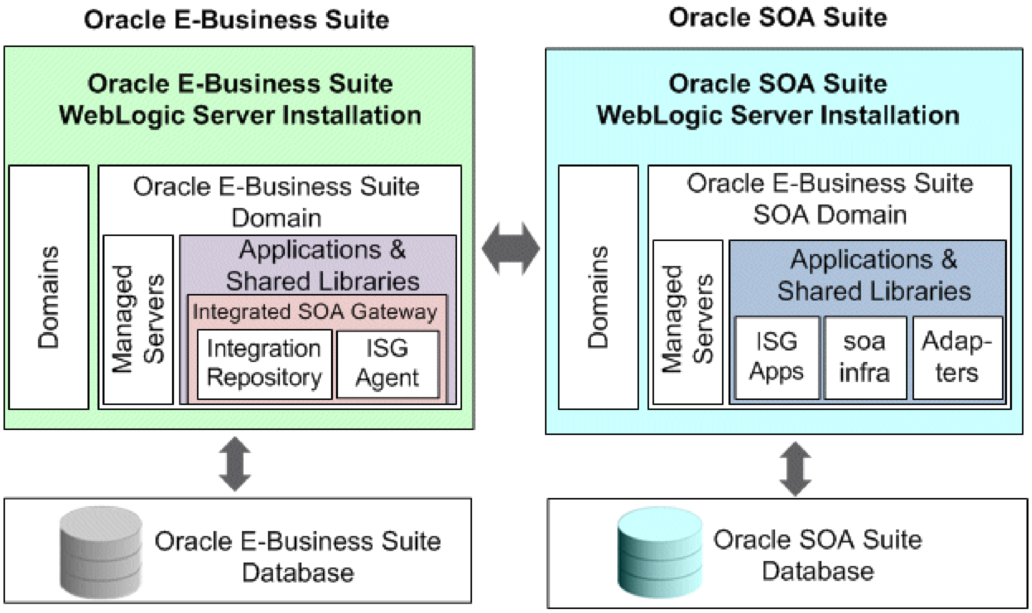
- In REST-based service enablement, it provides light weight, out-of-box services for mobile applications and chatty UI applications.

The high level service enablement diagram can be illustrated in the following diagram:



SOAP Service Enablement Architecture and Design Time

SOAP services, once successfully generated, are deployed to an Oracle SOA Suite WebLogic environment. The seamless integration between Oracle E-Business Suite and Oracle SOA Suite forms the Oracle E-Business Suite Integrated SOA Gateway architecture.


Oracle E-Business Suite
Oracle SOA Suite
Oracle E-Business Suite WebLogic Server Installation
Domains
Oracle E-Business Suite Domain
Managed Servers
Applications & Shared Libraries
Integrated SOA Gateway
Integration Repository
ISG Agent
Oracle SOA Suite WebLogic Server Installation
Domains
Oracle E-Business Suite SOA Domain
Managed Servers
Applications & Shared Libraries
ISG Apps
soa infra
Adap-ters
Oracle E-Business Suite Database
Oracle SOA Suite Database

- **Oracle E-Business Suite on Oracle WebLogic Server**

  Oracle E-Business Suite is integrated with Oracle WebLogic Server (WLS) to provide a complete set of service infrastructure with great flexibility.

  Oracle WebLogic Server is an application server that provides an implementation of Java Platform Enterprise Edition (Java EE, formerly known as J2EE) specification. Its infrastructure enables enterprises to deploy mission-critical applications in a robust, secure, and highly scalable environment and is an ideal foundation for building applications based on service-oriented architecture.

  A WebLogic server can include many domains. A domain is an administrative unit or boundary that provides for a single point of administration for a collection of servers. Therefore, a single domain comprises one administration server and one or more managed servers.

  For more information on Oracle WebLogic Server features and system administration, see the *Oracle Fusion Middleware Introduction to Oracle WebLogic Server*.

- **Oracle SOA Suite on Oracle WebLogic Server**

  Oracle SOA Suite is an essential middleware layer of Oracle Fusion Middleware. It contains full range of service components for designing, deploying, and managing composite applications. Furthermore, Oracle SOA Suite provides various integrated capabilities, such as messaging, orchestration, Web services management, business monitoring, and so on. These capabilities facilitate the service integration between various enterprises in different platforms.

  With seamless integration with Oracle SOA Suite, Oracle E-Business Suite Integrated SOA Gateway becomes a self-contained Web application. Oracle

E-Business Suite integration interfaces can be exposed as Web services through SOA Composites in Oracle SOA Suite.

For more information on Oracle SOA Suite 11*g*, see the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*, and the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

At design time, a system integration developer or an integration repository administrator can select a desired interface and perform the service generation from the repository.

Once the service artifact has been generated, an integration repository administrator can deploy the service from Oracle Integration Repository to an Oracle SOA Suite WebLogic environment where the `soa-infra` application is running.

> **Note:** Users with different roles can perform various tasks in Oracle E-Business Suite Integrated SOA Gateway. Each user role representing a unique permission or permission set can be granted to appropriate users. For example, an integration repository administrator defined by the Integration Repository Administrator role can perform design-time operations, and other administrative tasks. For information on user roles and how to grant roles to users, see Assigning User Roles, page 2-2 and Role-Based Access Control (RBAC) Security for Oracle E-Business Suite Integrated SOA Gateway, page 6-4.

REST Service Design Time

Without the dependency on Oracle SOA Suite, REST services are developed based on Oracle E-Business Suite technology infrastructure.

At design time, a system integration developer or an integration repository administrator can select desired methods to be exposed as REST service operations before deploying them to Oracle E-Business Suite.

Additionally, the administrator can undeploy the service if needed.

Service Enablement Run Time

Oracle E-Business Suite services can be exposed as Web services and are interacted with Web service clients at run time.

When service consumers or Web service clients send request messages at run time, before invoking deployed services in the managed servers, all service-related security and policies are enforced.

After authenticating the requests, Oracle E-Business Suite Web services can be invoked. service response messages will be sent back to Web service clients if needed.

For each service operation, SOAP request and response messages passed through

Oracle SOA Suite will be captured in Service Monitor where all Oracle E-Business Suite Web service activities executed at run time can be monitored.

> **Note:** REST service monitoring and auditing features are not supported in this release.

For more information on how to monitor SOAP messages in Service Monitor, see Monitoring and Managing SOAP Messages Using Service Monitor, page 8-1.

Web Service Clients

Customers or third parties can use the following standard Web service client technologies or tools to invoke Oracle E-Business Suite Web services:

- Apache Axis

  Apache Axis is an open source, XML based Web service framework for constructing SOAP processors such as clients, servers, gateways, etc. It consists of a Java and a C++ implementation of the SOAP server, and various utilities and APIs for generating and deploying Web service applications. It can help create, publish, and consume Web services.

- .NET Web Service Client

  .NET Web service client enables you to create Web services and call these services from any client application.

- Oracle JDeveloper

  Oracle JDeveloper is used to help create Web service clients through Java SOAP APIs.

- Oracle BPEL Process Manager

  Business process execution language (BPEL) is particularly used in orchestrating complex business processes in a SOA composite application.

- Oracle Service Bus (OSB)

  Oracle Service Bus provides enterprise service level mediation. It can be used for simple transactional service to transport and route messages between service consumers and service providers.

2

Setting Up Oracle E-Business Suite Integrated SOA Gateway

Setup Overview

Oracle E-Business Suite Integrated SOA Gateway can be set up either on an existing installation of Oracle WebLogic Server or on a newly installed Oracle WebLogic Server. Before the installation, you must first understand the product dependencies.

Product Dependencies

Oracle E-Business Suite Integrated SOA Gateway depends on the following products to provide its functionality:

- **Oracle SOA Suite 11*g* running on Oracle WebLogic Server**

  In this release, Oracle E-Business Suite Integrated SOA Gateway leverages the features of Oracle SOA Suite 11*g* to expose public interfaces in Oracle E-Business Suite as Web services.

  Service Provider, one of the essential components in Oracle E-Business Suite Integrated SOA Gateway, uses Oracle SOA Suite for provisioning SOAP requests for Oracle E-Business Suite Web services. It generates the SOA Composites which are deployed on Oracle SOA Suite server.

- **Oracle Applications Adapter** (also called Oracle E-Business Suite Adapter)

  Oracle E-Business Suite Adapter provided from Oracle SOA Suite is part of the Oracle Fusion Middleware components. Oracle E-Business Suite Integrated SOA Gateway leverages its features for PL/SQL, Concurrent Program, and XML Gateway based Oracle E-Business Suite Web services. The invocation of the Web service is handled by Oracle SOA Suite after the parameters in the inbound SOAP headers are validated by Oracle E-Business Suite Adapter.

For information on how to install Oracle SOA Suite 11*g*, see *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

For information on how to configure, troubleshoot, or upgrade Oracle E-Business Suite Integrated SOA Gateway from earlier releases, refer to the following documents:

- For information on how to install or upgrade Oracle E-Business Suite Integrated SOA Gateway from earlier releases, and how to perform setup tasks, see *Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12.2*, My Oracle Support Knowledge Document 1311068.1.

- For troubleshooting information on potential problem symptoms and corresponding solutions for Oracle E-Business Suite Integrated SOA Gateway, see *Oracle E-Business Suite Integrated SOA Gateway Troubleshooting Guide, Release 12.2*, My Oracle Support Knowledge Document 1317697.1 for details.

- If you are planning to use the Java APIs for Forms interfaces published in Oracle Integration Repository that encapsulate Oracle Forms logic, see *Oracle E-Business Suite Java APIs for Forms Troubleshooting Guide, Release 12.2*, My Oracle Support Knowledge Document 1469785.1 for troubleshooting information related to Oracle Supply Chain Management Web services.

After configuring Oracle E-Business Suite Integrated SOA Gateway, administrators should set the required profile options and assign appropriate roles to users which allow them to perform design-time operations, monitor the Web services and view logs. The next sections on assigning roles and setting profile options explain these features.

Assigning User Roles

Oracle E-Business Suite Integrated SOA Gateway uses the following user roles to perform needed administrative and user tasks. A system administrator can assign these user roles to appropriate users if necessary.

- Integration Repository Administrator role (UMX|FND_IREP_ADMIN)

- System Integration Analyst role (UMX|FND_SYSTEM_INTEGRATION_ANALYST)

- System Integration Developer role (UMX|FND_SYSTEM_INTEGRATION_DEVELOPER)

For example, users with the System Integration Analyst role can browse integration interfaces and services through the Integration Repository user interface as well as view each interface details. Users with the System Integration Developer role not only can view each interface through the repository, but also annotate custom integration interfaces based on annotation standards, and perform service generation task. Users with the Integration Repository Administrator role can perform all user and administrative tasks including browsing and viewing each integration interface and service, generating, deploying, and undeploying services, as well as retiring active services, activating retired services, and resetting services.

To assign a user role:

1. Log in to Oracle E-Business Suite with an administrator role and choose the User Management responsibility.

2. Select the Users link from the navigation menu.

3. Enter appropriate information in the search area to locate a desired user account. Click **Go**.

4. Click the **Update** icon next to the user with 'Active' account status to open the Update User window.

5. Click **Assign Roles**.

6. In the search window, search for either one of the following user roles:

   - Integration Repository Administrator

   - System Integration Analyst

   - System Integration Developer

   Choose a desired role and click **Select**.

7. Enter a justification in the Justification filed and click **Apply**.

   You will see a confirmation message indicating you have successfully assigned the role.

For more information on assigning or revoking user roles, see *Oracle E-Business Suite Security Guide*.

Setting Profile Options

The following table lists the profile options used in Oracle E-Business Suite Integrated SOA Gateway:

Profile Option	Description	Required	Default Value
FND: XML Gateway Map Generic Service	Use this profile option to display or hide the generic XML Gateway service information for the selected XML Gateway map. • If it is set to 'Yes', the Generic XML Gateway Service subregion is displayed within the Web Service region in the XML Gateway Map interface details page. • If it is set to 'No', the Generic XML Gateway Service subregion will not be displayed in the XML Gateway Map interface details page.	Yes	Yes **Important:** If you do not start from this release and you have been using generic XML Gateway Web service, set the profile option to 'Yes'. This allows the Generic XML Gateway Services subregion to be displayed within the Web Service region. Otherwise, subregion will not be shown and any invocations of generic XML Gateway Web services will return a fault message.

Profile Option	Description	Required	Default Value
ISG: Generic Service WSDL URL for XMLG	Once a generic XML Gateway Web service has been deployed, the deployed service WSDL URL is populated as the profile value and the URL is also displayed in the 'Generic XML Gateway Service' subregion. If the generic service is not deployed, the profile value will not be shown and hence no WSDL URL is displayed in the subregion for the selected XML Gateway interface.	Yes	N/A

Use the *FND: XML Gateway Map Generic Service* profile option to display generic XML Gateway service information contained in the subregion only if your system is upgraded from a previous release and you have been using generic XML Gateway Web services.

For information on setting profile options, see User Profiles and Profile Options in Oracle Application Object Library, *Oracle E-Business Suite Setup Guide*.

3

Administering Native Integration Interfaces and Services

Overview

Various Oracle E-Business Suite application interface definitions shipped with Oracle Integration repository are referred as native integration interfaces. This chapter describes the steps to transform these interface definitions into SOAP Web services or REST Web services.

An Oracle E-Business Suite user who has the Integration Repository Administrator role, hereafter referred as an integration repository administrator or the administrator, can manage each state of the services throughout the service life cycle as well as manage grants for them.

To better understand how to administer and manage these two types of Web services, the following topics are included in this chapter:

- Administering SOAP Web Services, page 3-1
- Administering REST Web Services, page 3-33

Administering SOAP Web Services

Interfaces Supported for SOAP Service Enablement

Oracle E-Business Suite Integrated SOA Gateway supports the following interface types for SOAP-based service enablement:

- PL/SQL
- XML Gateway Map (Inbound)
- Concurrent Program

Important: Concurrent programs that are linked to Open Interfaces can be viewed and displayed under the Open Interface category which Oracle Integration Repository does not support for service enablement.

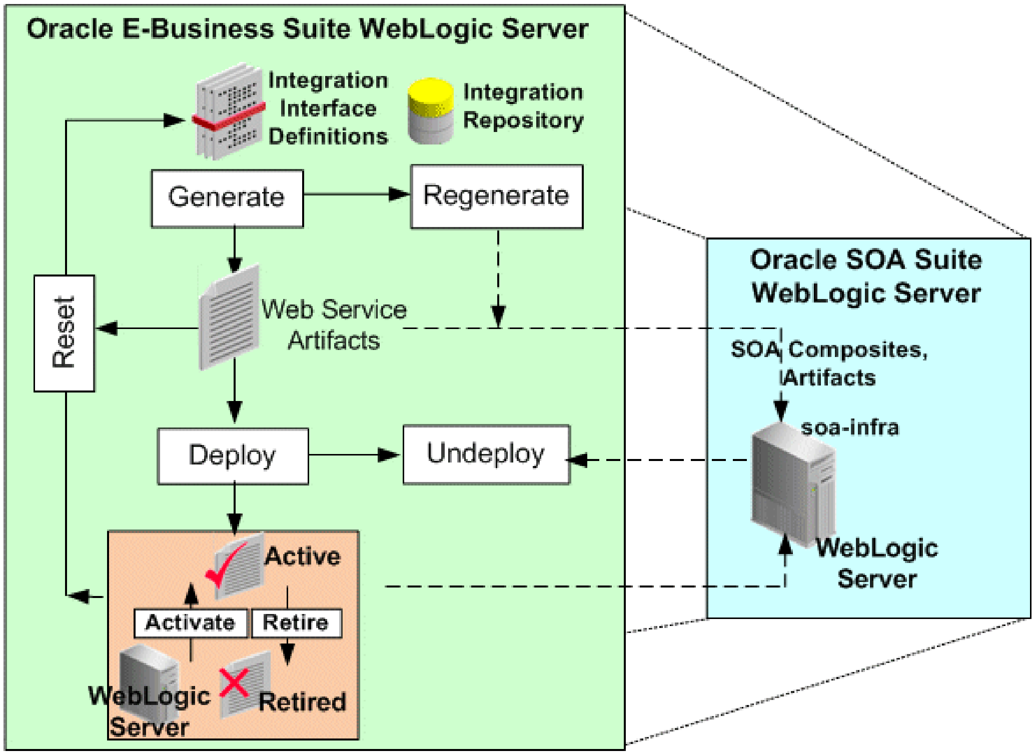
- Business Service Object
- Java APIs for Forms

  Note: Java APIs for Forms are XML document-based integration points wrapped in Java classes for executing business logic in Oracle Forms. These specialized Java classes are categorized as a subtype of Java interface.

Managing SOAP Service Life Cycle Activities

The integration repository administrator can perform the following administrative tasks in managing each state of SOAP services throughout the entire service life cycle:

Service Generation and Deployment Process Flow



- Generating SOAP Web Services, page 3-4
- Deploying and Undeploying SOAP Web Services, page 3-10
- Resetting SOAP Web Services, page 3-15
- Retiring SOAP Web Services, page 3-17
- Activating SOAP Web Services, page 3-18
- Subscribing to Business Events, page 3-20
- Viewing and Enabling Design-Time Log Configuration, page 3-24
- Viewing Generate and Deploy Time Logs for SOAP Services, page 3-24
- Managing SOAP Service Life Cycle Activities Using An Ant Script, page 3-27

Managing Other Administrative Tasks for SOAP Services

Some administrative tasks are performed outside the Integration Repository user interface. These tasks are performed in the **Administration** tab including configuring log setups and monitoring runtime SOAP activities. See:

- Logging for Web Services, page 7-1
- Monitoring and Managing SOAP Messages Using Service Monitor, page 8-1

Generating SOAP Web Services

Oracle E-Business Suite Integrated SOA Gateway allows users with the Integration Repository Administrator role or the System Integration Developer role (UMX|FND_SYSTEM_INTEGRATION_DEVELOPER) to transform interface definitions to SOAP services.

SOAP services can be generated with the support for synchronous or asynchronous interaction pattern, or both synchronous and asynchronous patterns. Before generating a service, the administrator or a system integration developer must specify interaction pattern(s) for desired methods to be exposed as service operations. This can be achieved at the method level for one or more methods, or at the interface level for all methods.

> **Important:** In this release, asynchronous operation is supported only in PL/SQL interfaces in enabling SOAP-based services.
>
> - For XML Gateway and Concurrent Program interface types
>
>   Each interface contains only one method and it can only be service enabled synchronously by default; therefore, the Interaction Pattern table will not be displayed in the Web Service region.
>
> - For Business Service Object and Java APIs for Forms interface types
>
>   Each interface may contain more than one method; therefore, only the Synchronous column is displayed in the Interaction Pattern table for method selection.

Please note that by default, none of the interaction pattern would be selected. However, if your system is upgraded from a previous release, for backward compatibility, 'synchronous' pattern is selected for all the methods contained in a service.

For more information about synchronous and asynchronous operation patterns, see Synchronous and Asynchronous Web Services, page B-1.

Generating Services

For interfaces with the support for SOAP services only, service activities are managed in the Web Service region. For interfaces with the support for both REST and SOAP services, these activities are managed in the SOAP Web Service tab of the interface details page.

Once a service is generated, the associated service artifacts are also generated for the selected methods. If only one method is selected, then only that selected method has a service artifact generated.

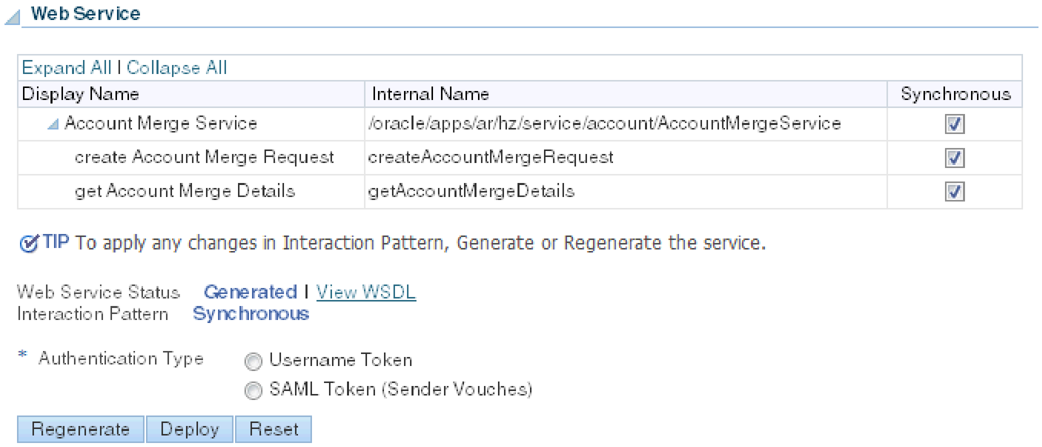
Note: It's important to note the following for PL/SQL based concurrent program:

- Although at a PL/SQL layer, any concurrent programs can be submitted by FND_REQUEST API, Oracle E-Business Suite Integrated SOA Gateway supports calling of different concurrent programs through separate concurrent program services.
- There may be PL/SQL based APIs exposed through the Integration Repository that are not consistent with the synchronous, auto-committed transaction state of the Web Service Framework in Oracle E-Business Suite Integrated SOA Gateway.
- The WSDL generated by Oracle E-Business Suite Integrated SOA Gateway marks schema elements (parameters) and its related schemas as optional or mandatory, based on the method signature of the underlying API. However, runtime behavior may vary based on API internal implementation.

After Service Generation

The Web Service region or the SOAP Web Service tab contains the following information:

Web Service Region



Web Service

Expand All | Collapse All

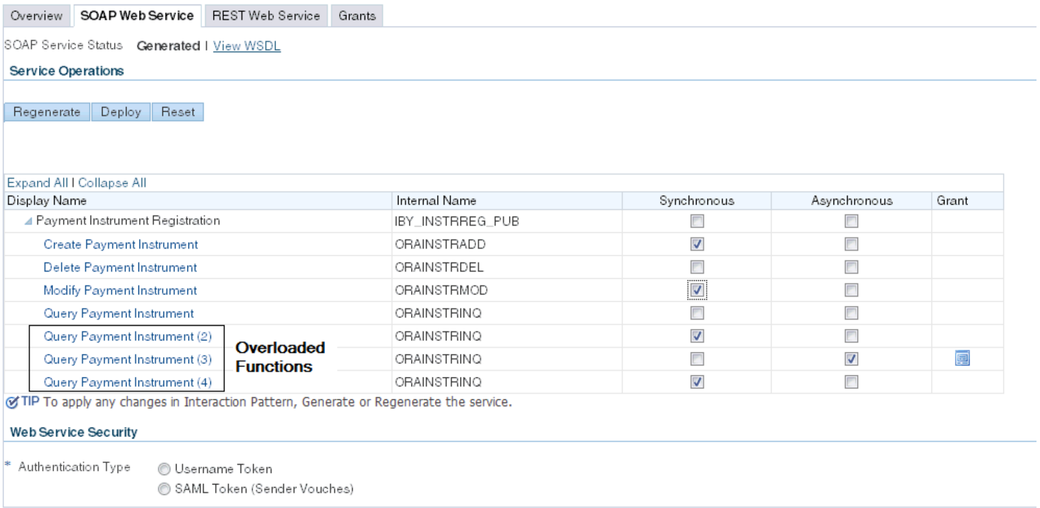
Display Name	Internal Name	Synchronous
Account Merge Service	/oracle/apps/ar/hz/service/account/AccountMergeService	☑
create Account Merge Request	createAccountMergeRequest	☑
get Account Merge Details	getAccountMergeDetails	☑

TIP To apply any changes in Interaction Pattern, Generate or Regenerate the service.

Web Service Status Generated | View WSDL
Interaction Pattern Synchronous

* Authentication Type ○ Username Token
○ SAML Token (Sender Vouches)

SOAP Web Service Tab



Overview | **SOAP Web Service** | REST Web Service | Grants

SOAP Service Status **Generated** | View WSDL

Service Operations

Regenerate | Deploy | Reset

Expand All | Collapse All

Display Name	Internal Name	Synchronous	Asynchronous	Grant
Payment Instrument Registration	IBY_INSTRREG_PUB	□	□	
Create Payment Instrument	ORAINSTRADD	☑	□	
Delete Payment Instrument	ORAINSTRDEL	□	□	
Modify Payment Instrument	ORAINSTRMOD	☑	□	
Query Payment Instrument	ORAINSTRINQ	□	□	
Query Payment Instrument (2)	ORAINSTRINQ	☑	□	
Query Payment Instrument (3)	ORAINSTRINQ	□	☑	▦
Query Payment Instrument (4)	ORAINSTRINQ	☑	□	

Overloaded Functions

TIP To apply any changes in Interaction Pattern, Generate or Regenerate the service.

Web Service Security

* Authentication Type ○ Username Token ○ SAML Token (Sender Vouches)

Note: For overloaded functions, sequence number is added to the end of the overloaded method name. Each overloaded function can be uniquely selected and generated with your desired interaction pattern.

- **Interaction Pattern Table:** Selected method names with desired interaction pattern(s) are displayed in the table.

  If change on the interaction pattern table is required for a generated service:

  - If the generated service has not yet been deployed, after modification you must regenerate the service. Upon regeneration, the service definition will be changed to reflect the changes made in the table. You need to modify its Web service clients based on the new service definition.

  - If the generated service has already been deployed, you must first undeploy the service, modify the pattern selection, regenerate the service, and then deploy the service again.

    For information on service deployment, see Deploying and Undeploying SOAP Web Services, page 3-10.

  If the generated service is of XML Gateway or Concurrent Program interface type, this table is not displayed.

- **Web Service Status:** After a service has been generated successfully, the service status is changed from 'Not Generated' to 'Generated'.

  Important: Multiple requests to generate Web services for an integration interface are not allowed. If service generation is still in progress, then 'Generating' is displayed as the service status and the **Generate** button is disabled.

- **Interaction Pattern:** 'Synchronous' is displayed by default in the Web Service region if the selected interface is not a PL/SQL API.

- **View WSDL Link:** Click this link to view the generated WSDL description for the selected interface.

  Please note that if a method is exposed as a serviceable operation with the support of asynchronous pattern, then ASYNCH appears in the WSDL for that method to distinguish it from the rest of the operations generated synchronously.

  For example, if 'Asynchronous' is selected specifically for the 'CREATE_INVOICE' method within the Invoice Creation API (AR_INVOICE_API_PUB) interface, after service generation, the ASYNCH appears in the CREATE_INVOICE operation for both input and output messages as well as binding.

```
...
<portType name="AR_INVOICE_API_PUB_PortType">
   <operation name="CREATE_INVOICE_ASYNCH">
     <input name="tns:CREATE_INVOICE_Input_Msg"/>
   </operation>
</portType>
<portType name="AR_INVOICE_API_PUB_Callback_PortType">
   <operation name="CREATE_INVOICE_ASYNCH_RESPONSE">
     <input name="tns:CREATE_INVOICE_Output_Msg"/>
   </operation>
</portType>
...

<binding name="AR_INVOICE_API_PUB_Binding"
type="tns:AR_INVOICE_API_PUB_PortType">
   <operation name="CREATE_INVOICE_ASYNCH">
     <soap:operation soapAction="CREATE_INVOICE_ASYNCH" />
       <input>
         <soap:header message="tns:CREATE_INVOICE_Input_Msg"
part="header" use="literal" />
         <soap:body use="literal" parts="body" />
       </input>
     </operation>
</binding>
<binding name="AR_INVOICE_API_PUB_CallBack_Binding"
type="tns:AR_INVOICE_API_PUB_CallBack_PortType">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="CREATE_INVOICE_ASYNCH_RESPONSE">
      <soap:operation soapAction="CREATE_INVOICE_ASYNCH_RESPONSE" />

        <input>
         ...
        </input>
    </operation>
 </binding>
```

> For more information about WSDL, see: Reviewing SOAP Service WSDL Source, *Oracle E-Business Suite Integrated SOA Gateway User's Guide*.

After service generation, if the interface definition has been changed or the selected interaction pattern information has been modified before service deployment, you can regenerate the service by clicking **Regenerate**. However, if interface definition is not changed, then regenerating the service will not change the service definition.

Click **Reset** to clear up the existing service artifact and change the Web Service Status field from 'Generated' to 'Not Generated'. See: Resetting SOAP Web Services, page 3-15 .

To deploy the generated service, the administrator must select one desired authentication type in the Authentication Type region. The selected authentication type will be used to authenticate Oracle E-Business Suite users at run time. For more information on deploying a service, see Deploying and Undeploying SOAP Web Services, page 3-10.

Displaying Generic XML Gateway Service Subregion for Generic XML Gateway Services

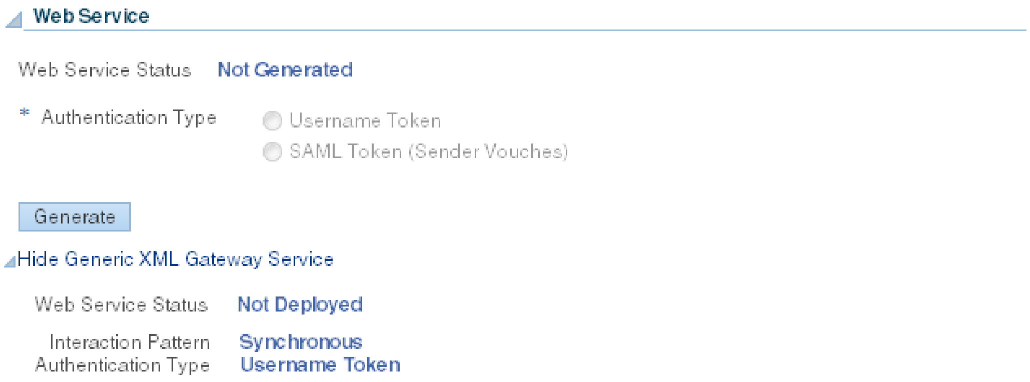
For XML Gateway interface type, if your system is upgraded from a previous release and if you have been using generic XML Gateway Web services, the generic XML Gateway service information can be displayed by setting the *FND: XML Gateway Map Generic Service* profile value to 'Yes'.

In the Web Service region, click the **Show Generic XML Gateway Service** or **Hide Generic XML Gateway Service** link to display or close the Generic XML Gateway Service subregion for the selected XML Gateway interface.

For more information on setting profile options, see Setting Profile Options, page 2-3.

In addition to setting profile options, the administrator needs to perform additional setup tasks for generic XML Gateway services. For setup information, see *Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12.2*, My Oracle Support Knowledge Document 1311068.1 for details.

Generic XML Gateway Service Subregion



Web Service Status Not Generated

* Authentication Type ○ Username Token ○ SAML Token (Sender Vouches)

Web Service Status Not Deployed

Interaction Pattern Synchronous

Authentication Type Username Token

The Generic XML Gateway Service subregion contains the following fields:

- **Web Service Status:** This field indicates the current state of the selected XML Gateway interface.

  If the setup is not configured for generic XML Gateway services, the Web Service Status field is displayed as 'Not Deployed'.

- **View Generic WSDL:** Click the **View Generic WSDL** link to display the deployed generic WSDL URL for the selected XML Gateway interface.

  The deployed generic WSDL URL has the following syntax:

  ```
  http://<SOA server host>:<SOA Suite managed server
  port>/soa-infra/services/default/XMLGatewayService!<version
  chosen while deploying>XMLGateway?WSDL
  ```

  - `<SOA Suite managed server port>`: It is the port of the server where SOA composite is deployed.

  - `<version chosen while deploying>`: At the time of deployment, deployement version will be asked. Default version value is 1.0.

    For example, `http://<SOA server host>:<SOA Suite managed server port>/soa-infra/services/default/XMLGatewayService!1.0/XMLGateway?WSDL`.

  Please note that after the upgrade to Oracle E-Business Suite Release 12.2, the deployed WSDL URL information has been changed from an earlier release. Therefore, you may have to replace it with the new WSDL URL and service location or address accordingly in Web service clients while invoking the generic XML Gateway service.

  The updated WSDL URL is also populated in the *ISG: Generic Service WSDL URL for XMLG* profile option by default if the setup tasks for generic XML Gateway services are configured properly.

- **Interaction Pattern:** 'Synchronous' is displayed by default in read-only mode.
- **Authentication Type:** 'Username Token' is displayed by default in read-only mode.

To generate a Web service:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility or through custom responsibility and navigation path. Select the Integration Repository link.

2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.

3. Expand an interface type node to locate your desired interface definition.

4. Click the interface definition name link to open the interface details page.

5. If this selected interface definition does not have service generated, specify at least one interaction pattern in the Interaction Pattern table. This can be done at the interface level or at the method level before clicking **Generate** in the Web Service region to generate the WSDL description.

   For interfaces that can be supported with both REST and SOAP services, **Generate** is located in the Service Operations region of the SOAP Web Service tab in the interface details page.

   After service generation, the interaction pattern table and the Interaction Pattern field are displayed with selected pattern information for your interface.

   The Web Service Status field marked as 'Generated' also appears which indicates that this selected interface has WSDL description available.

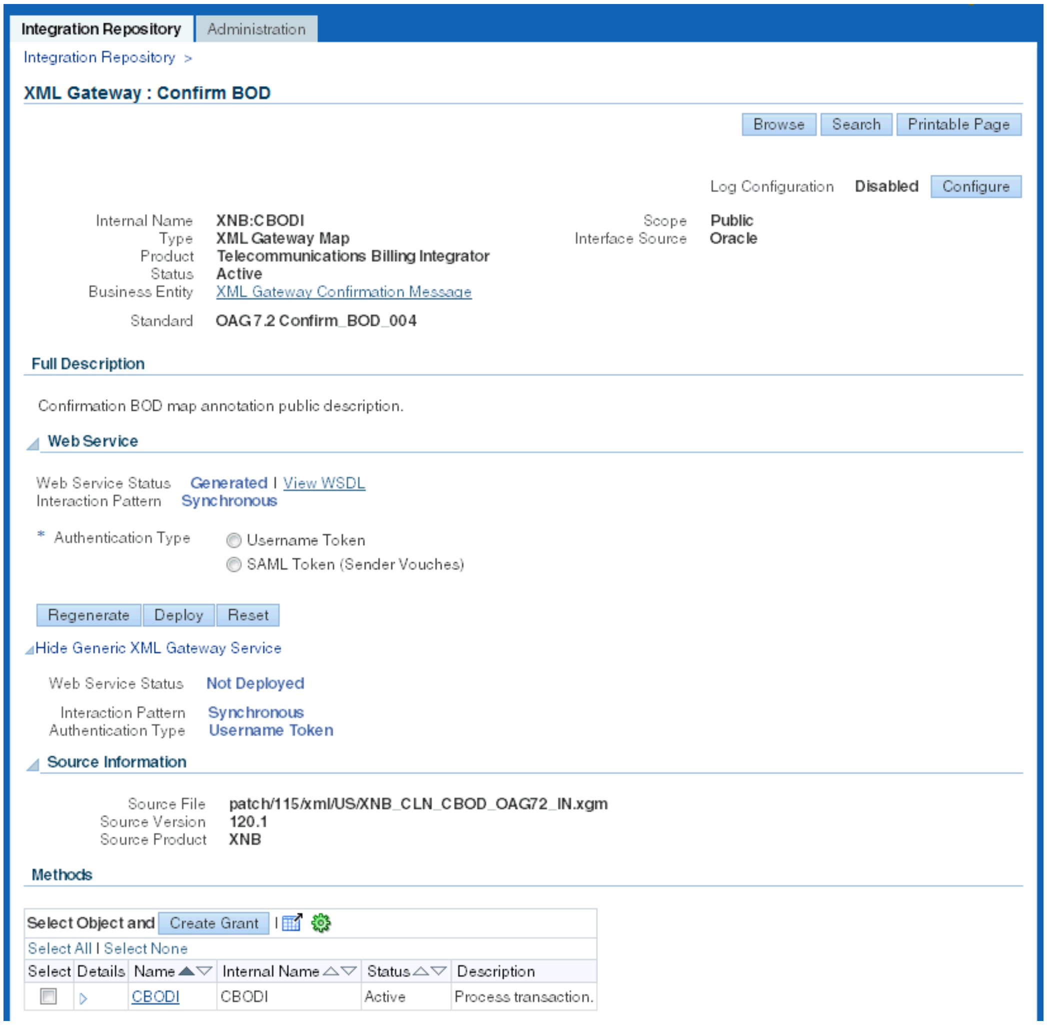
6. Click the View WSDL link to view the WSDL description.

7. Click **Regenerate** to regenerate the WSDL description if necessary.

Deploying and Undeploying SOAP Web Services

If a SOAP service has been generated successfully, the administrator has the privilege to deploy the generated service in the Web Service region or the SOAP Web Service tab if the interface can be exposed as both SOAP and REST services.

Deploying Web Services



XML Gateway : Confirm BOD

Log Configuration **Disabled**

Internal Name	XNB:CBODI	Scope	Public
Type	XML Gateway Map	Interface Source	Oracle
Product	Telecommunications Billing Integrator		
Status	Active		
Business Entity	XML Gateway Confirmation Message		
Standard	OAG 7.2 Confirm_BOD_004		

Full Description

Confirmation BOD map annotation public description.

Web Service

Web Service Status Generated | View WSDL
Interaction Pattern Synchronous

* Authentication Type
○ Username Token
○ SAML Token (Sender Vouches)

Hide Generic XML Gateway Service

Web Service Status Not Deployed
Interaction Pattern Synchronous
Authentication Type Username Token

Source Information

Source File patch/115/xml/US/XNB_CLN_CBOD_OAG72_IN.xgm
Source Version 120.1
Source Product XNB

Methods

Select	Details	Name	Internal Name	Status	Description
☐	▷	CBODI	CBODI	Active	Process transaction.

Deploying Web Services with Authentication Types

Prior to deploying a SOAP Web service, the administrator must first select one of the following authentication types:

- Username Token

  This authentication type provides username and password in the security header for a Web service provider to use in authenticating the users. It is the concept of Oracle E-Business Suite username/password (or the username/password created through the Users window in defining an application user).

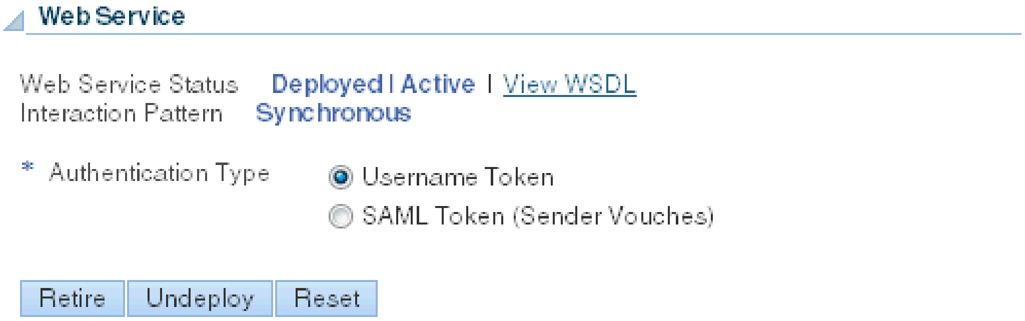
- SAML Token (Sender Vouches)

  This authentication type is used for Web services relying on sending a username only through SAML Assertion.

Deployment with Active State

Once a SOAP Web service has been successfully deployed, the newly deployed service has 'Deployed with Active' service status in Oracle SOA Suite where Oracle E-Business Suite services can be used at run time.

Web Service Region After Service Deployment



Web Service

Web Service Status Deployed | Active | View WSDL

Interaction Pattern Synchronous

* Authentication Type

● Username Token

○ SAML Token (Sender Vouches)

The Web Service region (or the SOAP Web Service tab for the interface with the support for both SOAP and REST services) has the following changes:

- The service status is changed from 'Generated' to 'Deployed' with 'Active' state indicating that the deployed service is ready to be invoked and accept new SOAP requests.

- The selected authentication type is displayed.

- Click the **View WSDL** link to display the deployed WSDL information. It shows the physical location of service endpoint where the service is hosted in `soa-infra`.

- The following buttons appear if the service has been successfully deployed with 'Active' state:

  - **Retire**: It disables the active service. The service status is changed to 'Deployed' with 'Retired' state indicating that this deployed service will no longer accept new requests. It also ensures that current running requests are finished.

    Once the service has been successfully retired, the **Activate** button appears allowing you to activate the retired service. For more information on retiring and activating Web services, see:

    - Retiring SOAP Web Services, page 3-17

    - Activating SOAP Web Services, page 3-18

- **Undeploy**: It undeploys the Web service from Oracle SOA Suite back to Oracle Integration Repository. Deployed services can be undeployed with the following reasons:

    - Changes on an interface definition for a deployed service.
    - Changes on interaction pattern for a deployed service.
    - Changes on the Authentication Type field for a deployed service.
    - The original service was corrupt.

    After undeploying the service, make desired changes first (such as interaction pattern or authentication type). Next, regenerate the service, and then deploy the service again.

- **Reset**: It clears up the deployed service artifact and changes the service status from 'Deployed' with 'Active' to 'Not Generated'.

  For more information, see Resetting SOAP Web Services, page 3-15.

For more information on service generation, see Generating SOAP Web Services, page 3-4.

For more information on supported authentication types, see Managing Web Service Security, page 6-8.

Reviewing Deployed WSDL

To view the deployed Web service, click the **View WSDL** link. The following example shows the deployed WSDL code:

> **Note:** Please note that the deployed WSDL shows the physical location of service endpoint where the service is hosted in the `soa-infra` in `<soap:address location>` element. Generated WSDL does not display the physical service endpoint, but with the following information:
>
> ```
> <soap:address location="#NOT_DEPLOYED#" />
> ```

```
<definitions name="ECRDTLD"
targetNamespace="http://xmlns.oracle.com/apps/ec/soaprovider/concurrentp
rogram/ecrdtld/">
<documentation>
 <abstractWSDL>

http://<hostname>:<port>/soa-infra/services/default/<jndi_name>_CONCURRE
NTPROGRAM_ECRDTLD!1/ECRDTLD_soap.wsdl
</abstractWSDL>
</documentation>
<types>
   <schema elementFormDefault="qualified"
targetNamespace=http://xmlns.oracle.com/apps/ec/soaprovider/concurrentpr
ogram/ecrdtld/">
  <include
schemaLocation="http://<hostname>:<port>/soa-infra/services/default/<jnd
i_name>_CONCURRENTPROGRAM_ECRDTLD/ECRDTLD_Service/?XSD=APPS_ISG_CP_REQUE
ST_CP_SUBMIT.xsd"/>
   </schema>
   <schema elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/apps/ec/soaprovider/concurrentp
rogram/ecrdtld/">
  <element name="SOAHeader">
    <complexType>
     <sequence>
      <element name=="Responsibility" minOccurs="0" type="string"/>
      <element name="RespApplication" minOccurs="0" type="string"/>
      <element name="SecurityGroup" minOccurs="0" type="string" />
      <element name="NLSLanguage" minOccurs="0" type="string" />
      <element name="Org_Id" minOccurs="0" type="string" />
     </sequence>
   </complexType>
  </element>
 </schema>
</types>
<message name="ECRDTLD_Input_Msg">
   <part name="header" element="tns1:SOAHeader"/>
   <part name="body" element="tns1:InputParameters"/>
</message>
<message name="ECRDTLD_Output_Msg">
   <part name="body" element="tns1:OutputParameters"/>
</message>
<portType name="ECRDTLD_PortType">
   <operation name="ECRDTLD">
     <input message="tns1:ECRDTLD_Input_Msg"/>
     <output message="tns1:ECRDTLD_Output_Msg"/>
   </operation>
 </portType>
<binding name="ECRDTLD_Binding" type="tns1:ECRDTLD_PortType">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="ECRDTLD">
   <soap:operation soapAction="ECRDTLD"/>
     <input>
       <soap:header message="tns1:ECRDTLD_Input_Msg" part="header"
use="literal"/>
       <oap:body use="literal" parts="body"/>
     </input>
     <output>
       <soap:body use="literal"/>
     </output>
```

```
</operation>
  </binging>
  <service name="ECRDTLD_Service">
    <port name="ECRDTLD_Port" binding="tns1:ECRDTLD_Binding">
     <soap:address
location="http://<hostname>:<port>/soa-infra/services/default/<jndi_name
>_CONCURRENTPROGRAM_ECRDTLD/ECRDTLD_Service/"/>
   </port>
 </service>
</definitions>
```

To deploy or undeploy a Web service:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility or through custom responsibility and navigation path. Select the Integration Repository link.

2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.

3. Expand an interface type node to locate your desired interface definition.

4. Click the interface definition name link to open the interface details page.

5. From the Web Service region (or the SOAP Web Service tab), select one of the following authentication types:

   - Username Token

   - SAML Token (Sender Vouches)

6. Click **Deploy** to deploy the service with active state to an Oracle SOA Suite WebLogic environment.

7. Click the deployed **View WSDL** link to view the deployed WSDL description.

8. Click **Undeploy** to undeploy the service.

9. If a service has been deployed with active state, **Retire** appears lets you disable the active service so that it will no longer accept new requests.

10. Click **Reset** to clear up the existing service artifact.

Resetting SOAP Web Services

Once an integration interface becomes a Web service, the associated service artifact is also generated. No matter if the generated service has been deployed or not, you can clear up the service artifact and *reset* the Web service status to its initial state - 'Not Generated' regardless of its current state. This action can be performed at any stage of service generation and deployment life cycle.

For example, an interface definition needs to be modified or has been changed. Instead of regenerating the service if it has not yet been deployed, or undeploying the service if it has been deployed, you can:

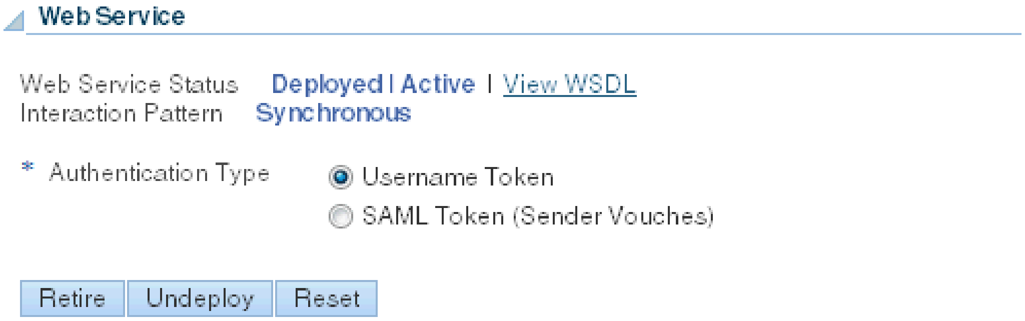
1. Reset the service to clear up the existing service artifact.

2. Modify the interface.

3. Generate the service again.

For information on how to generate a Web service for a given interface, see Generating SOAP Web Services, page 3-4.

Resetting a Service After Service Deployment



Web Service

Web Service Status Deployed | Active | View WSDL

Interaction Pattern Synchronous

* Authentication Type ● Username Token

○ SAML Token (Sender Vouches)

To reset a Web service:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility or through custom responsibility and navigation path. Select the Integration Repository link.

2. Click **Search** to open the main Search page.

3. Enter appropriate search information such as product family, product, interface type, or business entity.

4. Click **Show More Search Options** and select 'Deployed' or 'Generated' in the Web Service Status field.

5. Locate the interface definition that match your search criteria from the result table.

6. Click the interface definition name link to open the interface details page.

7. In the Web Service region (or the SOAP Web Service tab for the interface with the support for both SOAP and REST services), click **Reset** to clear up existing service artifact for the selected service. The service status is changed to 'Not Generated'.

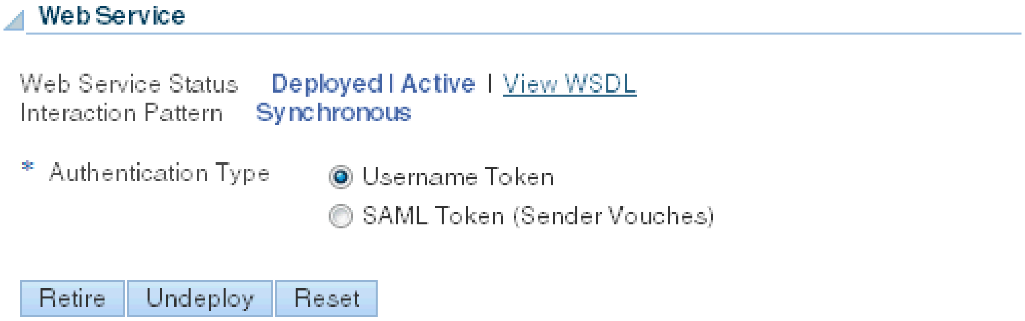
Retiring SOAP Web Services

When a service has been successfully deployed to Oracle SOA Suite with active state, **Retire** appears allowing you to change the state of the deployed service from 'Active' to 'Retired'.

> **Note:** This action also ensures that current running requests are finished while retiring the service.

Service with 'Retired' state means that the deployed service is no longer active for service invocation and will not accept new SOAP requests.

Web Service Region After Service Deployment



Web Service

Web Service Status Deployed | Active | View WSDL

Interaction Pattern Synchronous

* Authentication Type ● Username Token

○ SAML Token (Sender Vouches)

Please note that a service with 'Retire' state, the selected interaction pattern and authentication type information remains the same.

After retiring a deployed service, the Web Service region (or the SOAP Web Service tab for the interface with the support for both SOAP and REST services) has the following changes:

- **Web Service Status:** 'Deployed' with 'Retired' state appears indicating that this deployed service will no longer accept new requests.

- **Activate:** This action lets you change the retired service back to an active service again.

  For information on how to activate a Web service, see Activating SOAP Web Services, page 3-18.

- **Undeploy:** This action lets you undeploy the retired service from an Oracle SOA Suite managed server to the repository. See: Deploying and Undeploying SOAP Web Services, page 3-10.

- **Reset:** This action lets you reset the retired service to its initial state - 'Not Generated'. See: Resetting SOAP Web Services, page 3-15.

To retire a Web service:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility or through custom responsibility and navigation path. Select the Integration Repository link.

2. Click **Search** to open the main Search page.

3. Enter appropriate search information such as product family, product, interface type, or business entity.

4. Click **Show More Search Options** and select 'Deployed' for the Web Service Status field.

5. Locate the interface definition that match your search criteria from the result table.

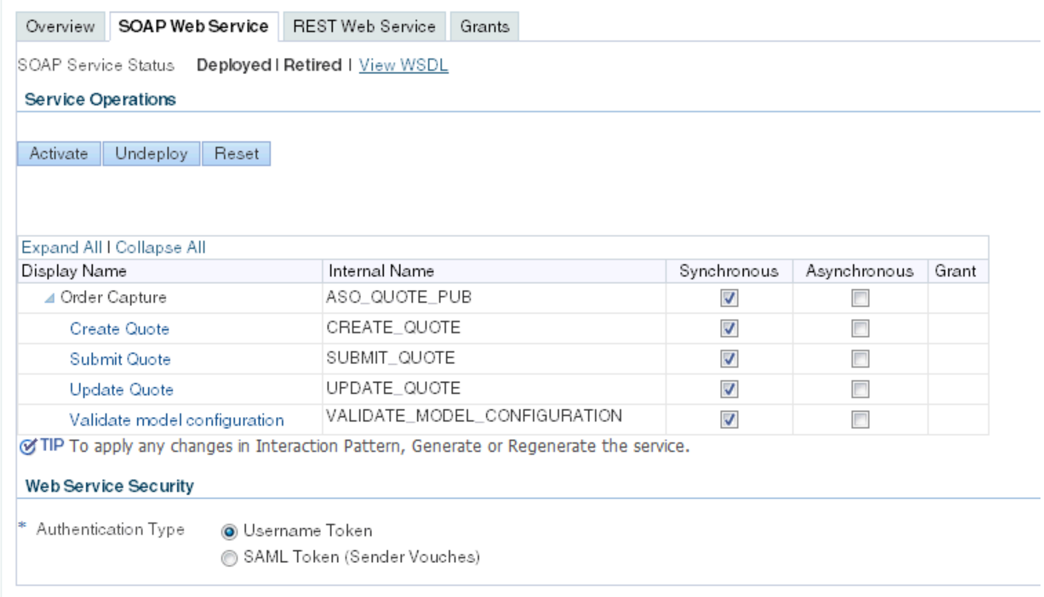
6. Click the interface definition name link to open the interface details page.

7. In the Web Service region (or the SOAP Web Service tab for the interface with the support for both SOAP and REST services), click **Retire** if needed to retire the active deployed service.

Activating SOAP Web Services

After a service has been deployed with 'Retired' state, it is not available to participate in any Web service activities at run time. To bring it back to work and to be invoked by Web service clients, you must change the 'Retired' state to 'Active'. This can be achieved by clicking **Activate** to take the retired service back to an active state again.

SOAP Web Service Tab After Service Deployment for a PL/SQL API



SOAP Service Status Deployed | Retired | View WSDL

Service Operations

Display Name	Internal Name	Synchronous	Asynchronous	Grant
Order Capture	ASO_QUOTE_PUB	☑	□	
Create Quote	CREATE_QUOTE	☑	□	
Submit Quote	SUBMIT_QUOTE	☑	□	
Update Quote	UPDATE_QUOTE	☑	□	
Validate model configuration	VALIDATE_MODEL_CONFIGURATION	☑	□	

TIP To apply any changes in Interaction Pattern, Generate or Regenerate the service.

Web Service Security

* Authentication Type ● Username Token
○ SAML Token (Sender Vouches)

Please note that activating a service will not change its service definition. This means that the selected interaction pattern and authentication type information remains the same as it was before.

After activating a service, the following fields are changed in the Web Service region (or the SOAP Web Service tab for the interface with the support for both SOAP and REST services):

- **Web Service Status:** This field is changed from 'Deployed' with 'Retired' state back to 'Deployed' with 'Active' state. This indicates that the deployed service becomes available again and is ready to be invoked and accept new requests.

- **Retire:** This action lets you retire the activated service again. See: Retiring SOAP Web Services, page 3-17.

- **Undeploy:** This action lets you undeploy the active service from an Oracle SOA Suite managed server to the repository. See: Deploying and Undeploying SOAP Web Services, page 3-10.

- **Reset:** This action cleans up the service artifact and takes it back to its initial state - 'Not Generated'. See: Resetting SOAP Web Services, page 3-15.

To activate a retired Web service:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility or through custom responsibility and navigation path. Select the Integration Repository link.

2. Click **Search** to open the main Search page.

3. Enter appropriate search information such as product family, product, interface type, or business entity.

4. Click **Show More Search Options** and select 'Deployed' for the Web Service Status field.

5. Locate the interface definition that match your search criteria from the result table.

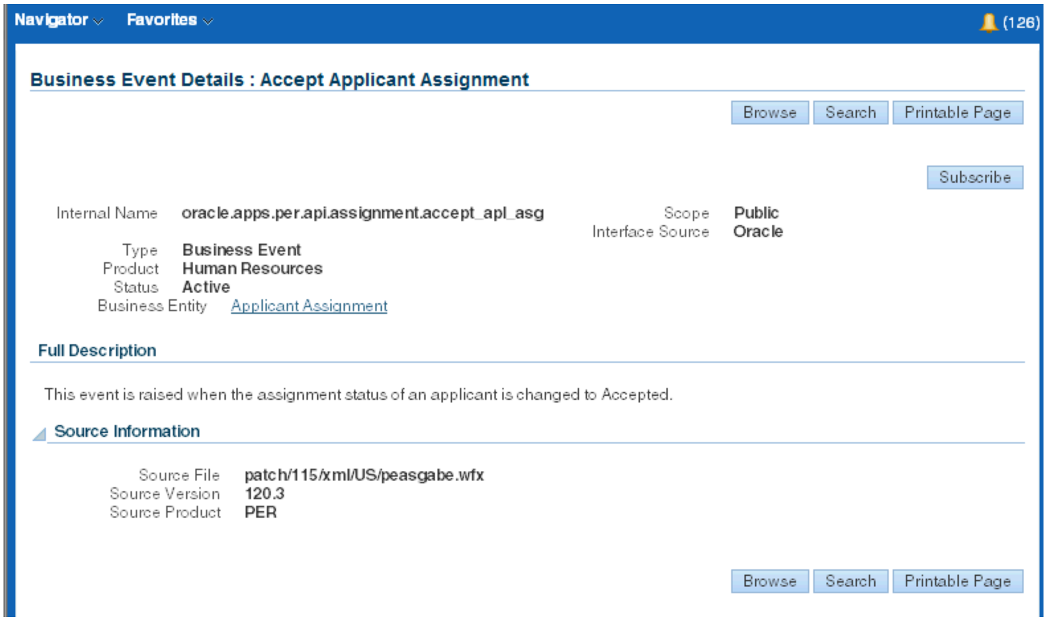
6. Click the interface definition name link to open the interface details page.

7. In the Web Service region (or the SOAP Web Service tab), click **Activate** if available to activate the retired service.

Subscribing to Business Events

An integration repository administrator can find **Subscribe** in the business event interface details page which allows the administrator to subscribe to a selected business event and create an event subscription for that selected event.

Subscribing to a Business Event



Business Event Details : Accept Applicant Assignment

Internal Name	oracle.apps.per.api.assignment.accept_apl_asg	Scope	Public
		Interface Source	Oracle
Type	Business Event		
Product	Human Resources		
Status	Active		
Business Entity	Applicant Assignment		

Full Description

This event is raised when the assignment status of an applicant is changed to Accepted.

Source Information

Source File	patch/115/xml/US/peasgabe.wfx
Source Version	120.3
Source Product	PER

Internally, an event subscription is automatically created for that event with `WF_BPEL_QAGENT` as Out Agent. Once the event subscription has been successfully created, a confirmation message appears on the Business Event interface detail page.

To consume the business event message, you should register to dequeue the event from Advanced Queue `WF_BPEL_Q`. If a business event is enabled and if there is at least one subscriber registered to listen to the WF_BPEL_Q queue, then the event message will be enqueued in WF_EVENT_T structure to Advanced Queue WF_BPEL_Q.

Unsubscribing to Business Events

Once an event subscription has been successfully created, **Unsubscribe** appears instead. Clicking **Unsubscribe** removes the event subscription from the WF_BPEL_Q queue. A confirmation message also appears after the subscription has been successfully removed.

For more information on how to dequeue messages, see the *Oracle Streams Advanced Queuing User's Guide and Reference*.

For more information about business events, see Managing Business Events, *Oracle Workflow Developer's Guide*.

To subscribe to a business event:

1. Log on to Oracle Integration Repository with the Integration Repository Administrator role through the Integrated SOA Gateway responsibility or through custom responsibility and navigation path. Select the Integration Repository link.

2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.

3. Expand the Business Event interface type node to locate your desired event.

4. Click the business event interface that you want to subscribe to it to open the Interface details page for the event.

5. Click **Subscribe** to subscribe to the selected event. Internally, an event subscription is created with Out Agent as WF_BPEL_QAGENT. A confirmation message appears after the event subscription is successfully created.

   Remove the subscribed event by clicking **Unsubscribe** to remove or delete the event subscription if needed.

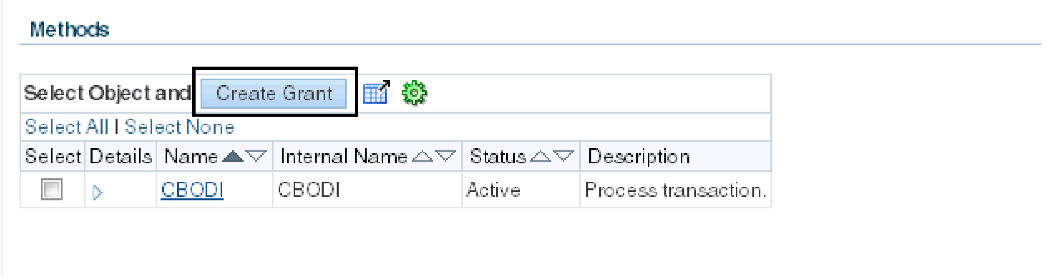
Managing Security Grants for SOAP Web Services Only

To protect application data from unauthorized access, Oracle E-Business Suite Integrated SOA Gateway provides security grant feature allowing only authorized users to execute certain methods in an API through Integration Repository.

Managing Grants in the Methods Region

For interfaces that can be exposed as SOAP services only, security grants are managed in the Methods region.

Managing Grants in the Methods Region



Select	Details	Name ▲▽	Internal Name △▽	Status △▽	Description
□	▷	CBODI	CBODI	Active	Process transaction.

> **Note:** In this release, only PL/SQL interfaces can be exposed as both SOAP and REST services. For this type of interfaces, security grants are managed in the Grants tab instead. Once a PL/SQL API method access permission is authorized to a grantee, it grants the permission to the associated SOAP and REST services simultaneously. For information on how to manage security grants for PL/SQL interfaces, see Managing Security Grants for SOAP and REST Web Services, page 3-41.

Creating Security Grants

In the interface details page, an integration repository administrator can select appropriate method name check boxes in the Methods region. Click **Create Grant** to open the Create Grants page where the administrator can grant the selected method access permissions to a user, user group, or all users.

Select one of the following values as the grantee type:

- `Specific User` - The grantee is an individual user who was selected directly.
- `Group of Users` - The grantee is a group of users or a member of a group of users.
- `All Users` - The grant was given to all users.

If you select `Specific User` or `Group of Users`, specify the user or group for which to create the grants in the Grantee Name field.

> - Only users with the Integration Repository Administrator role can create and revoke security grants.
> - Each overloaded function contained in an interface can be uniquely granted to a specific user, user group, or all users through the grant feature. If you select more than one overloaded function in the Procedures and Functions region (or the Methods region), an Overloaded column appears in the Selected Methods table with the

selected overloaded functions checked.

Viewing Grant Details

To view the grant details, click the **Show** link for a given method in the Methods region. If you specified a group of users as the grantee, then all members within the group (i.e. 'Jackson, Lou' and 'Payment, John'), plus the group name itself (i.e. 'OIC Payment Analyst Manger Group') are listed as a grantee.

> **Note:** For each member, the Granted Via column displays the name of the group. For grantees who were selected directly in the Create Grants page, the value in the Granted Via column is `Direct`.

Revoking Security Grants

In the Methods region, click the **Show** link for a given method. Click the **Revoke** icon to revoke a grant for a specific grantee. A confirmation page appears, where you can click **Apply** or **Cancel** to execute or cancel the action.

> **Note:** For users who are granted as members of a group, you cannot revoke their grants individually, but revoke the grant for the entire group instead. The **Revoke** icon is disabled for group members.

To create grants:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility or through custom responsibility and navigation path. Select the Integration Repository link.

2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.

3. Expand an interface type node and click an interface definition name link you want to open the interface details page.

4. Select one or multiple method names for which you want to create grants.

5. Click **Create Grant**. The Create Grants page appears.

6. Select a grantee type from the list of values.

7. If you selected `Specific User` or `Group of Users,` specify the user or group for which to create the grants in the Grantee Name field.

8. Click **Apply**.

   The interface details page reappears.

To view or revoke grants:

You can view and revoke existing grants directly in the methods list on the interface details page.

1. Navigate to the interface details page that you want to view or revoke the grants.

2. In the Methods region, click **Show** for a given method to view its grant details in a table.

3. You can revoke a grant by clicking the **Revoke** icon. Click **Apply** to confirm your action.

Enabling Design-Time Log Configuration

To troubleshoot any issues or exceptions encountered during service generation and deployment life cycle, users with the Integration Repository Administrator role can enable design-time log setting for a selected interface.

If the design-time log is enabled for the selected interface, 'Enabled' is displayed as the Log Configuration value in the header section of the interface details page. Otherwise, 'Disabled' is displayed instead.

To change the existing design-time log configuration for the selected interface, click **Configure** next to the Log Configuration field. The Log & Audit Setup Details page is displayed with the selected interface where the administrator can add a new log configuration or update existing configurations.

> **Note:** The Log & Audit Setup Details page can also be accessed by selecting the **Administration > Configuration** from the navigation menu.

Please note that this feature applies to an interface with the support for SOAP services only.

- For detailed information about how to configure log settings at the integration interface level, see Adding a New Configuration, page 7-6.

- For information on how to view logs and errors collected for the selected interface during the design-time activities, see Viewing Generate and Deploy Time Logs, page 3-24

Viewing Generate and Deploy Time Logs

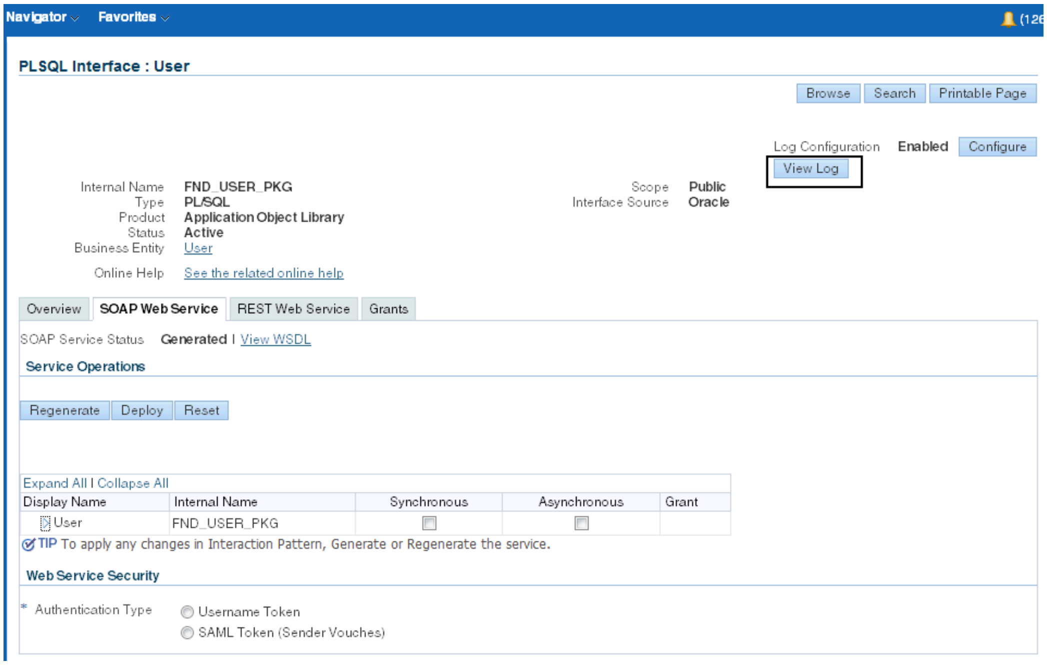
To effectively troubleshoot any issues or exceptions encountered at design time during each stage of service generation and deployment life cycle including generating, deploying, retiring, resetting, and activating services, error messages and activity information can be logged and viewed through the interface details page.

Note: Logging is supported for SOAP services only.

- If the design-time log is enabled for an interface, **View Log** appears in that interface details page allowing you to view both log messages and error messages if occurred during design time.

- If the design-time log is not enabled for an interface, and errors occurred while performing the design-time activities, **View Error** appears instead allowing you to view the error messages only.

For information on how to enable the design-time log for an interface, see Adding a New Configuration, page 7-6.



PLSQL Interface : User

Log Configuration Enabled

Internal Name FND_USER_PKG
Type PL/SQL
Product Application Object Library
Status Active
Business Entity User

Online Help See the related online help

Scope Public
Interface Source Oracle

SOAP Service Status Generated | View WSDL

Service Operations

Display Name	Internal Name	Synchronous	Asynchronous	Grant
User	FND_USER_PKG	□	□	

TIP To apply any changes in Interaction Pattern, Generate or Regenerate the service.

Web Service Security

* Authentication Type ○ Username Token
○ SAML Token (Sender Vouches)

Viewing Error and Log Details from the View Log Button

If an interface has the design-time log enabled, **View Log** appears in the interface details page allowing you to access the Log & Error Details page.

The Log & Error Details page contains the following regions:

- **Error Details region:** If any errors or exceptions encountered during the design-time activities such as Generate, Deploy, Undeploy, Reset, Retire, and Activate services, error messages are displayed in the Error Details region.

- **Log Details region:** All design-time logs recorded for the selected service are listed in the table. Each log contains log sequence, log timestamp, module, log level, and actual message recorded at the design time.

Deleting and Exporting Logs in the Log Details Region

After viewing log messages retrieved for an interface in the Log Details region, you can delete them if needed by clicking **Delete Log**. A warning message appears alerting you that this will permanently delete all the logs retrieved in the region. Click **Yes** to confirm the action. An empty log table appears after logs have been successfully deleted.

Before deleting the logs, you can save a backup copy by clicking **Export**. This allows you to export the records listed in the Log Details region to Microsoft Excel and use it later.

Viewing Error Details from the View Error Button

If the selected interface does not have the design-time log enabled, and if any errors occurred during design-time activities, **View Error** appears instead allowing you to view only the error or exception messages displayed in the Error Details region.

For example, if the administrator receives errors or exceptions while trying to perform any actions at design time such as Generate, Deploy, Activate, Retire, or Reset for an interface, these errors are recorded and displayed in the Error Details region even if the design-time log is not configured for the interface.

For error messages, error codes, and possible solutions, see Error Messages, page C-1.

The Log Details region will not appear in this page because the design-time log is not configured for the selected interface.

For more logging information, see Logging for Web Services, page 7-1. For information on how to add a new configuration, see Adding a New Configuration, page 7-6.

At run time during the invocation of Oracle E-Business Suite services by Web service clients, if a service has the runtime log enabled, log messages can be viewed in Service Monitor against that instance. For information on viewing log messages through Service Monitor, see Viewing Service Processing Logs, page 8-8.

To view service development log messages:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility. Select the Integration Repository link.

2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.

3. Expand an interface type node to locate your desired interface definition.

4. Click the interface definition name link to open the interface details page.

5. If the selected interface does not have the design-time log enabled, **View Error** appears instead in the interface details page if errors occurred during the

design-time activities.

Click **View Error** to view the error details that occurred during design time.

6. If the selected interface has the design-time log enabled, **View Log** appears in the interface details page.

   Click **View Log** to view the log and error details.

   Click **Delete Log** to delete all the logs listed in the table if needed.

   Click **Export** to export log list table to Microsoft Excel and save the records.

Managing SOAP Service Life Cycle Activities Using An Ant Script

An Ant script `$JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml` is used to execute the design-time activities for SOAP services such as generate, regenerate, deploy, undeploy, activate, retire, and reset services as well as to upgrade or postclone services from command line.

Please note that `$JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml` is a multipurpose script. It can also be used to run the diagnostic tests or download the configuration file from the instance. The configuration file is the present state of instance in the view of Oracle E-Business Suite Integrated SOA Gateway context. The same configuration file is sometimes referred as service descriptor file.

> **Note:** When services are generated from command line, the settings selected from the Integration Repository user interface will take effect while generating the service artifacts. For example, if 'Asynchronous' interaction pattern is selected for a method contained in a PL/SQL interface, no matter if the service is generated from the UI or command line, only that selected single method has the associated artifact generated for asynchronous operation.

Usage of $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml:

```
ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml usage
```

> **Note:** Script creates log file at the script location; hence, it is suggested to copy `isgDesigner.xml` to some `<TEMP_DIRECTORY>` and then use the script present in `<TEMP_DIRECTORY>`.

Usage Related to Design Activities

1. Enter `ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml`. It will prompt for the arguments.

Note: Do not enclose any input between double quotes.

2. Enter the arguments in the following ways:

   -
     ```
     ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
     -Dactions=<comma separated list of operations>
     -DserviceType=SOAP -DirepNames=<comma separated list of
     API Names> -Dverbose=<ON|OFF>
     ```

     While passing actions and irepNames using this method, be aware of the following conditions:

     - If more than one actions or irepNames are passed as command line argument, enclose them between double quotes. For example,

       ```
       -Dactions="method1, method2,.."
       ```

       ```
       -DirepNames="ECRDTLD,FND_USER_PKG[func1:SY::func2:AS::.
       ..]"
       ```

     - If only one action or irepName is passed as command line argument, then there is no need to enclose between double quotes.

   -
     ```
     ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
     -Dfile=<absolute path of service descriptor file>
     -Dverbose=<ON|OFF>
     ```

Argument Description

Valid arguments for `isgDesigner.xml` are described as follows:

- **actions**: Comma separated list of actions to be performed. Supported operations are: generate, deploy, undeploy, activate, retire, reset, upgrade, postclone.

  - generate: It will generate or regenerate the service.

  - deploy: It will deploy the generated service.

  - undeploy: It will undeploy the deployed service.

  - activate: It will activate the deployed service if it is in 'Retire' state.

  - retire: It will retire the deployed service if it is in 'Active' state.

  - reset: It will reset the Web service status to its initial state - 'Not Generated' and will also delete artifacts from the file system of Oracle SOA Suite server.

  - upgrade: It will upgrade a service from Oracle E-Business Suite Release 12.1.X to Release 12.2.

  - postclone: It will carry out postclone steps, such as redeploying the services, on the Release 12.2 cloned environment.

  While passing the action names, ensure that they have been given in the order of their life cycle. For example:

  - Incorrect Usage: `-Dactions="deploy,generate"`
  - Correct usage: `-Dactions="generate,deploy"`

  Actions 'upgrade' and 'postclone' should be called independently. This means if upgrade action is given, actions argument should look like `-Dactions=upgrade`. It is similar to the case with action 'postclone'. More information on how actions arguments are used is described in the following examples:

  - `-Dactions="generate,deploy,retire,activate,undeploy,reset"`
  - `-Dactions=upgrade`
  - `-Dactions=postclone`

  Additionally, if action is 'upgrade' or 'postclone', only 'actions' and 'verbose' arguments will be used. However if you have given other arguments as well, only the three arguments mentioned above will be used.

- **serviceType**: [SOAP|REST]: Choose the default value SOAP.
- **irepNames**: Comma separated list of API names.

  For example, `-DirepNames="FND_USER_PKG"`
- **file**: Absolute path of the (service descriptor) XML file containing interfaces and actions to be performed on these interfaces.

  For example, `-Dfile=/u01/oracle/isg_service.xml`
- **verbose**: [ON|OFF] Default value is OFF.

  For example, `-Dverbose=OFF`

Usage Examples

- Sample command for actions other than 'upgrade' and 'postclone' (actions and interface names are being passed):

  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
  -Dactions="generate,deploy,undeploy" -DserviceType=SOAP
  -DirepNames="ECRDTLD,FND_USER_PKG"
  ```
- Sample command for performing design time actions from XML file:

  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
  -Dfile=/u01/oracle/isg_service.xml
  ```

- Sample command for action 'upgrade':

  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
  -Dactions=upgrade -Dverbose=OFF
  ```

- Sample command for action 'postclone':

  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
  -Dactions=postclone -Dverbose=ON
  ```

Other Usages

In addition to performing design time activities, this `$JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml` script can be used for the following purposes:

- Deploying Generic XML Gateway Services, page 3-30
- Downloading Service Descriptor File for the Current Environment, page 3-30
- Downloading the Sample Service Descriptor File, page 3-30
- Obtaining Argument irepNames Usage Information, page 3-32
- Running Diagnostic Tests, page 3-33

Deploying Generic XML Gateway Services

To deploy a generic XML Gateway service for the current environment, invoke this script with target *deployGenericXMLG*

```
ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
deployGenericXMLG
```

For more information on deploying generic XML Gateway services, see *Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12.2*, My Oracle Support Knowledge Document 1311068.1 for details.

Downloading Service Descriptor File for the Current Environment

To download service descriptor file for the current environment, invoke this script with target *DownloadConfiguration*

```
ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
DownloadConfiguration
```

Downloading the Sample Service Descriptor File

To download the sample service descriptor file, invoke this script by giving target *filehelp*

```
ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
filehelp
```

This downloads one sample service descriptor file and also prints the grammar for service descriptor file. Grammar will look like as following:

Service descriptor file should conform to following grammar:

Note: Meaning of the special characters:

- + : One or more occurrence
- * : Zero or more occurrence
- ? : Zero or one occurrence

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<IntegrationRepository name="Instance_SID">
  <services>
    <interface>
      <name>API_Name</name>
<serviceType>SOAP</serviceType>
      <actions>
        <action>valid action name</action> +
      </actions>
      <!--functions node is necessary if generate action is given-->
      <functions selective="true">
        <function name="function_name" pattern="SY"/> *
      </functions> ?
      <!--policies node is necessary if deploy action is given-->
      <policies>
        <policy>policy_name</policy> +
      </policies> ?
    </interface> +
  </services>
</IntegrationRepository>
```

For example, a SOAP service descriptor file can be like:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<IntegrationRepository name="atgisgqa">
  <services>
    <interface>
      <name>ASO_QUOTE_PUB</name>
      <serviceType>SOAP</serviceType>
      <actions>
       <action>reset</action>
       <action>generate</action>
       <action>deploy</action>
      </actions>
      <functions selective="false"/>
      <policies>
       <policy>USERNAME</policy>
      </policies>
    </interface>
  </services>
</IntegrationRepository>
```

Important elements in the descriptor file for SOAP services are:

- `<serviceType>` node must be present if 'generate' action is given.
- `<functions>` node must be present if 'generate' action is given.
- `<policies>` node must be present if 'deploy' action is given.

- The 'selective' attribute of `<functions>` node must be false if individual function names are not given.

  If 'selective' attribute of `<functions>` node is false then user can give one more attribute of this node namely 'pattern'. All function of the corresponding interface will be generated with this pattern. If 'pattern' attribute is not present, all functions will be generated with default pattern such as synchronous pattern.

  Supported values for attribute 'pattern' are described in the following:

  - **SY:** This is for synchronous generation.
  - **AS:** This is for asynchronous generation.
  - **BO:** This is for both synchronous and asynchronous generations.

Obtaining Argument irepNames Usage Information

To know how to pass argument `irepNames`, invoke this script with target *irepNamehelp*

```
ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
irepNamehelp
```

This prints the following information on console window:

Each interface name for the `irepNames` argument should be given in one of the following way:

- `interface_name[func1:pattern1;func2:pattern2;...]`
- `interface_name[pattern1]`
- `interface_name`

Usage Example:
FND_USER_PKG[TESTUSERNAME:SY;CHANGE_USER_NAME:AS],FND_MESSAGE[AS],FND_GLOBAL

> **Note:** Patterns supported here are described in the following:
>
> - **SY:** This is for synchronous generation.
> - **AS:** This is for asynchronous generation.
> - **BO:** This is for both synchronous and asynchronous generations.

interface_name[func1:pattern1;func2:pattern2]

- Function `func1` of interface `interface_name` will be generated with pattern pattern1.
- Function `func2` of interface `interface_name` will be generated with pattern

pattern2.

interface_name[pattern1]

All functions of interface `interface_name` will be generated with pattern pattern1.

interface_name

All functions of the interface `interface_name` will be generated with old pattern or default pattern.

Running Diagnostic Tests

Oracle E-Business Suite Integrated SOA Gateway provides a suite of diagnostic tests to help determine specific causes or issues with installation steps. When a test suite is run, multiple tests would be executed on both Oracle E-Business Suite and Oracle SOA Suite environments for diagnosing issues on various categories.

To know how to run different diagnostic tests, invoke this script with *diagnosticshelp*

```
ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
diagnosticshelp
```

Additionally, you can run different diagnostics through the backend script with different targets. For more information on how to run these diagnostic tests, see Oracle E-Business Suite Integrated SOA Gateway Diagnostic Tests, page A-1.

Administering REST Web Services

Oracle E-Business Suite Integrated SOA Gateway exposes PL/SQL interfaces as both REST services and SOAP services. REST services support only *synchronous* (request-response and request-only) interaction pattern and have a simplified service life cycle.

> **Note:** In this release, PL/SQL API is the only interface type that can be exposed as both SOAP and REST services.

PL/SQL REST services can be used for user-driven applications such as mobile, tablet, or handheld devices. Security services are used for mobile applications to validate or invalidate user credentials, initialize user sessions with applications context, and authorize users.

Simplified Service Life Cycle

REST services have a simplified service life cycle. The administrator can perform the following tasks in the REST Web Service tab to manage the REST service life cycle:

- Deploy a Service

  A PL/SQL interface can be exposed as a REST service through a 'Deploy' action. Unlike SOAP services deployed to an Oracle SOA Suite WebLogic managed server, REST services are deployed to an Oracle E-Business Suite WebLogic managed server.

- Undeploy a Service

  The administrator can undeploy a deployed REST service. This action not only undeploys the REST service, but also resets the service to its initial state - 'Not Deployed'. Any existing or running service requests will be completed and no new request is honored.

The administrator can manage security grants in the Grants tab of the interface details page. It assigns grants to specific users to access or invoke the deployed REST services.

Supporting Security Services - Predeployed REST Services

In addition to exposing PL/SQL APIs as both REST and SOAP services, Oracle E-Business Suite Integrated SOA Gateway supports Oracle Application Object Library's Authentication and Authorization services as REST security services.

Unlike other service-enabled interfaces requiring administrative actions on service development, security services are a set of predeployed REST services which can be invoked by all the Oracle E-Business Suite users.

Security services support token based authentication for invoking other REST services. With token based authentication, it is possible to authenticate a user once based on username and password, and then authenticate the user in the consecutive REST requests using a security token (such as Oracle E-Business Suite user session ID). For more information about the REST service security, see REST Service Security, page 3-36 .

REST services can be deployed and undeployed from the Integration Repository user interface and from the Ant script $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml.

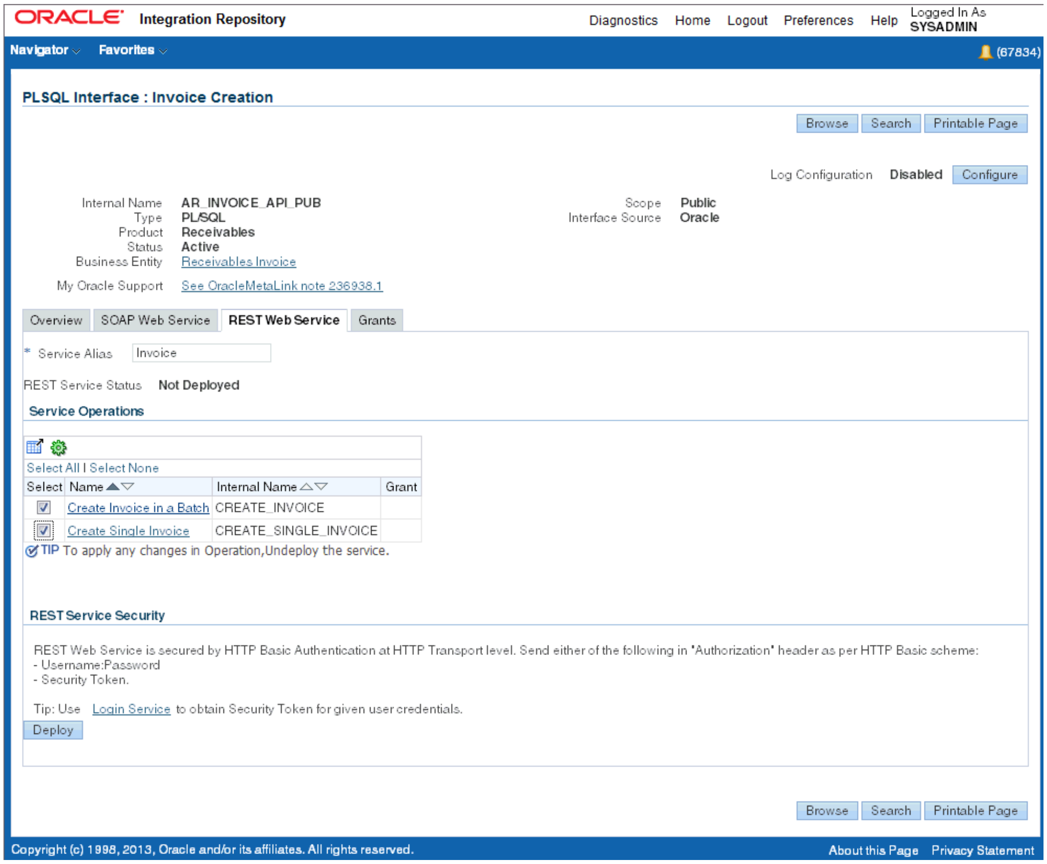
To better understand each administrative task, the following topics are included in this section:

- Deploying REST Web Services, page 3-34
- Undeploying REST Web Services, page 3-39
- Managing Grants for Interfaces with Support for Both SOAP and REST Services, page 3-41
- Managing REST Service Life Cycle Activities Using An Ant Script, page 3-46

Deploying REST Web Services

Oracle E-Business Suite Integrated SOA Gateway allows the administrator or the system integration developer to deploy a PL/SQL interface definition as a REST service.

Deploying Web Services



PLSQL Interface : Invoice Creation

Log Configuration Disabled

Internal Name	AR_INVOICE_API_PUB	Scope	Public
Type	PL/SQL	Interface Source	Oracle
Product	Receivables		
Status	Active		
Business Entity	Receivables Invoice		
My Oracle Support	See OracleMetaLink note 236938.1		

* Service Alias: Invoice

REST Service Status: Not Deployed

Service Operations

Select All | Select None

Select	Name	Internal Name	Grant
☑	Create Invoice in a Batch	CREATE_INVOICE	
☑	Create Single Invoice	CREATE_SINGLE_INVOICE	

TIP To apply any changes in Operation,Undeploy the service.

REST Service Security

REST Web Service is secured by HTTP Basic Authentication at HTTP Transport level. Send either of the following in 'Authorization' header as per HTTP Basic scheme:
- Username:Password
- Security Token.

Tip: Use Login Service to obtain Security Token for given user credentials.

Deploying REST Services in the REST Web Service Tab

Before deploying a REST service, the administrator must perform the following tasks:

- **Specify service alias**

  Each REST service should be associated with a unique alias name. Alias is a set of characters and used in the service endpoint which shortens the URL for the service.

  For example, 'Invoice' is entered as the service alias for an interface Create Invoice (AR_INVOICE_API_PUB) before being deployed. The alias will be displayed as the service endpoint in the schema for a selected service operation CREATE_INVOICE as follows:

  ```
  href="https://<hostname>:<port>/webservices/rest/Invoice
  /?XSD=CREATE_INVOICE_SYNCH_TYPEDEF.xsd" />
  ```

- **Select desired service operations**

  In the Service Operations table, select one or more methods to be exposed as REST service operations.

  For example, select CREATE_INVOICE from the table for the selected interface

Create Invoice (AR_INVOICE_API_PUB). After service deployment, only the selected method CREATE_INVOICE will be exposed as a REST service operation.

REST Service Security

All REST services are secured by HTTP Basic Authentication or Token Based Authentication at HTTP or HTTPS transport level. Either one of the authentication methods will be used in authenticating users who invoke the REST services.

- *HTTP Basic Authentication:* This authentication is for an HTTP client application to provide username and password when making a REST request that is typically over HTTPS.

- *Token Based Authentication:* This security method authenticates a user using a security token provided by the server. When a user tries to log on to a server, a token (such as Oracle E-Business Suite session ID) may be sent along with username in place of password. This authentication method can be used in multiple consecutive REST invocations.

  For example, an Oracle E-Business Suite user has been initially authenticated on a given username and password. After successful login, the security Login service creates an Oracle E-Business Suite user session and returns the session ID. The session ID that points to the user session will be passed to HTTP headers of all subsequent Web service calls for user authentication.

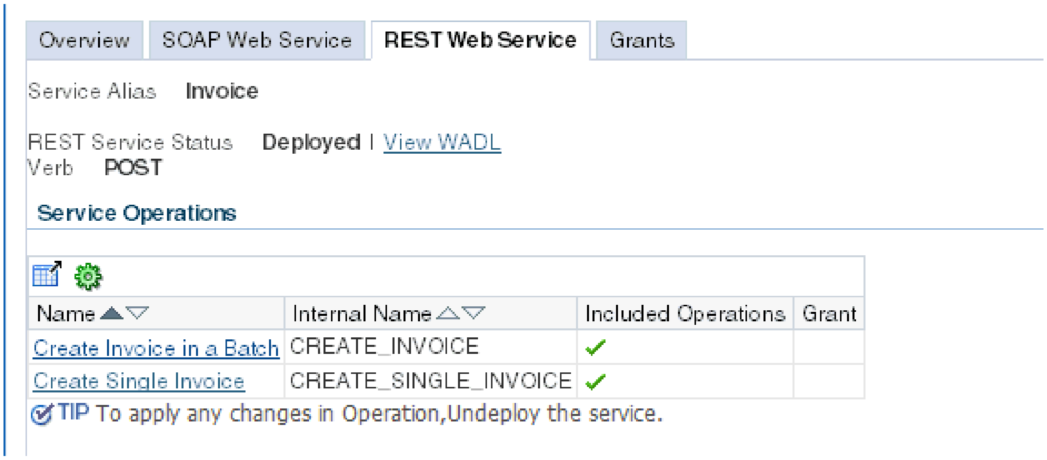
  > **Note:** Login service validates the user credentials and returns an access token. It is a predeployed Java security service, and is part of the Authentication services that help validate and invalidate users, as well as initialize applications context required by the service before being invoked.
  >
  > For more information on applications context in REST service, see REST Header for Applications Context, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.
  >
  > For more information on supported authentication types, see Managing Web Service Security, page 6-8.

Click **Deploy** to deploy the selected service operations to an Oracle E-Business Suite managed server for consumption.

After Service Deployment

Once the REST service has been successfully deployed, the REST Web Service tab has the following changes:

REST Web Service Tab After Service Deployment



Service Alias **Invoice**

REST Service Status **Deployed** | View WADL
Verb **POST**

Service Operations

Name ▲▽	Internal Name △▽	Included Operations	Grant
Create Invoice in a Batch	CREATE_INVOICE	✓	
Create Single Invoice	CREATE_SINGLE_INVOICE	✓	

TIP To apply any changes in Operation,Undeploy the service.

- **Service Alias**: The REST alias should be displayed as a read-only text field.

- REST Service Status: This field is changed from its initial state 'Not Deployed' to 'Deployed' indicating that the deployed service is ready to be invoked and to accept new requests.

- **View WADL**: The **View WADL** link is displayed. Click the link to display the deployed WADL information.

  It shows the physical location of the service endpoint where the service is hosted.

- **Verb**: This field displays the Verb value indicating how the REST service is implemented using an HTTP method. Please note that 'POST' is the only method supported in this release.

  This field is displayed only when the REST service has 'Deployed' status.

- **Service Operations**: This table displays the list of methods (or procedures and functions) contained in the selected interface.

  - If the methods have been exposed as REST service operations, the Included Operations will be selected.

  - Click the **Grant** icon to view the read-only grant details for a selected method.

Reviewing Deployed WADL

To view the deployed REST service WADL, click the **View WADL** link.

The following example shows the deployed WADL for the selected CREATE_INVOICE service operation contained in the PL/SQL API Invoice Creation (`AR_INVOICE_API_PUB`):

> **Note:** Please note that 'Invoice' highlighted here is the service alias entered earlier prior to the service deployment. After the service is deployed, the specified alias name (Invoice) becomes part of the service endpoint in the .xsd schema file.
>
> For more information about WADL description, see Reviewing WADL Element Details, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<application
xmlns:tns="http://xmlns.oracle.com/apps/ar/soaprovider/plsql/rest/ar_inv
oice_api_pub/" xmlns="http://wadl.dev.java.net/2009/02"
xmlns:tns1="http://xmlns.oracle.com/apps/ar/rest/ar/create_invoice/"
name="AR_INVOICE_API_PUB"
targetNamespace="http://xmlns.oracle.com/apps/ar/soaprovider/plsql/rest/
ar_invoice_api_pub/">
   <grammars>
    <include xmlns="http://www.w3.org/2001/XMLSchema"
href="https://host01.example.com
:1234/webservices/rest/Invoice/?XSD=CREATE_INVOICE_SYNCH_TYPEDEF.xsd" />

   </grammars>
 <resources base="http://host01.example.com:1234/webservices/rest/
Invoice/">
  <resource path="/create_invoice/">
   <method id="CREATE_INVOICE" name="POST">
    <request>
     <representation mediaType="application/xml"
type="tns1:InputParameters" />
     <representation mediaType="application/json"
type="tns1:InputParameters" />
    </request>
    <response>
     <representation mediaType="application/xml"
type="tns1:OutputParameters" />
     <representation mediaType="application/json"
type="tns1:OutputParameters" />
    </response>
   </method>
  </resource>
 </resources>
</application>
```

To deploy a REST Web service:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility or through custom responsibility and navigation path. Select the Integration Repository link.

2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.

3. Expand an interface type node to locate your desired interface definition.

4. Click the interface definition name link to open the interface details page.

5. In the REST Web Service tab, enter the following information:

   - Service Alias: Specify service alias information.

   - In the Service Operations table, select one or more methods to be exposed as REST service operations.

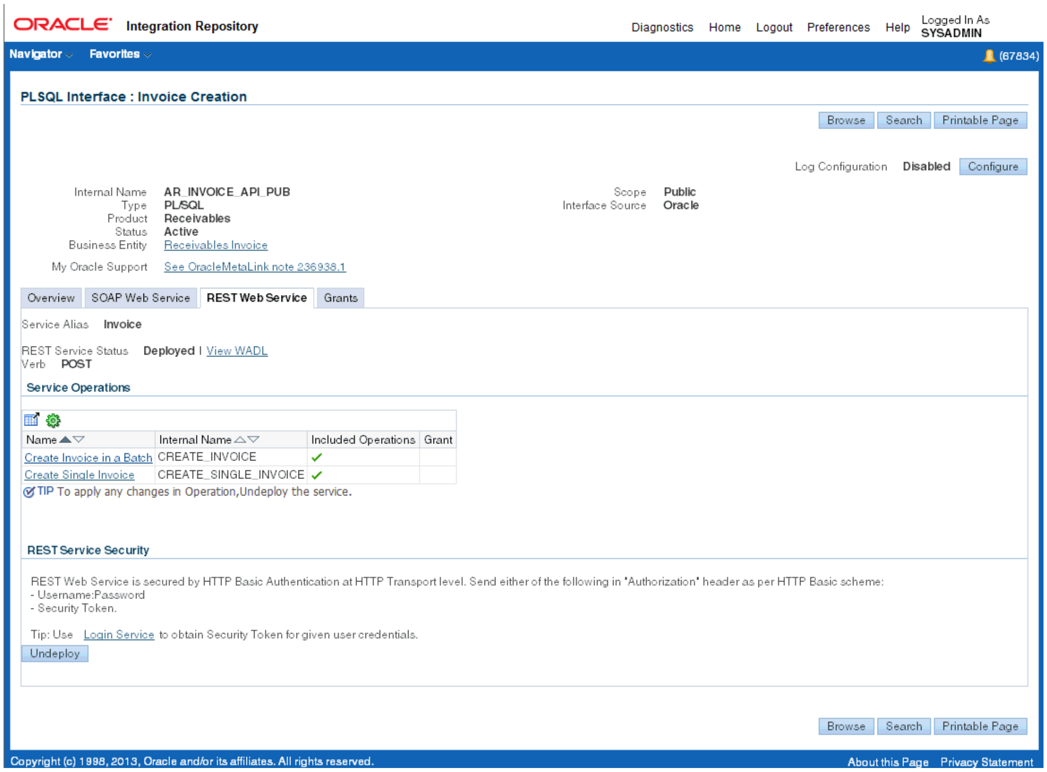
6. Click **Deploy** to deploy the service to an Oracle E-Business Suite environment.

7. Click the deployed **View WADL** link to view the deployed WADL description.

Undeploying REST Web Services

Once a REST service has been successfully deployed, the **Undeploy** button appears in the REST Web Service tab. This allows the administrator to undeploy the service and at the same time to bring the service back to its initial state - 'Not Deployed'.

Undeploying REST Services


ORACLE Integration Repository
Diagnostics Home Logout Preferences Help
Navigator Favorites
(67834)
Browse Search Printable Page
Configure
Overview SOAP Web Service REST Web Service Grants
Undeploy
Browse Search Printable Page
About this Page Privacy Statement

Logged In As SYSADMIN

PLSQL Interface : Invoice Creation

Log Configuration Disabled

Internal Name	AR_INVOICE_API_PUB	Scope	Public
Type	PL/SQL	Interface Source	Oracle
Product	Receivables		
Status	Active		
Business Entity	Receivables Invoice		
My Oracle Support	See OracleMetaLink note 236938.1		

Service Alias Invoice

REST Service Status Deployed | View WADL
Verb POST

Service Operations

Name	Internal Name	Included Operations	Grant
Create Invoice in a Batch	CREATE_INVOICE	✓	
Create Single Invoice	CREATE_SINGLE_INVOICE	✓	

TIP To apply any changes in Operation,Undeploy the service.

REST Service Security

REST Web Service is secured by HTTP Basic Authentication at HTTP Transport level. Send either of the following in 'Authorization' header as per HTTP Basic scheme:
- Username:Password
- Security Token.

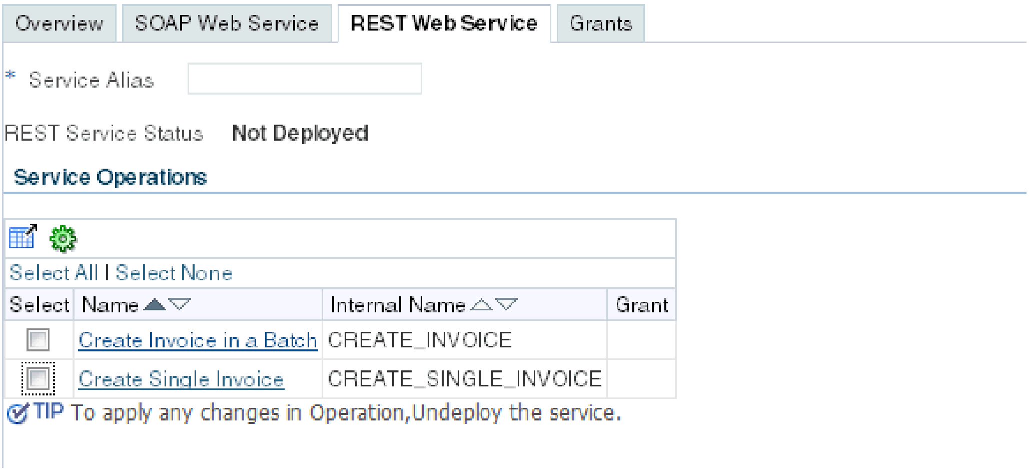
Tip: Use Login Service to obtain Security Token for given user credentials.

Copyright (c) 1998, 2013, Oracle and/or its affiliates. All rights reserved.

Please note that when a service is undeployed, any existing or running service requests will be completed and no new request is honored. The associated service artifact will be removed from the system.

After a successful undeployment, 'Not Deployed' is shown in the REST Service Status field. The value of the service alias entered earlier now disappears which allows the administrator to enter it again before next deployment.

REST Web Service Tab After Service Undeployment



* Service Alias

REST Service Status Not Deployed

Service Operations

Select	Name	Internal Name	Grant
☐	Create Invoice in a Batch	CREATE_INVOICE	
☐	Create Single Invoice	CREATE_SINGLE_INVOICE	

TIP To apply any changes in Operation,Undeploy the service.

To undeploy a REST Web service:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility or through custom responsibility and navigation path. Select the Integration Repository link.

2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.

3. Expand an interface type node to locate your desired interface definition.

4. Click the interface definition name link to open the interface details page.

5. In the REST Web Service tab, click **Undeploy** to undeploy the service.

Managing Grants for Interfaces with Support for Both SOAP and REST Web Services

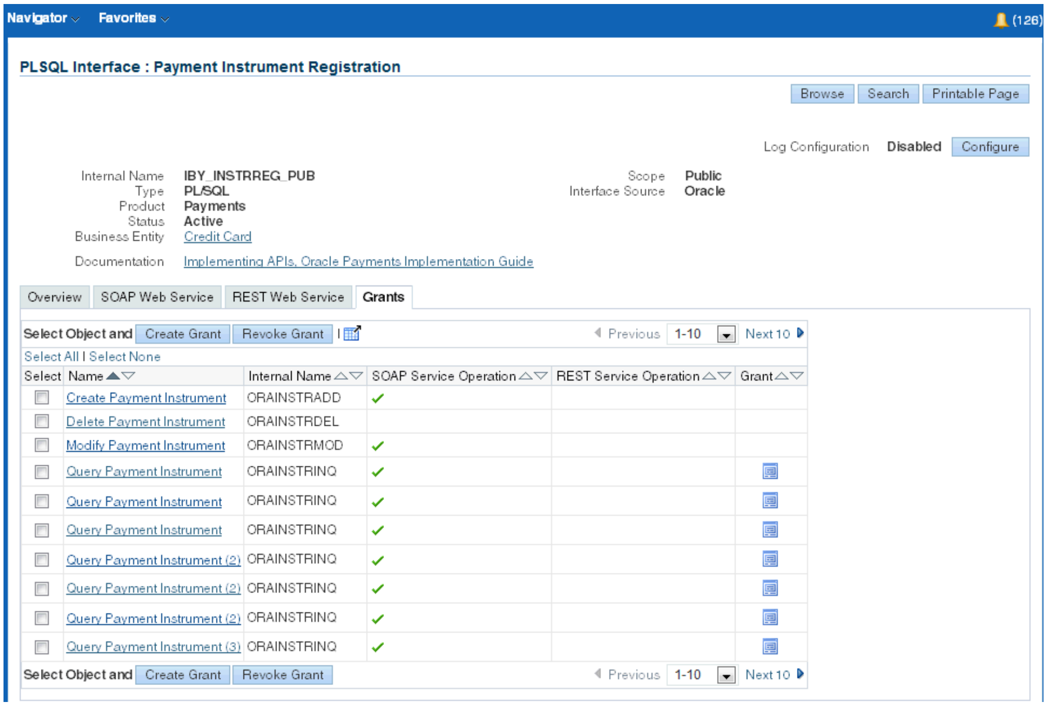
Users with the Integration Repository Administrator role can create grants to a specific user, users, or a group of users. Grants given to a user for specific services or operations are applicable for both SOAP and REST services.

> **Note:** In this release, only PL/SQL APIs can be exposed as both SOAP and REST services.

Managing Grants in the Grants Tab for PL/SQL Interfaces

Security grants for PL/SQL APIs are managed in the Grants tab of the interface details page.

Managing Grants in the Grants Tab



PLSQL Interface : Payment Instrument Registration

Log Configuration **Disabled** Configure

Internal Name	IBY_INSTRREG_PUB	Scope	Public
Type	PL/SQL	Interface Source	Oracle
Product	Payments		
Status	Active		
Business Entity	Credit Card		
Documentation	Implementing APIs, Oracle Payments Implementation Guide		

Select Object and Create Grant Revoke Grant | ◀ Previous 1-10 Next 10 ▶

Select All | Select None

Select	Name	Internal Name	SOAP Service Operation	REST Service Operation	Grant
□	Create Payment Instrument	ORAINSTRADD	✓		
□	Delete Payment Instrument	ORAINSTRDEL			
□	Modify Payment Instrument	ORAINSTRMOD	✓		
□	Query Payment Instrument	ORAINSTRINQ	✓		▤
□	Query Payment Instrument	ORAINSTRINQ	✓		▤
□	Query Payment Instrument	ORAINSTRINQ	✓		▤
□	Query Payment Instrument (2)	ORAINSTRINQ	✓		▤
□	Query Payment Instrument (2)	ORAINSTRINQ	✓		▤
□	Query Payment Instrument (2)	ORAINSTRINQ	✓		▤
□	Query Payment Instrument (3)	ORAINSTRINQ	✓		▤

Select Object and Create Grant Revoke Grant ◀ Previous 1-10 Next 10 ▶

For interfaces with the support for SOAP services only, security grants are managed in the Methods region instead. See: Managing Security Grants for SOAP Web Services Only, page 3-21.

Creating Security Grants

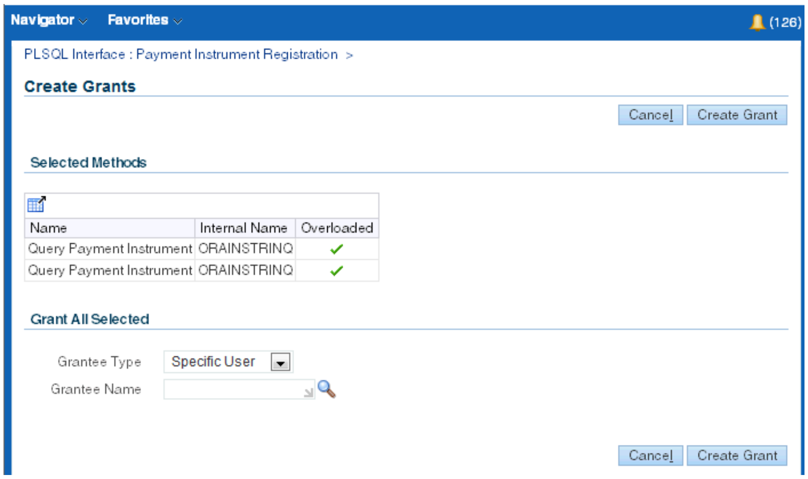
The administrator can select one or more procedures and functions or methods contained in the selected interface, and then click **Create Grant**. The Create Grants page is displayed where the administrator can grant the selected method access permissions to a user, user group, or all users.

Once a method access permission is authorized to a grantee, it grants the permission to access the associated SOAP and REST service operations simultaneously. For example, when a user (OPERATIONS) is authorized to have access permission on a method called 'Change User Name', regardless if the method has been exposed as a SOAP or REST service operation or not, the user OPERATIONS has the permission to access the 'Change User Name' operation of BOTH service types through the same grant.

- PL/SQL interfaces can be exposed as SOAP services with the support for both synchronous and asynchronous patterns. The security grants given for the selected method names would be applicable to the generated services of both patterns.

- If a selected interface contains overloaded functions, each of them can be uniquely granted through the create grant feature. If you select more than one overloaded function for the grant, an Overloaded column appears in the table with the selected function names checked.

Security Grants with Overloaded Functions



PLSQL Interface : Payment Instrument Registration >

Create Grants

Selected Methods

Name	Internal Name	Overloaded
Query Payment Instrument	ORAINSTRINQ	✓
Query Payment Instrument	ORAINSTRINQ	✓

Grant All Selected

Grantee Type: Specific User

Grantee Name:

Revoking Security Grants

The administrator can revoke security grants in the following ways:

- *Revoking Commonly Assigned Grants to All Selected Procedures or Methods*

  Select more than one procedure and function or method that you want to revoke the grants created earlier, and click **Revoke Grant**. This opens the Revoke Grants page where you can find the existing grants that are commonly assigned to the selected methods.

  For example, a selected interface has the following grants:

Method Names	Grantee
Change User Name	SYSADMIN **OPERATIONS**
Test User Name	**OPERATIONS** MKTMGR BUSER

Method Names	Grantee
Validate User Name	BUSER **OPERATIONS**

A specific User (grantee type) 'OPERATIONS' (grantee name) is commonly authorized to all the methods contained in the selected interface. Therefore, only User 'OPERATIONS' is listed as the common grant for all the methods.

To revoke this common grant, select these three method check boxes first, and then click **Revoke Grant**. This revokes the common grant, User 'OPERATIONS, assigned to these selected methods.

If there is more than one common grant listed in the table, select desired common grants from the table before clicking **Revoke Grant**.

- *Revoking Grants for a Single Procedure and Function or Method*

  In the Grants tab of the interface details page, select a desired method and then click **Revoke Grant**. The Revoke Grants page displays the existing grants that have been created for the selected method.

  Select the grants that you want to revoke from the table, and click **Revoke Grant** to revoke the selected grants.

Viewing Grant Details

Each grant contains information about grantee type, grantee name, and whether the grant is authorized through a direct grant (such as a specific user 'OPERATIONS') or other grant method (such as through a user group 'Marketing Group').

To view grant details, click the **Grant** icon for the method that you want to view. A pop-up window appears with the grant details.

In addition to the Grants tab, you can view the grant details for a desired method from the SOAP Web Service tab and the REST Web Service tab.

To create grants:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility or through custom responsibility and navigation path. Select the Integration Repository link.

2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.

3. Expand an interface type node and click an interface definition (such as a PL/SQL API) that can be exposed as both SOAP and REST services.

The interface details page appears.

4. In the Grants tab, select one or more procedure and function or method names for which you want to create grants.

5. Click **Create Grant**. The Create Grants page appears.

6. Select a grantee type:

   - `Specific User`

   - `Group of Users`

   - `All Users`

7. If you select `Specific User` or `Group of Users`, specify the user or group for which to create the grants in the Grantee Name field.

8. Click **Create Grant**.

   The interface details page reappears.

To view or revoke grants:

You can view and revoke existing grants directly in the methods list on the interface details page.

1. Navigate to the selected PL/SQL API that can be exposed as both SOAP and REST services.

2. To view grant details:

   In the Grants tab, the SOAP Web Service tab, or the REST Web Service tab, click the **Grant** icon for a given operation. A pop-up window appears allowing you to view the grant details for the selected operation.

3. To revoke grants in the Grants tab:

   - To revoke common grants for all selected methods

     Select more than one method from the table and click **Revoke Grant**. The Revoke Grants page appears. Select one or more common grants from the table and click **Revoke Grant**.

   - To revoke grants for a single method

     Select a desired method from the table and then click **Revoke Grant**.

     Select one or more existing grants from the table and click **Revoke Grant** to revoke the grants.

Managing REST Service Life Cycle Activities Using An Ant Script

Similar to SOAP services, the administrator can use an Ant script `$JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml` to execute the design-time activities for REST services such as deploy and undeploy services from command line.

Usage of $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml:

```
ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml usage
```

> **Note:** Script creates log file at the script location; hence, it is suggested to copy `isgDesigner.xml` to some `<TEMP_DIRECTORY>` and then use the script present in `<TEMP_DIRECTORY>`.

Usage Related to Design Activities

1. Enter `ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml`. It will prompt for the arguments.

   > **Note:** Do not enclose any input between double quotes.

2. Enter the arguments in the following ways:

   -
     ```
     ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
     -Dactions=<comma separated list of operations>
     -DserviceType=REST -DirepNames=<func1:SY,func2:SY>
     -Dverbose=<ON|OFF> -Dalias=<Alias>
     ```

     For example:

     -
       ```
       ant -f
       $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
       -Dactions=deploy -DserviceType=REST
       -DirepNames=FND_USER_PKG[TESTUSERNAME:SY] -Dverbose=ON
       -Dalias="FndUserPkgSvc"
       ```

     -
       ```
       ant -f
       $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
       -Dactions=deploy -DserviceType=REST
       -DirepNames=FND_USER_PKG[TESTUSERNAME:SY],FND_MESSAGE[G
       ET_TEXT_NUMBER:SY] -Dverbose=ON
       -Dalias="FndUserPkgSvc,FndMessageSvc"
       ```

     While passing actions and irepNames using this method, be aware of the following conditions:

     - If more than one actions or irepNames are passed as command line argument, enclose them between double quotes. For example,

```
-Dactions="method1, method2,.."
```

```
-DirepNames="FND_USER_PKG[func1:SY::func2:SY::...]"
```

  - If only one action or irepName is passed as command line argument, then there is no need to enclose between double quotes.

-
  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
  -Dfile=<absolute path of service descriptor file>
  -Dverbose=<ON|OFF>
  ```

Argument Description

Valid arguments for `isgDesigner.xml` are described as follows:

- **actions**: Comma separated list of actions to be performed. Supported operations are: deploy, undeploy, and postclone.

  - deploy: It will generate the REST service artifacts and deploy the generated service.

  - undeploy: It will undeploy the deployed service and reset the service status to its initial state - 'Not Deployed'. This also deletes the service artifacts from the Oracle E-Bsuiness Suite managed server.

  Actions 'postclone' should be called independently. This means if postclone action is given, actions argument should look like `-Dactions=postclone`. More information on how action arguments are used is described in the following examples:

  - `-Dactions="deploy,undeploy"`

  - `-Dactions=postclone`

  Additionally, if action is 'postclone', only 'actions' and 'verbose' arguments will be used. However if you have given other arguments as well, only the three arguments mentioned above will be used.

- **serviceType**: [SOAP|REST]: Choose the value REST.

- **irepNames**: Comma separated list of API names.

  For example, `-DirepNames="AR_INVOICE_API_PUB,FND_USER_PKG,EFND_SIGNON"`

- **file**: Absolute path of the (service descriptor) XML file containing interfaces and actions to be performed on these interfaces.

  For example, `-Dfile=/u01/oracle/isg_service.xml`

- **verbose**: [ON|OFF] Default value is OFF.

For example, `-Dverbose=OFF`

- **alias**: It is mandatory for REST services. If multiple services are deployed, use comma separated alias names.

  For example, `-Dalias="FndUserPkgSvc,FndMessageSvc"`

Usage Examples

- Sample command for actions other than 'postclone' (actions and interface names are being passed):

  -
    ```
    ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
    -Dactions=deploy -DserviceType=REST
    -DirepNames=FND_USER_PKG[TESTUSERNAME:SY] -Dverbose=ON
    -Dalias="FndUserPkgSvc"
    ```

  -
    ```
    ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
    -Dactions=deploy -DserviceType=REST
    -DirepNames=FND_USER_PKG[TESTUSERNAME:SY],FND_MESSAGE[GET_
    TEXT_NUMBER:SY] -Dverbose=ON
    -Dalias="FndUserPkgSvc,FndMessageSvc"
    ```

- Sample command for performing design time actions from XML file:

  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
  -Dfile=/u01/oracle/isg_service.xml
  ```

- Sample command for action 'postclone':

  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
  -Dactions=postclone -Dverbose=ON
  ```

Please note that `$JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml` is a multipurpose script. It can be used to manage life cycle activities for SOAP services. See: Managing SOAP Service Life Cycle Activities Using An Ant Script, page 3-27.

You can also use this script to run the diagnostic tests or download the configuration file from the instance.

- Deploying Generic XML Gateway Services, page 3-30
- Downloading the Sample Service Descriptor File, page 3-30
- Obtaining Argument irepNames Usage Information, page 3-32
- Running Diagnostic Tests, page 3-33

Downloading the Sample Service Descriptor File

To download the sample service descriptor file, invoke this script by giving target *filehelp*

```
ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
```

`filehelp`

This downloads one sample service descriptor file and also prints the grammar for service descriptor file. Grammar will look like as following for REST services:

> **Note:** Meaning of the special characters:
>
> - + : One or more occurrence
> - * : Zero or more occurrence
> - ? : Zero or one occurrence

```
<IntegrationRepository name="Instance_SID">
  <services>
    <interface>
      <name>API_Name</name>
<serviceType>REST</serviceType>
      <actions>
        <action>valid action name</action> +
      </actions>
      <!--functions node is necessary if deploy action is given-->
      <functions selective="false" pattern="SY"/> *
      </functions> ?
      <!--policies node is necessary if deploy action is given-->
      <policies>
        <policy>policy_name</policy> +
        <alias>service alias name</alias> +
                 </policies> ?
    </interface> +
  </services>
</IntegrationRepository>
```

Important elements in the descriptor file for REST services are:

- `<serviceType>` node must be present if 'deploy' action is given.
- `<functions>` node must be present if 'deploy' action is given.
- `<policies>` node must be present if 'deploy' action is given.
- `<alias>` node must be present if 'deploy' action is given. It's a unique REST service name for the deployed REST service.
- The 'selective' attribute of `<functions>` node must be false if individual function names are not given.

  If 'selective' attribute of `<functions>` node is false, then user can give one more attribute of this node namely 'pattern'. All function of the corresponding interface will be deployed with this pattern.

  The only supported value for attribute 'pattern' in REST service is **SY** indicating a synchronous pattern.

For example, a REST service descriptor file can be like:

```
<IntegrationRepository name="atgisgqa">
  <services>
    <interface>
      <name>FND_MESSAGE</name>
      <serviceType>REST</serviceType>
      <functions selective="false" pattern="SY"/>
      <actions>
       <action>undeploy</action>
       <action>deploy</action>
      </actions>
      <policies>
       <policy>BASIC</policy>
       <alias>fndMessageSvc</alias>
      </policies>
    </interface>
  </services>
</IntegrationRepository>
```

For information on the SOAP service descriptor file, see Downloading Service Descriptor File for the Current Environment, page 3-30.

4

Administering Composite Services - BPEL

Overview

A composite service is a set of specifications that define a way of assembling SOA-based application. It may consist of one or more services to describe a complex business process requirement. For example, a composite service - BPEL type can be used for service orchestration to manage more complex business processes (such as Order-to-Receipt) which may be handled by various applications.

A composite service - BPEL type contains its own WSDL definition and service endpoints allowing external Web service clients to invoke the services at run time.

Please note that in Oracle SOA Suite 11*g*, BPEL process is managed and deployed together with the associated SOA composite application. In Oracle SOA Suite 10*g*, it is developed and deployed as a separate component. Integration Repository displays 'Composite Services - BPEL' of Oracle SOA Suite 10*g* as catalogue in this release.

This chapter includes the following topics:

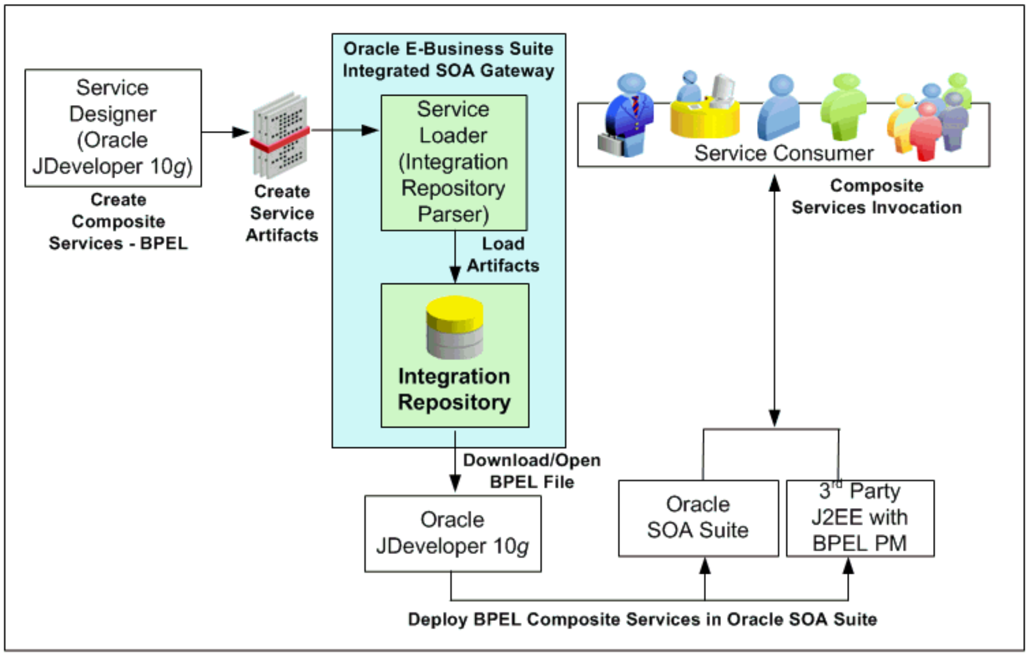
Understanding the Enablement Process for Composite Services - BPEL

To design a composite service, a system integration developer uses BPEL process component in Oracle JDeveloper 10*g* (Service Designer) to assemble a series of service components together for a business function. The newly created composite service - BPEL definition needs to be annotated first based on the Integration Repository annotation standards. Users with the Integration Repository Administrator role need to validate the annotated files using a standalone design time tool called Integration Repository Parser. An Integration Repository loader (iLDT) file is generated after the validation and then uploaded to the Integration Repository using the FNDLOAD command. The composite service - BPEL type can be displayed and searched from the

Integration Repository user interface.

The following diagram illustrates the high level enablement process:

Enablement Process for Composite Services - BPEL


Service Designer (Oracle JDeveloper 10*g*)
Create Composite Services - BPEL
Create Service Artifacts
Oracle E-Business Suite Integrated SOA Gateway
Service Loader (Integration Repository Parser)
Load Artifacts
Integration Repository
Service Consumer
Composite Services Invocation
Download/Open BPEL File
Oracle JDeveloper 10*g*
Oracle SOA Suite
3rd Party J2EE with BPEL PM
Deploy BPEL Composite Services in Oracle SOA Suite

Users granted the download composite service privilege through Integration Repository Download Composite Service Permission Set (FND_REP_DOWNLOAD_PERM_SET) can download the composite - BPEL file to their local directories. A system integration developer can open the downloaded BPEL file using Oracle JDeveloper 10*g* and modify it if necessary before deploying it to a BPEL server in Oracle SOA Suite 10*g* for service consumption.

> **Note:** Composite services - BPEL type is supported in Oracle SOA Suite 10*g*. For example, a composite - BPEL type can be deployed through Oracle JDeveloper to a BPEL server in Oracle SOA Suite 10*g* BPEL Process Manager or a third party BPEL PM in a J2EE environment.

For detailed information on how to upload composite - BPEL definitions to the Integration Repository, see Enabling Custom Integration Interface Process Flow, page 5-2.

For information on Integration Repository annotation standards, see Composite Service - BPEL Annotations, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

Administering Composite Services - BPEL

Oracle E-Business Suite Integrated SOA Gateway allows you to perform the following tasks on composite services:

- Viewing Composite Services - BPEL, page 4-3

  Similar to all other users, integration repository administrators can view composite service - BPEL details, including the abstract WSDL file and BPEL file of the composite service.

- Downloading Composite Services - BPEL, page 4-4

  Apart from viewing the composite service - BPEL details, the administrators can download the .ZIP file for a composite service - BPEL type if it is available for download.

Viewing Composite Services - BPEL

Once annotated custom composite - BPEL definitions are uploaded to the Integration Repository, 'Composite - BPEL' option can be listed when searching by Interface Type and visible to all users.

Integration repository administrators can view composite details for a selected composite service including service name, description, BPEL file, WSDL file, and other annotated information.

To locate a composite service - BPEL, navigate to the Composite Service interface type from the Oracle Integration Repository browser window with View By 'Interface Type' or perform a search by selecting Composite service (such as 'Composite - BPEL') interface type in the Search page. Click your desired composite service name link from the browser tree or the search result to display the composite service - BPEL interface details page where you can:

- View the composite service - BPEL details.

- View the composite service - BPEL abstract WSDL file by clicking the **View Abstract WSDL** link.

- View the BPEL file by clicking the **View BPEL File** link in the BPEL Files region.

- Download a corresponding composite service - BPEL project file to your local directory.

For information on Integration Repository annotation standards, see Composite Service - BPEL Annotations, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

For detailed information on how to upload composite - BPEL definitions to the Integration Repository, see Enabling Custom Integration Interface Process Flow, page 5-

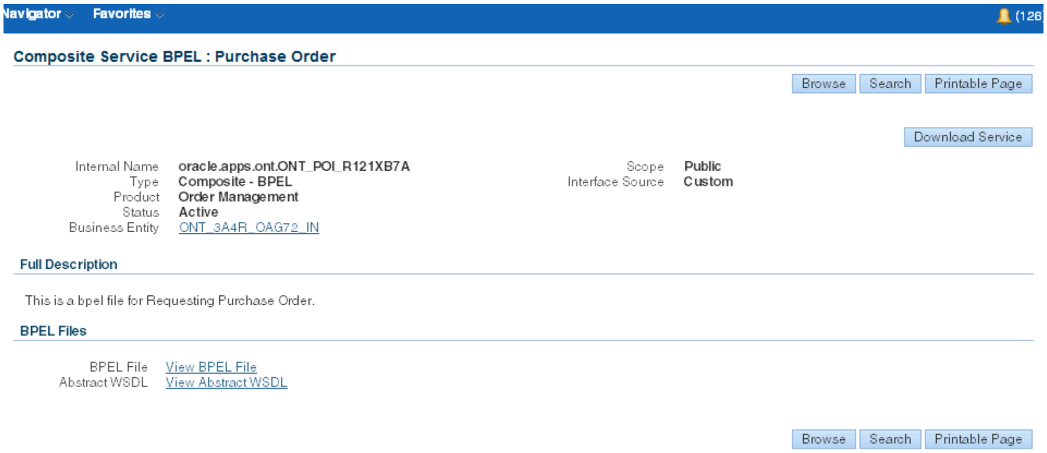
.

Downloading Composite Services - BPEL

In addition to viewing composite service - BPEL details, a WSDL file, and BPEL file, users with the integration repository administrator role can download a BPEL .JAR file containing relevant composite service files to their local machines by clicking **Download Service** in the composite service - BPEL details page.

> **Important:** In general, only users with the system integration developer role and the integration repository administrator role can download the composite services - BPEL. However, users who are granted the download composite service privilege through Integration Repository Download Composite Service Permission Set (FND_REP_DOWNLOAD_PERM_SET) can also perform the download action. Otherwise, **Download Service** may not appear in the details page by default.
>
> For more information about how to grant the download composite service privilege, see Role-Based Access Control (RBAC) Security, page 6-3.

Composite Service - BPEL Details Page with Download Privilege



Composite Service BPEL : Purchase Order

Internal Name oracle.apps.ont.ONT_POI_R121XB7A
Type Composite - BPEL
Product Order Management
Status Active
Business Entity ONT_3A4R_OAG72_IN
Scope Public
Interface Source Custom

Full Description

This is a bpel file for Requesting Purchase Order.

BPEL Files

BPEL File View BPEL File
Abstract WSDL View Abstract WSDL

To download a composite service - BPEL:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility or through custom responsibility and navigation path. Select the Integration Repository link.

2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.

3. Expand the Composite - BPEL interface type node to locate your desired composite service.

4. Click the composite service - BPEL that you want to download it to open the Composite Service- BPEL interface details page.

5. Click **Download Service** to download the selected composite - BPEL file to your local directory.

5

Administering Custom Integration Interfaces and Services

Overview

Oracle E-Business Suite Integrated SOA Gateway supports custom integration interfaces and allows them to be published along with Oracle seeded ones through the Oracle Integration Repository where they can be exposed to all users.

Custom interface definitions can be created for various interface types, including custom interface definitions for XML Gateway Map, Business Event, PL/SQL, Concurrent Program, Business Service Object, Java APIs and Composite Service for BPEL type. Depending on your business needs, system integration developers can create and annotate custom interface definitions based on Integration Repository Annotation Standards. The annotated definitions can then be validated and uploaded to Oracle Integration Repository.

> **Note:** Please note that custom interface types of EDI, Open Interface Tables, Interface Views, and Java APIs for Forms interfaces are not supported in this release.
>
> Oracle Integration Repository currently does not support the creation of custom Product Family and custom Business Entity.

After the upload, these custom integration interfaces are displayed in the Integration Repository based on the interface types they belong to. To easily distinguish them from Oracle integration interfaces, Interface Source "Custom" is used to categorize those custom integration interfaces in contrast to Interface Source "Oracle" for Oracle seeded interfaces in Oracle E-Business Suite. Custom integration interfaces can now seamlessly leverage the Oracle E-Business Suite Integrated SOA Gateway capabilities. Custom integration interfaces of service enabled interface type can be exposed as standard Web service. The administrator performs the same administrative tasks for custom integration interfaces as he or she does for native integration interfaces. These tasks include creating security grants, as well as generating and managing services throughout the deployment life cycle.

Usage Guidelines for Custom Web Services

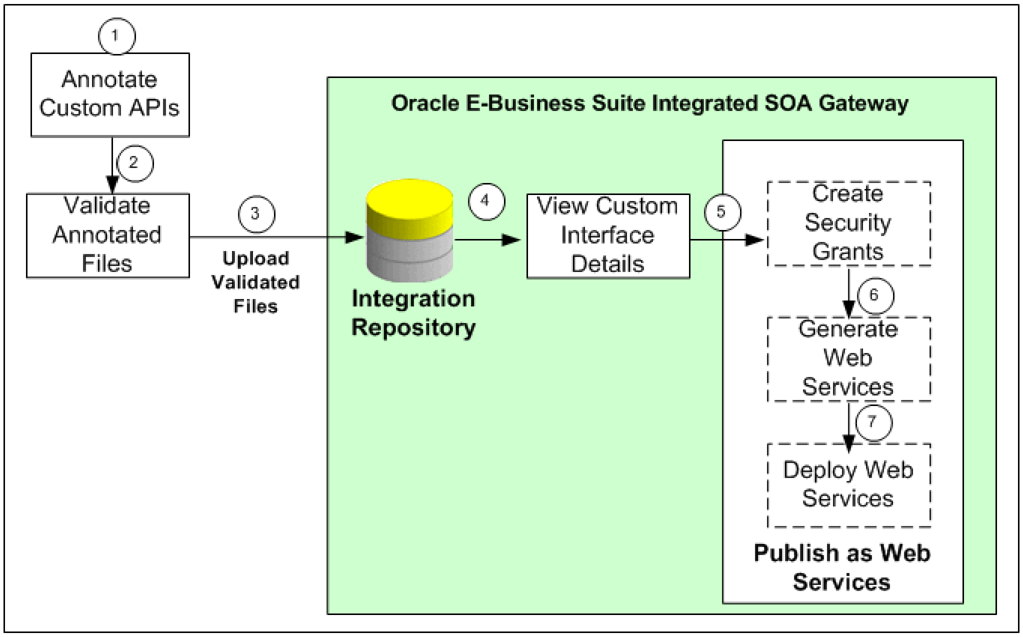
While creating or developing custom Web services for your business needs, consider the following conditions:

Requirement	Use
To enable existing or new Oracle E-Business Suite customizations built on native Oracle E-Business Suite technologies (such as PL/SQL, Business Service Objects, and other supported custom integration interface types described earlier), as Web services	Oracle E-Business Suite Integrated SOA Gateway
To integrate Oracle E-Business Suite with SOA application that requires rich service infrastructure and integration capabilities such as Business Rules, Business Activity Monitoring (BAM), Web service development and orchestration	Oracle SOA Suite in conjunction with Oracle E-Business Suite Integrated SOA Gateway
To develop custom Web services that are not associated with Oracle E-Business Suite	Oracle WebLogic Web service stack

Enabling Custom Integration Interface Process Flow

The following diagram illustrates the entire process flow of enabling custom integration interfaces:

Enabling Custom Integration Interfaces and Services



1. Users with the system integration developer role annotate custom integration interface definition based on the Integration Repository annotation standards for the supported interface types.

   See: Integration Repository Annotation Standards, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

2. Users with the integration repository administrator role validate the annotated custom interface definitions against the annotation standards. This validation is performed by executing the Integration Repository Parser (IREP Parser), a design time tool, to read the annotated files and then generate an Integration Repository loader file (iLDT) if no error occurred. For more information, see:

   - Setting Up and Using the Integration Repository Parser, page 5-5

   - Generating ILDT Files, page 5-9

3. Users with the integration repository administrator role upload the generated iLDT file to Oracle Integration Repository.

   See: Uploading ILDT Files to Integration Repository, page 5-14.

4. All users can view the uploaded custom interfaces from the Integration Repository user interface.

5. (Optional) Users with the integration repository administrator role then create necessary security grants for the custom integration interfaces if needed.

This is achieved by first locating the custom interface from the Integration Repository, and then selecting methods contained in the selected custom interface before clicking **Create Grant**. The Create Grants page is displayed where the administrators can grant the selected method access permissions to a user, user group, or all users.

- For custom interfaces with the support for SOAP services only, security grants are managed in the Methods region of the interface details page. See: Creating Security Grants for SOAP Services Only, page 5-20.

- If the custom interfaces are PL/SQL APIs that can be exposed as both SOAP and REST services, security grants are managed in the Grants tab instead. See: Managing Security Grants for Custom REST Web Services, page 5-22.

6. (Optional) Users with the integration repository administrator role can generate SOAP Web services if the custom interfaces can be service enabled.

   This is achieved by first locating the custom interface, and then specifying the interaction pattern either at the interface level or the method level before clicking **Generate** in the selected custom interface details page. See: Generating Custom SOAP Web Services, page 5-20.

7. (Optional) Users with the integration repository administrator role deploy the services from Oracle Integration Repository to the application server.

   To deploy generated SOAP Web services, the administrators must first select one authentication type (Username Token or SAML Token) for each selected Web service and then click **Deploy** in the selected interface details page. This deploys the generated service with 'Active' state to Oracle SOA Suite where Oracle E-Business Suite services can be exposed as standard Web services for service execution at run time. See: Deploying and Undeploying SOAP Custom Web Services, page 5-20.

   If the custom interfaces are PL/SQL APIs that can also be exposed as REST services, the administrators must enter a unique service alias for each selected custom interface and specify the desired service operations within the interface before deploying the REST service.

   REST services are deployed to an Oracle E-Business Suite environment. For more information on how to deploy custom REST services, see Deploying Custom REST Web Services, page 5-22.

To better understand how to use Integration Repository Parser to validate and upload annotated custom interface definitions to Integration Repository, as well as perform administrative tasks on these uploaded custom integration interfaces, the following topics are discussed in this chapter:

- Setting Up and Using Integration Repository Parser, page 5-5

- Administering Custom Integration Interfaces and Services, page 5-18

For information on how to create and annotate custom integration interfaces, see Creating and Annotating Custom Integration Interfaces, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

Setting Up and Using the Integration Repository Parser

Setup Tasks

Integration Repository Parser is a standalone design-time tool used by the integration repository administrator to validate annotated custom interface definitions against the annotation standards and generate an Integration Repository loader file (iLDT). The generated iLDT files are uploaded to the Integration Repository using the FNDLOAD command so that the custom interfaces can be searched, generated, and deployed from the Integration Repository user interface.

> **Note:** Please note that Integration Repository Parser does not support the integration interfaces registered under custom applications.

Before executing the Integration Repository Parser, you need to install `Perl` modules with the following steps:

> **Note:** It is required to obtain a native C compiler for the platform and operating system version that you are running on to build the `Perl` modules. The following are the minimum versions of compilers certified for Oracle E-Business Suite platforms:
>
> - Linux x86-64: Intel C/C++ Compiler (icc) version 7.1.032
> - Oracle Solaris on SPARC (64-bit): Oracle Studio 12
> - HP-UX Itanium: HP ANSI C B3910B A.0.06.05
> - IBM AIX on Power Systems (64-bit): XL C Enterprise 8.0

Installing Perl Modules on all UNIX platforms

Perform the following steps to install `Perl` modules on all UNIX platforms mentioned above:

1. Establish the Oracle E-Business Suite application environment.

   From the Oracle E-Business Suite APPS_BASE, establish the run file system `APPL_TOP` environment by running the `EBSapps.env` script.

2. In both the run and patch file systems, locate the `Perl` configuration files that need to be modified and back up these files.

For example, on Oracle Solaris, the `Config.pm` is located in the following directory:

```
$FMW_HOME/webtier/perl/lib/5.10.0/sun4-solaris-thread-multi-64
```

3. In both the run and patch file systems, modify the `Perl` configuration file `Config.pm` to point to the `Perl` directory in `$FMW_HOME/webtier`.

   For example, on Oracle Solaris, these are the statements that need to be modified with the absolute path of `$FMW_HOME/webtier/perl`:

   > **Note:** Please note that `<FMW_HOME>` is the value of `$FMW_HOME`.

   - `archlibexp =>relocate_inc('<FMW_HOME>/webtier/perl/lib/5.10.0/sun4-solaris-thread-multi-64')`
   - `privlibexp =>relocate_inc('<FMW_HOME>/webtier/perl/lib/5.10.0')`
   - `sitearchexp =>relocate_inc('<FMW_HOME>/webtier/perl/lib/site_perl/5.10.0/sun4-solaris-thread-multi-64')`
   - `sitelibexp =>relocate_inc('<FMW_HOME>/webtier/perl/lib/site_perl/5.10.0')`

4. If your system is on Oracle Solaris, modify the `Config.pm` and `Config_heavy.pl` files to point to the C compiler installed as a requirement of the Integration Repository Parser. For example:

   Config.pm

   ```
   cc =>'/opt/SunProd/studio12u3/solarisstudio12.3/bin/cc',
   ```

   ```
   libpth =>'/opt/SunProd/studio12u3/solarisstudio12.3/lib
   /opt/SUNWspro/WS6U1/lib/v9 /usr/lib/sparcv9
   /usr/ccs/lib/sparcv9 /usr/local/lib/usr/lib /usr/ccs/lib,
   ```

   Config_heavy.pl

   ```
   cc='/opt/SunProd/studio12u3/solarisstudio12.3/bin/cc'
   ```

   ```
   ld='/opt/SunProd/studio12u3/solarisstudio12.3/bin/cc'
   ```

5. Create a directory '`perl`' in `$APPL_TOP_NE` where the new `Perl` modules will be installed. For example,

   ```
   mkdir $APPL_TOP_NE/perl
   ```

   ```
   chmod 755 $APPL_TOP_NE/perl
   ```

6. In the run file system, set the following environment variables in `APPL_TOP`

environment:

1. Prepend PATH with the path to the C compiler installed as a requirement of the Integration Repository Parser.

2. Prepend PERL5LIB with $FND_TOP/perl and $APPL_TOP_NE/perl in that order.

   For example, export PERL5LIB=$FND_TOP/perl:$APPL_TOP_NE/perl:$PERL5LIB.

3. Add $FMW_HOME/webtier/lib to LIBPATH if it is not present.

   For example, export LIBPATH=$LIBPATH:$FMW_HOME/webtier/lib.

4. Set $FMW_HOME/webtier as ORACLE_HOME.

   For example, export ORACLE_HOME=$FMW_HOME/webtier.

5. Prepend LD_LIBRARY_PATH with $ORACLE_HOME/lib32 and $ORACLE_HOME/lib.

   For example, export LD_LIBRARY_PATH=$ORACLE_HOME/lib32:$ORACLE_HOME/lib:$LD_LIBRARY_PATH.

6. Set JAVA_HOME to the JDK top directory.

   Obtain the path returned by 'which java' and set JAVA_HOME to the current JDK top directory.

   For example, on Oracle Solaris:

   ```
   which java
      /prod/EBS122/fs1/FMW_Home/jdk/jre/bin/java
   export JAVA_HOME=/prod/EBS122/fs1/FMW_Home/jdk
   ```

7. Download and unzip patch 13602850 (p13602850_R12_GENERIC.zip) into a temporary area.

   Patch 13602850 contains the following Perl modules:

   - Compress-Raw-Zlib-2.009
   - Compress-Zlib-2.009
   - Class-MethodMaker-1.12

   Install these modules in the order shown above using the following commands:

   > **Note:** If Perl command is not found, invoke Perl in $FMW_HOME/webtier/perl/bin/perl.

For Oracle Solaris, AIX, and HP-UX Itanium platforms Only

After installing the `Compress-Raw-Zlib-2.009` Perl module but before installing `Compress-Zlib-2.009`, prepend `PERL5LIB` with `$APPL_TOP_NE/perl/lib/5.10.0/<platform thread-multi directory>`.

For example, on Oracle Solaris:

export `PERL5LIB=$APPL_TOP_NE/perl/lib/5.10.0/sun4-solaris-thread-multi-64:$PERL5LIB`.

1. `cd $APPL_TOP_NE/perl`
2. Copy the module to be installed into `$APPL_TOP_NE/perl`.

   For example: `cp -r /temp/Compress-Raw-Zlib-2.009 .`
3. `cd <Perl module name>`

   For example: `cd Compress-Raw-Zlib-2.009`
4. `perl Makefile.PL`

   > **Note:** On HP-UX Itanium, the option `CC=cc` may be needed when installing `Compress-Raw-Zlib-2.009`. For example, `perl Makefile.PL CC=cc`.
   >
   > If errors occur, verify your setup and remove the Perl module being installed from `$APPL_TOP_NE/perl` before copying it into `$APPL_TOP_NE/perl` to try again.
5. `make`
6. `make install`

Using the Integration Repository Parser

Once the Integration Repository Parser has been installed and set up properly, you can execute the parser to generate iLDT files and then upload them to the Integration Repository if no error occurs.

> **Note:** For an object (or class) which is present in the Integration Repository, the Integration Repository Loader program reloads the new definition of that object ONLY if the new version is greater than the current version present in the Integration Repository. If the new file

version is the same or lower than the current one in the repository, then the new file will not be uploaded.

Therefore, before executing the parser, you need to increment the Header version of the target source file so that the modifications to the object defined in the source file can take effect in the Integration Repository.

The following sections explain the use of Integration Repository Parser and FNDLOAD utilities in greater detail.

Generating ILDT Files

To generate an iLDT (`*.ildt`) file, execute the Integration Repository Parser using the following syntax:

```
$IAS_ORACLE_HOME/perl/bin/perl $FND_TOP/bin/irep_parser.pl -g -v
-username=<a fnd username> <product>:<relative path from product
top>:<fileName>:<version>=<Complete File Path, if not in currect
directory>
```

Examples of generating iLDT files for custom PL/SQL APIs and custom composites of BPEL type:

-
  ```
  $IAS_ORACLE_HOME/perl/bin/perl $FND_TOP/bin/irep_parser.pl -g
  -v -username=sysadmin
  fnd:patch/115/sql:SOATest1S.pls:12.0=SOATest1S.pls
  ```

-
  ```
  $IAS_ORACLE_HOME/perl/bin/perl $FND_TOP/bin/irep_parser.pl -g
  -v -username=sysadmin
  fnd:<path>:ONT_POI_R121XB7A.bpel:12.0=<Path>/ONT_POI_R121XB7A
  .bpel
  ```

> **Note:** If an error message "Java runtime not found" appears while executing the Integration Repository Parser, then set the JRE location to variable `OA_JRE_TOP`. JRE location could be located at `$JAVA_HOME/jre`, If `JAVA_HOME` is not set, source `$FMW_HOME/wlserver_10.3/server/bin/setWLSEnv.sh` file.

While executing the parser, you need to pay attention to any error messages on the console. These errors would be due to incorrect annotation or some syntax errors in the annotated file. Ensure that the annotations are correct and the file has proper syntax.

If no error occurs in the annotated interface file, an iLDT (`*.ildt`) file would be generated. This generated iLDT file needs to be uploaded to the Integration Repository.

See: Uploading ILDT Files to Integration Repository, page 5-14.

Integration Repository Parser (irep_parser.pl) Usage Details

The usage for the Integration Repository Parser can be seen from the command prompt using the `-manual` option:

```
$IAS_ORACLE_HOME/perl/bin/perl $FND_TOP/bin/irep_parser.pl
-manual
```

Name `irep_parser.pl` Interface Repository Annotation Processor

Synopsis `irep_parser.pl [-verbose] [-logfile=file ? -append-logfile=file] [-generate] [-force] [-outdir=directory] [-java-source=version] [-cache-java=oper] [-cache-file=file] [-imports=file] [-username=username] <filespec>...`

Description The `irep_parser` reads interface annotation documentation in program source files and validates it according to its file type.

If the `-generate` flag is supplied (and other conditions met), then it will generate iLDT files. For more information, see `-generate` option, page 5-11.

Any validation errors will be reported, usually along with file name and line number, like the result of `grep -n`.

File Types

The `irep_parser` can handle almost all types of application source files. While validating the annotated files against the annotation standards of the supported interface types, files that do not match will be ignored.

Here is the list of supported file types:

> **Note:** Integration Repository Parser supports custom interface definitions for XML Gateway Map, Business Event, PL/SQL, Concurrent Program, Business Service Object, Java APIs, and Composite Service for BPEL type.
>
> Custom interface types of EDI, Open Interface Tables, Interface Views, and Java APIs for Forms interfaces are not supported in this release.

- `.java`: All Java files are completely parsed.
- `.p(kh/ls)`: PL/SQL package specifications are processed.

  If and when a package body is detected, the parser aborts processing and the file is ignored.
- `.ldt`: It processes the LDT file for annotated concurrent programs. Most LDT files will fail and be ignored right away because they are not concurrent program loader files (i.e. not created with `afcpprog.lct`).
- `.xgm`: It processes the XML Gateway map file, looking for an annotated map.

- `.xml`: It processes the XML file, scanning for signature contents indicating various kinds of Business Service Object data since the filename pattern is generic.

- `.wfx`: It processes the Business Event file, looking for annotated events.

Files Specifications

Argument `filespec` tokens have the following formats:

- `pathname`: A simple `pathname` argument directly indicates the file to be processed. Since path information is not included, the output iLDT can not be generated. For example, only validation is supported. See `-development` flag, page 5-12 (This is backward compatible with previous validation only usage.)

- `product:relative_path[:name[:version]]=pathname`: Specify the product and relative path from product top (and optionally file name and version) in addition to the physical location of the file to process.

  Please note that the source file information on the left-hand side of the "=" sign is imported verbatim into the output iLDT, and otherwise not examined. The `pathname` on the right-hand side must refer to a real file, which can be located anywhere.

  The `product` and `relative_path` correspond to file location on `APPL_TOP`.

Options

Options can be abbreviated by the smallest significant number of characters. Often this can be just the first character. Options cannot be combined. Here are the supported options:

- `-generate`: It generates iLDT (Interface Repository Seed Data) files. The file is created in either the current directory or the directory designated by `-outdir`.

  The generated file name is derived from the file name by replacing all periods with underscores, and then appending the suffix "`.ildt`".

  > **Note:** Use of the `-generate` flag requires that the command line filespecs to have (at least) the source product and path. For more information, see `prod:path[:name[:version]]=pathname`, page 5-11 and the `-development` flag, page 5-12.

- `-force`: If the `-generate` flag is used to request iLDT generation, and if the file is an incorrect file type for annotations or has no significant annotation contents (no annotation at all, or no `@rep:scope` tag in any master-level annotation), then an empty file is created anyway. If a file of the same name existed from a previous run, it is forced to be overwritten with a zero-length file.

  The net effect is that only files that had actual errors (parsing, validation, and incomplete for generation) will not be represented in the creation of (at least) in an empty iLDT file.

- `-development`: It is a special flag for developers to quickly verify syntax of annotations in a file. It is equivalent to using both `-generate` and `-verbose` flags with sample values of fields, such as 'product', 'relative path from product top' and 'version'. For example, `-d TestFileName` is equivalent to `-g -v nul:relative/path/unknown:TestFileName:1.0=TestFileName`.

  This allows you to generate test iLDTs using a simple list of filenames.

- `-outdir=directory`: It designates an alternate directory (other than the working directory) for generated output to be placed in.

- `-username=username`: A valid FND username (other than the default SEED username) which marks this interface as custom service.

  If tag `-username` is missed, it is considered as a seeded interface. A custom interface is identified on the Integration Repository user interface by the label 'Custom' and can be searched by selecting 'Custom' in the Interface Source field after clicking **Show More Search Options** in the Search page.

- `-logfile=file`: It writes all verbose tracing and validation error messages in a log file instead of printing to standard output. It is mutually exclusive with `-append-logfile`.

- `-append-logfile=file`: It is similar to `-logfile`, append all verbose tracing and validation error messages in a log file instead of printing to standard output. It is mutually exclusive with `-logfile`.

- `-verbose`: It provides chatty information about files processed and other internals, non-fatal warning messages, and so on. This is in addition to any error messages generated.

  It is useful for querying the parser version, if it's used without any filespec arguments.

- `-java-source=version`: It informs the parser what language version (via JDK version number) to support for Java parses. A minor change was introduced in 1.4 (the assert facility), and major changes were introduced in 1.5 (generics, enhanced for loop, autoboxing/unboxing, enums, varargs, static import and annotations). If it is not supplied, then 1.5 is assumed.

Return Value

The parser will return an exit value of 0 if no errors occurred during processing. Otherwise, it will return a count of the number of files that had errors.

Files with incomplete information for generation (class resolution) are considered errors only if the `-generate` flag is used.

Quick Validation Examples

Use the following statements in validating annotation in PL/SQL specification files during development:

- `$IAS_ORACLE_HOME/perl/bin/perl $FND_TOP/bin/irep_parser.pl *s.pls`
- `$IAS_ORACLE_HOME/perl/bin/perl $FND_TOP/bin/irep_parser.pl -v -g itg:patch/115/sql:12.0=fndav.pls`

Environment

1. Set the Oracle E-Business Suite application environment.

   From the Oracle E-Business Suite APPS_BASE, establish the run file system `APPL_TOP` environment by running the `EBSapps.env` script.

2. The following environment variables affect parser operation:

   - `LIBPATH`: Add the `$FMW_HOME/webtier/lib` to `LIBPATH` variable if it is not present. For example,

     export `LIBPATH=$LIBPATH:$FMW_HOME/webtier/lib`

   - `CLASSPATH`: It is used when parsing Java files. This is required to be properly set up (as if for a compile) when performing `-generate` with such files.

     If parser is not able to find a particular class, check for its availability in `CLASSPATH`.

     On a Linux machine, `CLASSPATH` can be set like `setenv CLASSPATH classpath1:classpath2`.

     For others, refer to your platform documentation on how to set `classpath` variable.

   - `JAVA_HOME`: It is used to find the Java runtime.

     If `JAVA_HOME` is not set, obtain the path returned by 'which java' from the `APPL_TOP` environment, and set `JAVA_TOP` to the JDK top directory. For example,

     - On AIX:

       export `JAVA_HOME=$COMMON_TOP/util/jdk32`

     - On Oracle Solaris:

       export `JAVA_HOME=$COMMON_TOP/util/jdk`

   - export `PERL5LIB=$APPL_TOP_NE/perl/lib/5.10.0:$APPL_TOP_NE/perl/lib/site_perl/5.10.0:$FND_TOP/perl:$PERL5LIB`

Uploading ILDT Files to Integration Repository

After validation is completed and iLDT files are generated, the administrator can upload the generated iLDT files to the Integration Repository using the FNDLOAD command. The custom interfaces can be displayed in the repository and exposed to all users.

Manual Steps for Uploading the iLDT File

Perform the following steps to upload the iLDT file to the Integration Repository:

1. Log on to the Oracle E-Business Suite Release 12 instance.

2. Set the Oracle E-Business Suite application environment.

   From the Oracle E-Business Suite APPS_BASE, establish the run file system `APPL_TOP` environment by running the `EBSapps.env` script.

   For HP-UX Itanium Only:

   Prepend `LD_LIBRARY_PATH` with `$FMW_HOME/webtier/lib` as follows:

   export `LD_LIBRARY_PATH=$FMW_HOME/webtier/lib:$LD_LIBRARY_PATH`

3. Use the following command to upload the iLDT file:

   ```
   $FND_TOP/bin/FNDLOAD <db_connect> 0 Y UPLOAD
   $fnd/patch/115/import/wfirep.lct <ildt file>
   ```

   Examples of uploading iLDT files for custom PL/SQL APIs and custom composites of BPEL type:

   - `$FND_TOP/bin/FNDLOAD apps/`*`password`*`@isg122d 0 Y UPLOAD $FND_TOP/patch/115/import/wfirep.lct SOATest1S_pls.ildt`

   - `$FND_TOP/bin/FNDLOAD apps/`*`password`*`@$TWO_TASK 0 Y UPLOAD $FND_TOP/patch/115/import/wfirep.lct ./ONT_POI_R121XB7A_bpel.ildt`

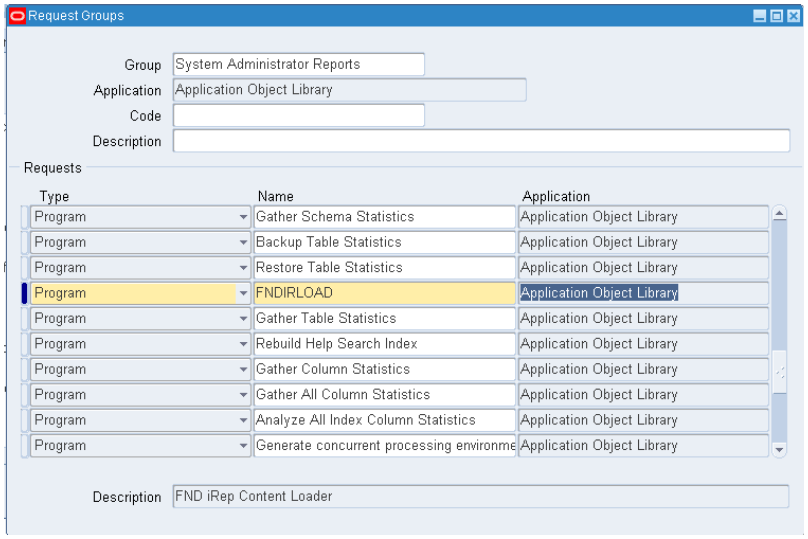
4. Pay attention to any error messages in the generated log file. Error messages mostly would be due to incorrect database connect string or incorrect `lct` file.

   Look for string "Concurrent request completed successfully" to determine whether the iLDT file was correctly uploaded.

5. For Business Service Object only - submit a concurrent program called FNDIRLOAD which loads all the iLDT files related to Business Service Object interfaces present on various product tops of the instances.

   > **Note:** Ensure that FNDIRLOAD concurrent program is associated with the user who will execute the concurrent request.

For example, if it will be run by a user with the system administrator responsibility, FNDIRLOAD should be listed as part of the requests for System Administrator Reports group in the Request Groups window.



Group: System Administrator Reports
Application: Application Object Library
Code:
Description:

Requests

Type	Name	Application
Program	Gather Schema Statistics	Application Object Library
Program	Backup Table Statistics	Application Object Library
Program	Restore Table Statistics	Application Object Library
Program	FNDIRLOAD	Application Object Library
Program	Gather Table Statistics	Application Object Library
Program	Rebuild Help Search Index	Application Object Library
Program	Gather Column Statistics	Application Object Library
Program	Gather All Column Statistics	Application Object Library
Program	Analyze All Index Column Statistics	Application Object Library
Program	Generate concurrent processing environme	Application Object Library

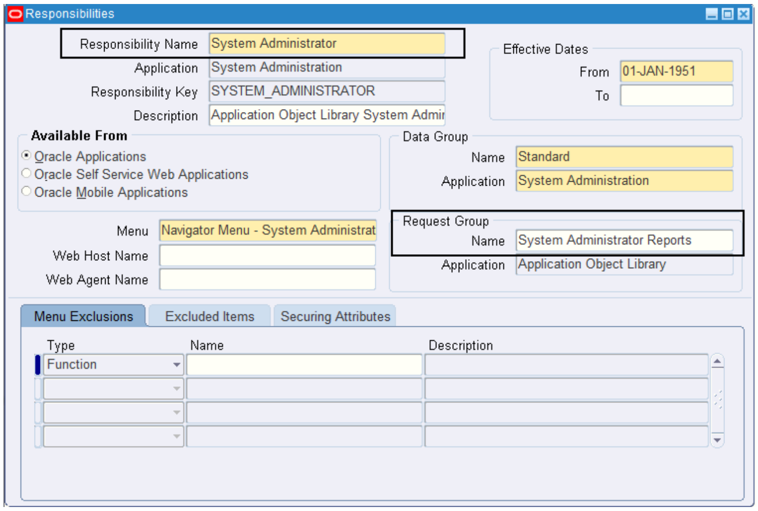
Description: FND iRep Content Loader

If you cannot find FNDIRLOAD from the name list, use the following steps to register it with the system administrator responsibility.

1. Log on to Oracle E-Business Suite with the System Administrator responsibility. Select **System Administrator > Security > Responsibility > Define** from the navigation menu.

2. In the Responsibilities window, locate 'System Administrator' as the value in the Responsibility Name field through a search.

   Ensure 'System Administrator Reports' is selected as the Request Group Name.



Responsibilities

Responsibility Name: System Administrator

Application: System Administration

Responsibility Key: SYSTEM_ADMINISTRATOR

Description: Application Object Library System Admir

Effective Dates

From: 01-JAN-1951

To:

Available From

● Oracle Applications

○ Oracle Self Service Web Applications

○ Oracle Mobile Applications

Data Group

Name: Standard

Application: System Administration

Menu: Navigator Menu - System Administrat

Web Host Name:

Web Agent Name:

Request Group

Name: System Administrator Reports

Application: Application Object Library

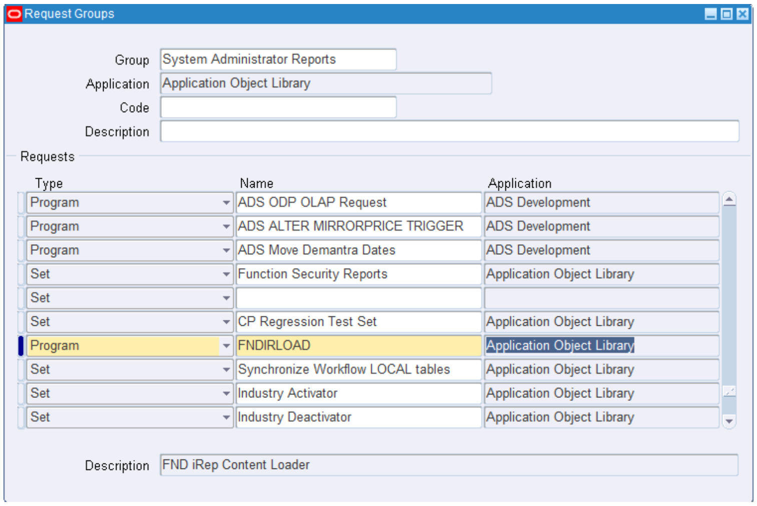
Type	Name	Description
Function		

Save the change and close the window.

3. Select **System Administrator** > **Security** > **Responsibility** > **Requests** from the navigation menu.

   In the Request Group window, locate 'System Administrator Reports' as the value in the Group field through a search.

   In the Requests region, add FNDIRLOAD program to the list and save your entry.



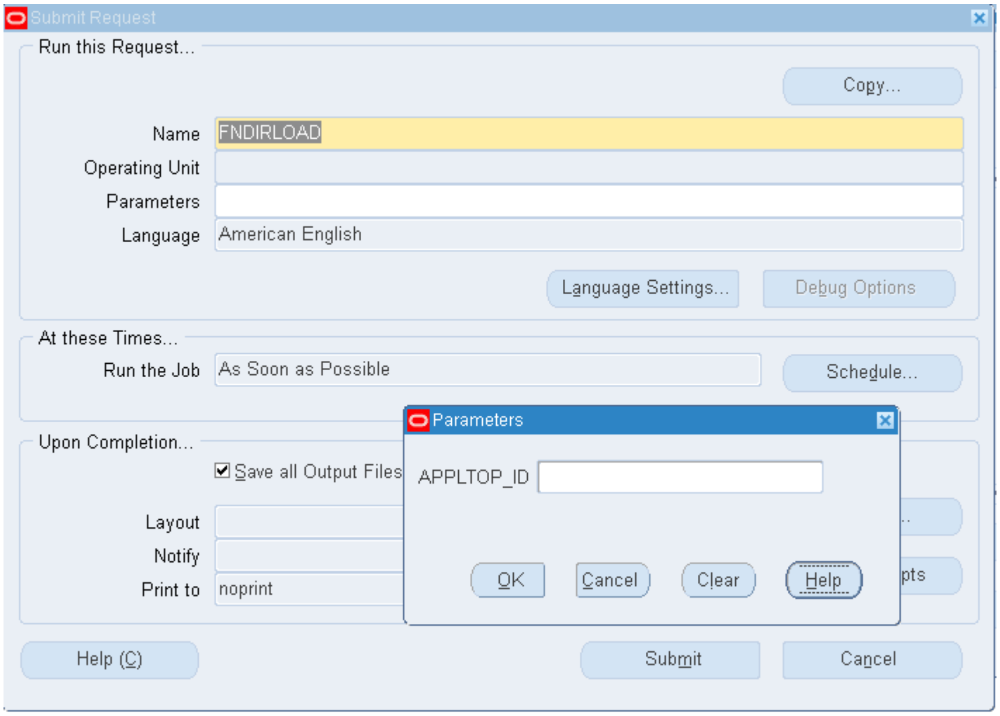
Group	System Administrator Reports
Application	Application Object Library
Code	
Description	

Requests

Type	Name	Application
Program	ADS ODP OLAP Request	ADS Development
Program	ADS ALTER MIRRORPRICE TRIGGER	ADS Development
Program	ADS Move Demantra Dates	ADS Development
Set	Function Security Reports	Application Object Library
Set		
Set	CP Regression Test Set	Application Object Library
Program	FNDIRLOAD	Application Object Library
Set	Synchronize Workflow LOCAL tables	Application Object Library
Set	Industry Activator	Application Object Library
Set	Industry Deactivator	Application Object Library

Description: FND iRep Content Loader

In the Parameters window, enter an appropriate value for APPLTOP_ID.



Name	FNDIRLOAD
Operating Unit	
Parameters	
Language	American English
Run the Job	As Soon as Possible

☑ Save all Output Files

Layout	
Notify	
Print to	noprint

APPLTOP_ID	

Note: To obtain the APPLTOP_ID parameter value, your system administrator can execute the following query:

```
SELECT max(appl_TOP_id)
FROM ad_appl_tops
WHERE active_flag = 'Y'
```

Click **Submit** to execute the request.

Examine the request log file to see if any issues occur while executing the concurrent request.

Once these annotated source files have been successfully uploaded, they will appear in the Integration Repository based on the interface types they belong to. The administrators can perform administrative tasks on these custom integration interfaces including generating, deploying, or undeploying Web services.

Administering Custom Integration Interfaces and Services

Custom integration interfaces are annotated based on Integration Repository annotation standards for the supported interface types. The behavior of these interfaces is the same as Oracle seeded interfaces except they are not native packaged, but custom ones. As a result, an integration repository administrator uses the same approach of managing native interfaces to manage custom interfaces and services.

These administrative tasks include:

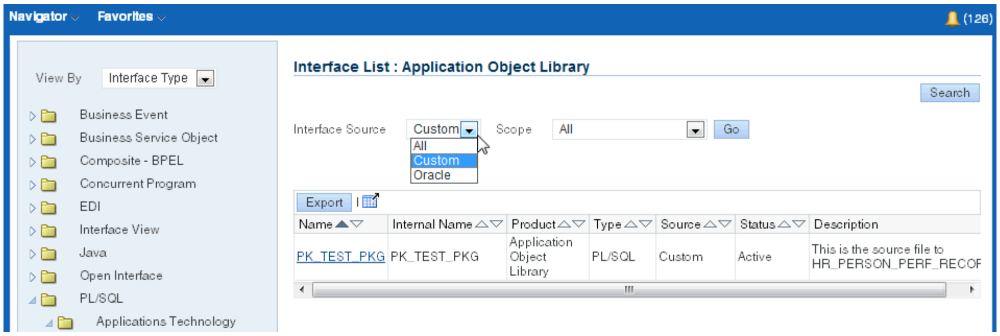
- **For Custom Integration Interfaces with Support for SOAP Web Services**
  - Creating Security Grants for SOAP Services Only, page 5-20
  - Generating Custom SOAP Web Services, page 5-20
  - Deploying and Undeploying Custom SOAP Web Services, page 5-20
  - Resetting Custom SOAP Web Services, page 5-21
  - Retiring Custom SOAP Web Services, page 5-21
  - Activating Custom SOAP Web Services, page 5-21
  - Subscribing to Custom Business Events, page 5-21
- **For Custom Integration Interfaces with Support for REST Web Services**
  - Deploying Custom REST Web Services, page 5-22
  - Undeploying Custom REST Web Services, page 5-22
  - Managing Security Grants for Custom REST Web Services, page 5-22
- **For Custom Composite Integration Interface**
  - Viewing and Downloading Custom Composite Services, page 5-23

Viewing Uploaded Custom Integration Interfaces from the Integration Repository

Use the following ways to locate custom interfaces:

- From the Interface List page, select 'Custom' from the Interface Source drop-down list along with a value for the Scope field to restrict the custom integration interface display. The search criteria 'Oracle' in the drop-down list is used for searching seeded interfaces.

Viewing from Interface List Page



View By: Interface Type

Interface List : Application Object Library

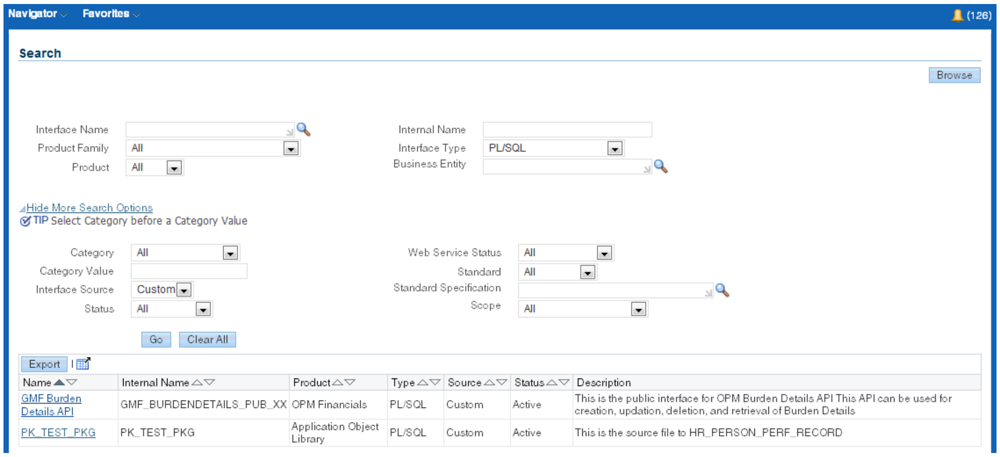
Interface Source: Custom Scope: All

Name	Internal Name	Product	Type	Source	Status	Description
PK_TEST_PKG	PK_TEST_PKG	Application Object Library	PL/SQL	Custom	Active	This is the source file to HR_PERSON_PERF_RECOF

- From the Search page, click **Show More Search Options** and select 'Custom' from the Interface Source drop-down list along with any interface type, product family, or scope if needed as the search criteria.

  For example, select 'Custom' as the Interface Source and 'PL/SQL' as the Interface Type to locate the custom interfaces for PL/SQL type.

Viewing from Interface Search Page



Search

Interface Name:
Product Family: All
Product: All
Internal Name:
Interface Type: PL/SQL
Business Entity:

Hide More Search Options
TIP Select Category before a Category Value

Category: All
Category Value:
Interface Source: Custom
Status: All
Web Service Status: All
Standard: All
Standard Specification:
Scope: All

Name	Internal Name	Product	Type	Source	Status	Description
GMF Burden Details API	GMF_BURDENDETAILS_PUB_XX	OPM Financials	PL/SQL	Custom	Active	This is the public interface for OPM Burden Details API This API can be used for creation, updation, deletion, and retrieval of Burden Details
PK_TEST_PKG	PK_TEST_PKG	Application Object Library	PL/SQL	Custom	Active	This is the source file to HR_PERSON_PERF_RECORD

For more information on how to search for custom integration interfaces, see the *Oracle E-Business Suite Integrated SOA Gateway User's Guide*.

Creating Security Grants for SOAP Services Only

To let appropriate users use these newly uploaded custom integration interfaces, the administrators can select one or more methods contained in a given custom interface and then grant the selected method access permissions to a user, user group, or all users.

For interfaces with the support for SOAP services only, security grants are managed in the Methods region of the interface details page. For more information about managing grants for interfaces with the support for SOAP services only, see Managing Security Grants for SOAP Web Services Only, page 3-21.

Generating Custom SOAP Web Services

Once custom integration interfaces have been uploaded to Oracle Integration Repository, an integration repository administrator or a system integration developer can transform these interface definitions into WSDL descriptions if the interface types they belong to can be service enabled.

To generate a Web service, the administrator must first locate a custom interface, and then specify the interaction pattern either at the interface level or the method level before clicking **Generate** in the interface details page.

If the Web service has been successfully generated, a WSDL link appears along with the 'Generated' Web service status information displayed in the Web Service region (or the SOAP Web Service tab if the interface can be exposed as both SOAP and REST services). The selected interaction pattern information ('Synchronous', 'Asynchronous', or both Synchronous and Asynchronous) for the selected custom service is also displayed.

For detailed information on how to generate SOAP services on native integration interfaces, see Generating SOAP Web Services, page 3-4.

Deploying and Undeploying Custom SOAP Web Services

Once a Web service has been successfully generated for a custom interface, like native packaged interfaces, the administrator will perform the same deployment activity to deploy the generated service to an Oracle SOA Suite WebLogic environment with Active state. Before deploying the custom service, the administrator must select one authentication type to authenticate the Web service.

The administrator can undeploy the service if needed.

> **Note:** Similar to the native Oracle E-Business Suite services, the deployed WSDL URL for the custom service shows the physical location of service endpoint where the service is hosted in `soa-infra` in this release. If your system is upgraded from a previous Oracle E-Business Suite release, after the upgrade to Release 12.2, the deployed

WSDL URL information for the custom service has already been changed. Therefore, you may need to replace it with the new WSDL URL and service location or address accordingly in Web service clients while invoking the deployed custom service.

For detailed information on how to deploy or undeploy SOAP Web services, see Deploying and Undeploying SOAP Web Services, page 3-10.

Resetting Custom SOAP Web Services

Once a custom service has been successfully generated or deployed, **Reset** appears in the Web Service region (or the SOAP Web Service tab if the interface can be exposed as both SOAP and REST services) allowing you to reset the 'Generated' or 'Deployed' Web service status to its initial state - 'Not Generated' if needed. This feature clears up the custom service artifact for a given service regardless of its current state.

For more information, see Resetting SOAP Web Services, page 3-15.

Retiring Custom SOAP Web Services

When a custom service has been successfully deployed to Oracle SOA Suite with active state, this deployed custom service is ready to accept new requests.

The administrator can change the active state of a deployed custom service by clicking **Retire** in the Web Service region (or the SOAP Web Service tab if the interface can be exposed as both SOAP and REST services). This retires a deployed custom service and it will no longer accept new requests.

For a retired custom service, the administrator can activate the retired service so that it can become active again.

For more information on retiring SOAP Web services, see Retiring SOAP Web Services, page 3-17.

Activating Custom SOAP Web Services

For a custom service that has been retired, you can activate it by clicking **Activate** in the interface details page. This action allows a retired custom service to become active again.

For more information on activating Web services, see Activating SOAP Web Services, page 3-18.

Subscribing to Custom Business Events

Similar to the native business events, the administrator can subscribe to a custom business event by clicking **Subscribe** from the business event interface details page. Internally, an event subscription is created for that selected event with

WF_BPEL_QAGENT Out Agent.

Once an event subscription for that custom event has been successfully created, **Unsubscribe** appears instead. Clicking **Unsubscribe** removes the event subscription from the WF_BPEL_Q queue.

For more information on subscribing to business events, see Subscribing to Business Events, page 3-20.

Deploying Custom REST Web Services

After custom interfaces that can be exposed as REST services are uploaded to the Integration Repository, the administrator or a system integration developer can deploy the custom REST services.

Before deploying a custom interface, users with the Integration Repository Administrator role must select one or more methods from the Service Operations table that will be deployed as REST service operations, and then click **Deploy** in the REST Web Service tab for the selected interface. If the service has been successfully deployed, the REST Service Status field is updated to 'Deployed' from 'Not Deployed' indicating that the deployed REST service is ready to accept new service requests.

For more information on deploying REST services, see Deploying REST Web Services, page 3-34.

Undeploying Custom REST Web Services

If a custom REST service has been successfully deployed to an Oracle E-Business Suite managed server, **Undeploy** appears in the REST Web Service tab. Undeploying a REST service not only brings the deployed REST service back to the Integration Repository, but also resets its status to its initial state - 'Not Deployed'.

For more information on undeploying REST services, see Undeploying REST Web Services, page 3-39.

Creating Security Grants for Custom REST Services

Similar to managing grants for the interfaces with the support for SOAP services only, the administrators can create grants by selecting one or more methods contained in a given custom interface and then grant the selected method access permissions to a user, user group, or all users. However, for interfaces, such as PL/SQL APIs, with the support for both SOAP and REST services, security grants are managed in the Grants tab instead.

Once an access permission to a procedure is authorized to a grantee, it grants the permission to access the associated SOAP and REST service operations simultaneously. For more information about managing grants for interfaces with the support for both SOAP and REST services, see Managing Security Grants for SOAP and REST Web Services, page 3-41.

Viewing and Downloading Custom Composite Services

Viewing Custom Composite Services

To view a custom composite service, from the Search page select 'Composite' from the Interface Type field. Click **Show More Search Options** and select 'Custom' from the Interface Source drop-down list along with any product family or scope as the search criteria.

click a custom composite service from the search result to display the composite service details.

Downloading Custom Composite Services

The administrators can click **Download Service** in the interface details page to download the relevant custom composite files aggregated in a .JAR file to your local directory.

For more information on how to view and download a composite service, see:

- Viewing Composite Services - BPEL, page 4-3
- Downloading Composite Services - BPEL, page 4-4

6

Securing Web Services

Overview

Security is the most critical feature that is designed to guard service content from unauthorized access.

To ensure secure access to Web service content, Oracle E-Business Suite integrated SOA Gateway uses the following security models to authenticate and authorize users to invoke a specific service or operation:

- Function Security and Data Security, page 6-1
- Role-Based Access Control (RBAC) Security, page 6-3
- Multiple Organization Access Control Security (MOAC Security), page 6-5
- WS-Service Security (Web Service Security), page 6-8

Managing Function Security and Data Security

By leveraging Oracle User Management function security and data security, Oracle E-Business Suite Integrated SOA Gateway provides a security feature which allows authorized users to invoke certain methods of an integration interface exposed through Oracle Integration Repository. This protects application data from unauthorized access or execution of the Java methods or functions within an API.

Function security is the basic access control in Oracle E-Business Suite. It restricts user access to individual menus and menu options within the system. Regardless of the interface types, APIs enable you to insert and update data in Oracle E-Business Suite. When an API has the function security layer enforced, it implicitly restricts user access to the application.

Building on function security, data security provides another layer of security control. In other words, data security further restricts user access to the application at the data level.

To allow users with appropriate privileges to execute certain methods within an API, the concept of security grant is used to reinforce the security. This approach enables the data access privileges to be granted to a user, user group, or all users. To accomplish this goal, an integration repository administrator can select one or more methods contained in an API and then grant the selected method(s) to users.

An integration repository administrator can create security grants in the following ways:

- If an interface has only one method, then this single method should be selected in creating security grants.

  Concurrent Program and XML Gateway interfaces contain only one method.

- If there is more than one method contained in an interface, then multiple methods can be selected simultaneously in creating security grants.

  Interface types containing multiple methods are PL/SQL, Business Service Object, and Java interfaces.

> **Note:** For PL/SQL interfaces that can be service enabled with the support for both synchronous and asynchronous interaction patterns, the security grants given for the selected method names in the Procedures and Functions region for a PL/SQL interface would be applicable to the generated synchronous and asynchronous operations of the service if both interaction patterns are selected during service generation.

To create and revoke a security grant

For example, in the PL/SQL interface details page, select one or more method name check boxes in the Methods region and click **Create Grant**. The Create Grants page is displayed where you can select a grantee type and grantee name to create the security grants.

To revoke a grant, in the interface details page select the **Show** link for the method that you want to revoke the grant. The Grant Details section of the selected method name appears detailing the grantee and grantee type information. Click the **Revoke** icon for the grantee that you want to revoke the method access permission.

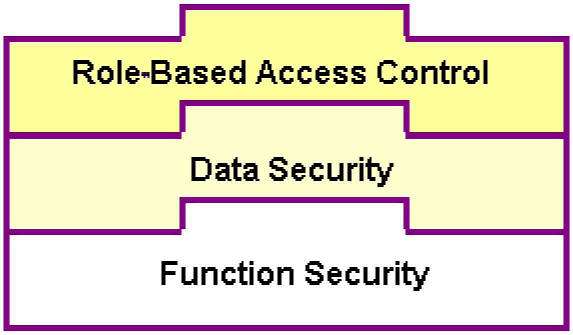
For information on how to create, view, and revoke security grants, see Managing Security Grants, page 3-21. For more information on function security and data security, refer to the Oracle Application Object Library Security chapter, *Oracle E-Business Suite Security Guide*.

Managing Role-Based Access Control Security

To allow only authorized users to perform certain administrative tasks, Oracle E-Business Suite Integrated SOA Gateway leverages Oracle User Management Role-Based Access Control (RBAC) security to build another layer of security. This RBAC security is enforced through user roles. As a result, whether a user can perform certain tasks, such as downloading a composite service from the application server, is determined by the roles granted to the user.

This approach builds upon Data Security and Function Security, but it goes beyond both of them.

Role-Based Access Control Security



As described earlier, function security is the base layer of access control in Oracle E-Business Suite. It restricts user access to individual menus and menu options within the system, but it does not restrict the access to the data contained within those menus. Data security provides access control on the application data, and the actions a user can perform on the data.

With RBAC, access control is defined through roles, and a role can be configured to consolidate the responsibilities, permissions, permission sets, and function security policies that users require to perform a specific function. This simplifies mass updates of user permissions because changes can be done through roles which will inherit the new sets of permissions automatically. Based on the job functions, each role can be assigned a specific permission or permission set if needed. For example, an organization may include 'Analyst', 'Developer', and 'Administrator' roles. The 'Administrator' role would include a permission set that contains all administrative related tasks or functions allowing the administrator role to perform a job function while the Analyst and Developer roles may not have the access privileges.

Gateway

In Oracle E-Business Suite Integrated SOA Gateway, each administrative function is considered as a permission. Relevant permissions are grouped into a permission set that will then be associated with appropriate function roles and assigned to appropriate users through security grants.

Oracle E-Business Suite Integrated SOA Gateway uses the following seeded permission sets to restrict administrative privileges only to authorized users:

- Integration Repository Administrator Permission Set (FND_REP_ADMIN_PERM_SET)
- Integration Repository Download Composite Service (FND_REP_DOWNLOAD_PERM_SET)

Integration Repository Administrator Permission Set

The Integration Repository Administrator Permission Set (FND_REP_ADMIN_PERM_SET) contains almost all administrative tasks performed by the Integration Repository Administrator role. It consists of the following administrative permissions:

Integration Repository Administrator Permission Set

Privilege	Permission	Permission Display Name
Generate/Regenerate	FND_REP_GENERATE	Generate Web Service
Deploy	FND_REP_DEPLOY	Deploy Web Service
Undeploy	FND_REP_UNDEPLOY	Undeploy Web Service
Subscribe to Agent	FND_REP_SUBSCRIBE	Subscribe to Agent
Create Grants	FND_REP_METHOD_GRNT	Grant execute privileges to methods

Integration Repository Download Composite Service Permission Set

Users with an appropriate privilege can download composite services and that privilege is associated with a permission set called Integration Repository Download Composite Service Permission Set (FND_REP_DOWNLOAD_PERM_SET) which is separated from the Integration Repository Administrator Permission Set described earlier. This approach allows the download feature to be granted separately to users through the Integration Repository Administrator role, System Integration Developer role, or System Integration Analyst role if necessary.

Integration Repository Download Composite Service Permission Set

Privilege	Permission	Permission Display Name
Download Composite Service	FND_REP_DOWNLOAD_CS	Download Composite Service

Managing MOAC Security

Multiple organizations can be sets of books, business groups, legal entities, operating units, or inventory organizations. You can define multiple organizations and the relationships between them in a single installation of Oracle E-Business Suite.

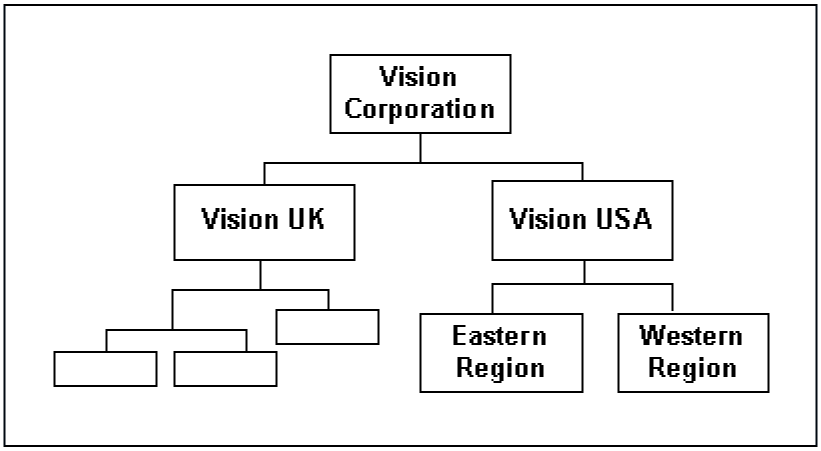
Oracle E-Business Suite Integrated SOA Gateway leverages the MOAC security feature to ensure that only authorized users have data access privilege within an operating unit.

With MOAC, a system administrator can predefine the scope of access privileges as a security profile, and then use the profile option *MO: Security Profile* to associate the security profile with a responsibility. By using this approach, multiple operating units are associated with a security profile and the security profile is assigned to a responsibility. Therefore, through the access control of security profiles, users can access to data in multiple operating units without changing responsibilities.

For example, a sales company consists of USA and UK operating units; the USA operating unit has Western Region Sales and East Region Sales. Sales managers are responsible for both USA and UK sales. Supervisors are responsible for either USA or UK. Sales representatives are only responsible for their designated sales regions.

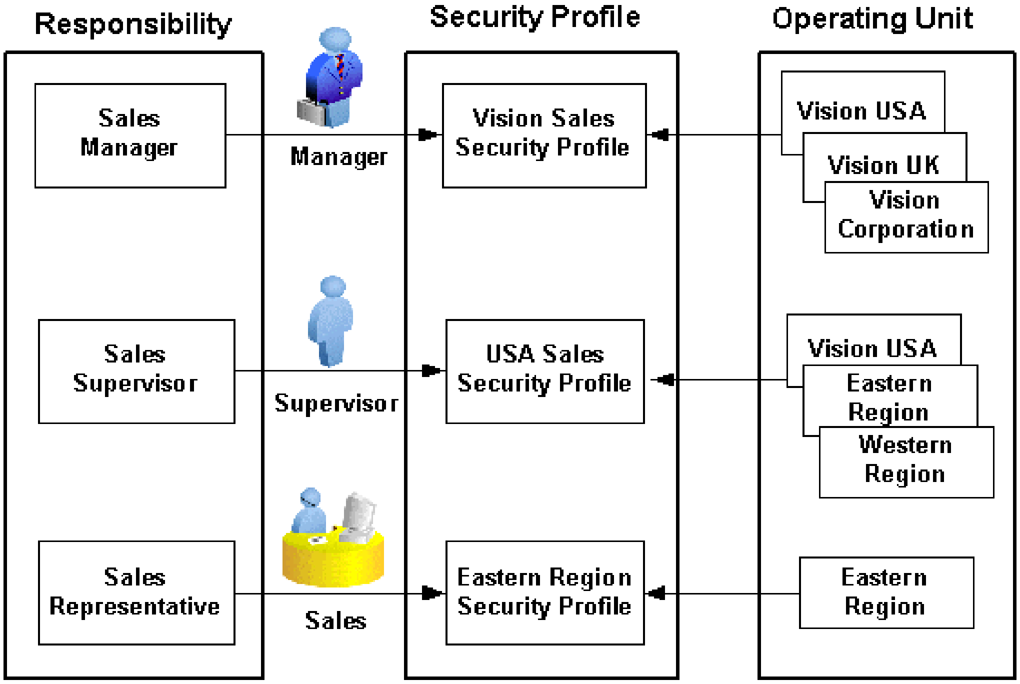
The following diagram illustrates the Sales organization hierarchy:

Sales Organization Hierarchy



To secure sales data within the company, relevant operating units can be associated with predefined security profiles. For example, all sales data access privileges are grouped into the Vision Sales security profile. A USA Sales security profile is for USA related data, and a regional security profile is for designated regional data. The system administrator can associate these security profiles containing multiple operating units with users through appropriate *responsibilities*. Therefore, sales supervisors can easily access sales data in the Eastern or Western region without changing their responsibilities. The following diagram illustrates the relationship between security profiles, responsibilities, and operating units for this sales company:

Relationship Diagram Between Security Profiles, Responsibilities, and Operating Units



Responsibility Determines Operating Units

Because responsibilities are associated with security profiles that are linked to operating units, your responsibility is the key to determine which operating units you will have the access privileges.

1. When integrating with Oracle E-Business Suite using PL/SQL, Concurrent Program, and Java APIs for Forms interfaces, applications context values passed in `SOAHeader` elements for SOAP requests are Responsibility, RespApplication, SecurityGroup, NLSLanguage, and Org_Id.

   For integrating with Oracle E-Business Suite using Business Service Object interfaces, applications context values passed in `ServiceBean_Header` elements for SOAP requests are RESPONSIBILITY_NAME, RESPONSIBILITY_APPL_NAME, SECURITY_GROUP_NAME, NLS_LANGUAGE, and ORG_ID.

2. MOAC setup is done based on the RespApplication or RESPONSIBILITY_APPL_NAME for Business Service Object interfaces to which the user belongs. If Org_Id is passed, the Organization access would be set to the passed Organization.

3. If the NLS Language element is specified, SOAP requests can be consumed in the language passed. All corresponding SOAP responses and error messages can also be returned in the same language. If no language is identified, then the default

language of the user will be used.

For more information on multiple organizations setup and implementation, see the *Oracle E-Business Suite Multiple Organizations Implementation Guide*.

Managing Web Service Security

Web service security (WS-Security) is a specification to enable applications to conduct secure message exchanges. It proposes a standard set of extensions that can be used when building secure Web services to implement message content integrity and confidentiality. It also provides support for multiple security tokens, the details of which are defined in the associated profile documents.

To secure Web service content and authenticate Web service operation, Oracle E-Business Suite Integrated SOA Gateway supports the following authentication security models for inbound service requests:

- For SOAP Services

  - UsernameToken Based Security, page 6-10

  - SAML Sender-Vouches Token Based Security, page 6-12

  At design time, an integration repository administrator must select one authentication type before deploying a service. If no authentication type is identified for the service, then a validation error occurs.

  If the authentication type of a deployed SOAP service needs to be changed, the administrator must first undeploy the SOAP service, make appropriate changes, regenerate the SOAP service, and then deploy it again. For more information on how to deploy and undeploy SOAP services, see: Deploying and Undeploying SOAP Web Services, page 3-10.

- For REST Services

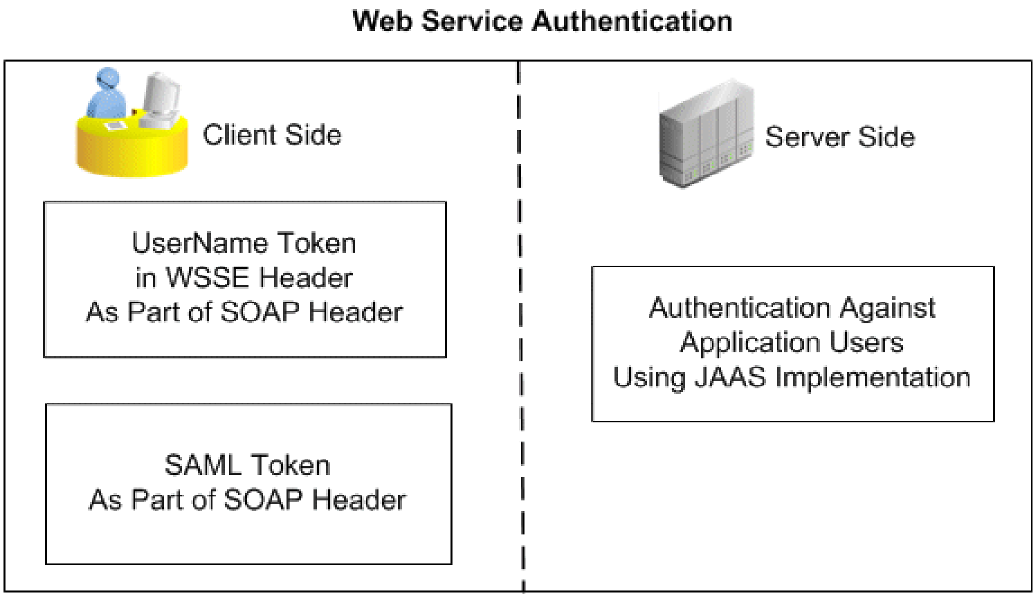
  - HTTP Basic Authentication, page 6-16

  - Token Based Authentication, page 6-17

  All REST services are secured by either HTTP Basic Authentication (username and password) or Token Based Authentication (username and a valid token, such as Oracle E-Business Suite session ID).

Subject Authentication to Establish User's Identity

At run time, when SOAP requests are received through Oracle SOA Suite for the deployed SOA Composites in an Oracle WebLogic managed server, each message is authenticated, depending on the selected authentication type, by a JAAS (Java Authentication and Authorization Service) based login module for Oracle E-Business Suite.


Web Service Authentication
Client Side
UserName Token in WSSE Header As Part of SOAP Header
SAML Token As Part of SOAP Header
Server Side
Authentication Against Application Users Using JAAS Implementation

> **Note:** JAAS (Java Authentication and Authorization Service) is a Java security framework that can be used for authentication of users (user login) to securely determine who is currently executing Java code, and for authorization of users to ensure they have appropriate access control privileges required to access or perform certain operations.
>
> To authenticate users, the JAAS based login module for Oracle E-Business Suite will be deployed into the WebLogic server containing Oracle SOA Suite.

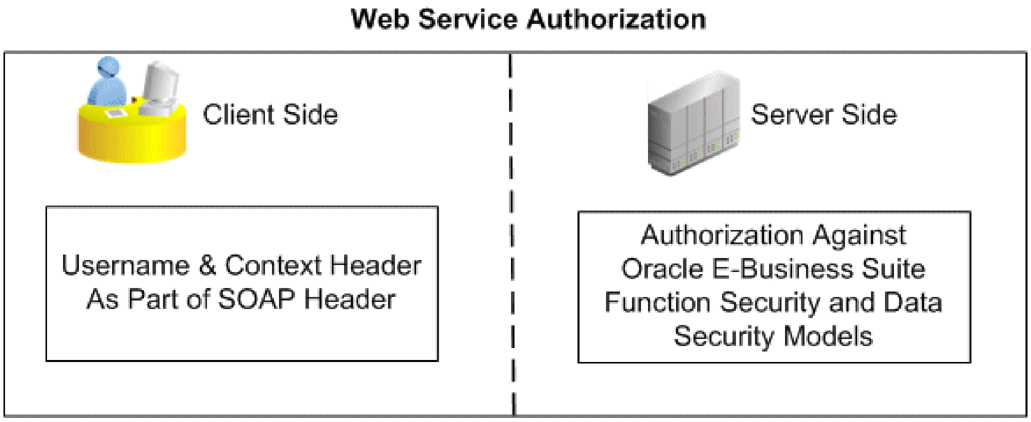
For REST services, when users are authenticating based on provided username/password information in REST requests, security Login service is used to validate user credentials and return a unique access token (such as Oracle E-Business Suite session ID). The token may be sent to LoginModule and used in subsequent requests for token based authentication.

Subject Authorization to Verify Execution Privileges

At design time, users are given appropriate access privileges to execute certain functions or APIs through security grants and RBAC-based function security.

Authorization for SOAP Services

At run time, SOAP message header information is used to determine whether the current context has access to the operation that is invoked. For example, Oracle E-Business Suite applications context contains many crucial elements that are used in passing values required in proper functioning of Oracle E-Business Suite Web services. This context header information is required for an API transaction or a concurrent program in order for an Oracle E-Business Suite user who has sufficient privileges to run the program.


Web Service Authorization
Client Side
Server Side
Username & Context Header As Part of SOAP Header
Authorization Against Oracle E-Business Suite Function Security and Data Security Models

The following code snippet shows the sample header of applications context:

```
<soapenv:Header>
 ..
 <!--wsse Header-->
 <fnd:SOAHeader>
 <fnd:Responsibility>SYSTEM_ADMINISTRATOR</fnd:Responsibility>
 <fnd:RespApplication>FND</fnd:RespApplication>
 <fnd:SecurityGroup>STANDARD</fnd:SecurityGroup>
 <fnd:NLSLanguage>AMERICAN</fnd:NLSLanguage>
 <fnd:Org_Id>204</fnd:Org_Id>
</fnd:SOAHeader>
</soapenv:Header>
```

For more information about SOAP header elements used for authorization, see SOAP Header for Applications Context, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

Authorization for REST Services

After authentication, the LoginModule verifies the authenticated user's role and access privilege and then authorizes the execution of underlying API only if the user has the required privilege to execute the API.

UsernameToken Based Security

In the UsernameToken based security, the username/password sent in the SOAP header for authentication is associated with the User created in Oracle E-Business Suite.

Username is a clear text; password is the most sensitive part of the UsernameToken profile. In this security model, the supported password type is plain text password (or PasswordText).

> **Note:** The PasswordText password type is the password written in clear text. SOAP requests invoking the Web services should include security header consisting of Username and plain text password. The password received as part of the SOAP request at runtime will be

validated against the encrypted password stored in Oracle E-Business Suite. After validation, the plain text password from the SOAP request will be discarded.

At run time, SOAP request messages received through Oracle SOA Suite are passed on to a JAAS based login module for Oracle E-Business Suite for authentication based on the `wsse:security` Web Security headers.

A basic UsernameToken security header can be explained as follows:

```
<S11:Envelope xmlns:S11="..." xmlns:wsse="...">
 <S11:Header>
...
   <wsse:Security>
   <wsse:UsernameToken>
     <wsse:Username>Zoe</wsse:Username>
     <wsse:Password>password</wsse:Password>
     </wsse:UsernameToken>
   </wsse:Security>
...
  </S11:Header>
...
</S11:Envelope>
```

Important: Authorization Check at Both the Trading Partner Level and WS-Security Header Level for XML Gateway Interfaces

In Oracle XML Gateway, each trading partner is configured with Oracle E-Business Suite users. Only these authorized users defined in the Trading Partner Setup form are allowed to perform XML transactions. External clients can pass such usernames in the `<USERNAME>` and `<PASSWORD>` elements defined within the `<ECX:SOAHeader>` element (or `<XMLGateway_Header>` element for generic XML Gateway services) in the SOAP body. These username parameters are validated by Oracle XML Gateway against the username defined in the trading partner setup before initiating a transaction.

Therefore, for XML Gateway interface type, the authorization check is performed at both the trading partner level, as well as on the username passed in the `wsse:security` header in the SOAP request. For information on trading partner setup and how to associate users with trading partners, see the *Oracle XML Gateway User's Guide*.

A WS-Security header in the SOAP message from Oracle E-Business Suite can be as follows:

```
<xml version="1.0" encoding="UTF-8">
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <env:Header>

 <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wsswssec
urity-secext-1.0.xsd>
  <wsse:UsernameToken>
     <wsse:Username>Kwalker</wsse:Username>
     <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-t
oken-profile-1.0#PasswordText">password</wsse:Password>
     </wsse:UsernameToken>
  </wsse:Security>
 </env:Header>

 <env:Body>
...
 </env:Body>
</evn:Envelope>
```

SAML Sender-Vouches Token Based Security

To authenticate Web services relying on sending a username only through SAML assertion, Oracle E-Business Suite Integrated SOA Gateway supports SAML Token (Sender Vouches) based Web service security.

Security Assertion Markup Language (SAML) is an XML-based standard for exchanging authentication and authorization data between security domains, that is, between an identity provider and a service provider.

How to Authenticate Users through a Trusted Sender-Vouches SAML Token

A SAML token uses SAML assertions as security tokens. One type of SAML token is the sender-vouches SAML token. This token uses a sender-vouches method to establish the correspondence between a SOAP message and the SAML assertions added to the SOAP message.

When a Web application invokes a service that uses SAML token as its authentication type, this SOAP request message containing or referencing SAML assertions is received through Oracle SOA Suite and passed on to a JAAS based login module for Oracle E-Business Suite to authenticate the service based on the `wsse:security` Web Security headers. As part of the validation and processing of the assertions, the receiver or the login module for Oracle E-Business Suite must establish the relationship between the subject, claims of the referenced SAML assertions, and the entity providing the evidence to satisfy the confirmation method defined for the statements.

In other words, in order to validate and authenticate a user who logs on to the enterprise information system, a trusted sender-vouches SAML token security must be used to establish the correspondence between the SOAP message and the SAML assertions added to the SOAP message.

> **Note:** Since everyone can send a SAML Token with valid conditions, the authentication framework only trusts certain SAML token sources and stores the public key of each of these sources in a common key store. This Public Key Infrastructure (PKI) based security provides more sophisticated trusted rules to authenticate Web services.
>
> Please note that the following algorithms have been certified for SAML Token security in this release:
>
> - Symmetric Encoding Algorithm: `http://www.w3.org/2001/04/xmlenc#aes128-cbc`
> - Key Encryption Algorithm: `http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p`

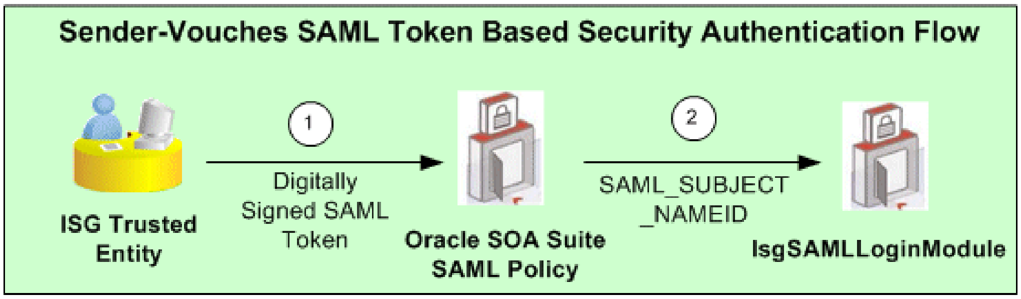
> **Important:** To ensure SAML Token security works properly, necessary setup steps need to be performed. For setup information on SAML Token security, see *Setting Up SAML Token Security for Oracle E-Business Suite Integrated SOA Gateway Release 12.2*, My Oracle Support Knowledge Document 1332262.1 for details.

To authenticate users, any entity that establishes a PKI trust with Oracle E-Business Suite Integrated SOA Gateway can send the SAML Assertion with a valid Username. A PKI trusted entity will send a SAML token profile with the username embedded with it and that must be digitally signed. The SAML Token policy attached to the Web service verifies attributes like "Issuer", "Conditions", and so on. After the verification, the login module (`IsgSAMLLoginModule`) extracts the SAML principal (username in `NameIdentifier`) through a NameCallback. This is verified against LDAP for Single Sign-On (SSO) users or against Oracle E-Business Suite `FND_USER` for non-SSO users.

Please note that for Oracle E-Business Suite Integrated SOA Gateway, it is mandatory that all users must be valid Oracle E-Business Suite users. If SSO is used, then the user in LDAP server for SSO should be in synchronous with Oracle E-Business Suite `FND_USER` table. Otherwise, the user authorization check will fail when looking up the application responsibilities for user authorization against entries in the `FND_USER` table. For more information on integrating Oracle E-Business Suite in an enterprise single sign-on environment, see the *Oracle E-Business Suite Security Guide*.

> **Note:** The login module `IsgSAMLLoginModule` gets invoked through the Authentication Provider `IsgAuthenticator`.

The following diagram illustrates the sender-vouches SAML Token based security authentication process flow:


Sender-Vouches SAML Token Based Security Authentication Flow
ISG Trusted Entity
1
Digitally Signed SAML Token
Oracle SOA Suite SAML Policy
2
SAML_SUBJECT _NAMEID
IsgSAMLLoginModule

1. A trusted application authenticates a user and creates a digitally signed SOAP request, containing a SAML Sender-Vouches Token.

   Please note that a trusted application can be any application whose Public Key is known to Oracle E-Business Suite Integrated SOA Gateway and which can send digitally signed SAML Assertions in SOAP requests using that public key.

2. SAML Token Policy attached to the Web service verifies signature and SAML conditions.

3. `IsgSAMLLoginModule` in Oracle SOA Suite extracts the SAML principal (username in `NameIdentifier`) through a NameCallback. This is verified against LDAP for Single Sign-On (SSO) users or against Oracle E-Business Suite `FND_USER` for non-SSO users.

   The format of the `NameIdentifier` indicates if the user has been authenticated against LDAP (for a SSO user) or Oracle E-Business Suite `FND_USER` (for a non-SSO user). If the format is `dn=xxxx`, then this is a SSO user who has been authenticated against LDAP. Otherwise, this is a non-SSO user who has been authenticated against Oracle E-Business Suite `FND_USER`.

A sample sender-vouches SAML assertion for a non-SSO environment can be as follows:

```
<Assertion AssertionID="be7d9814c36381c27fefa89d8f27e126"
IssueInstant="2010-02-27T17:26:21.241Z" Issuer="www.oracle.com"
MajorVersion="1" MinorVersion="1"
xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"><Conditions
NotBefore="2010-02-27T17:26:21.241Z"
NotOnOrAfter="2011-02-27T17:26:21.241Z"/>
 <AuthenticationStatement
AuthenticationInstant="2010-02-27T17:26:21.241Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
  <Subject>
    <NameIdentifier
Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"
NameQualifier="notRelevant">SYSADMIN</NameIdentifier>
    <SubjectConfirmation>

<ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-vouches</Confi
rmationMethod>
    </SubjectConfirmation>
  </Subject>
 </AuthenticationStatement>
</Assertion>
```

A sample sender-vouches SAML assertion for a SSO environment can be as follows:

```
<Assertion
IssueInstant="2010-02-27T17:26:21.241Z" Issuer="www.oracle.com"
MajorVersion="1" MinorVersion="1"
xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"><Conditions
NotBefore="2010-02-27T17:26:21.241Z"
NotOnOrAfter="2011-02-27T17:26:21.241Z"/>
<AuthenticationStatement
AuthenticationInstant="2010-02-27T17:26:21.241Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
  <Subject>
    <NameIdentifier
Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"

NameQualifier="notRelevant">orclApplicationCommonName=PROD1,cn=EBusiness
,cn=Products,cn=OracleContext,dc=us,dc=oracle,dc=com</NameIdentifier>
    <SubjectConfirmation>

<ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-vouches</Confi
rmationMethod>
    </SubjectConfirmation>
  </Subject>
 </AuthenticationStatement>
</Assertion>
```

- `Issuer`: The value of this attribute is defined through Oracle SOA Suite. It will appear in `jps-config.xml`. For information on how to add `Issuer`, see *Setting Up SAML Token Security for Oracle E-Business Suite Integrated SOA Gateway Release 12.2,* My Oracle Support Knowledge Document 1332262.1.

- `Conditions`: This tag defines the time limit in which this SAML Assertion is valid.

- `NameIdentifier`: The value of this tag contains the username.

If the username is of the form of LDAP DN, then the username is verified in the registered OID for a SSO user. Otherwise, the username is verified in `FND_USER` table for a non-SSO user.

- `SubjectConfirmation`: It should be sender-vouches.

For information on how the sender-vouches SAML Token is used in SOAP security header to authenticate Web services, see SAML Token-based SOAP Security Header, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

HTTP Basic Authentication

Oracle E-Business Suite Integrated SOA Gateway supports HTTP Basic Authentication security to authenticate the users who invoke REST services over secure transport protocol – HTTPS.

When an HTTP client application tries to access an Oracle E-Business Suite REST service, user security credentials (username/password) should be provided as input data in HTTP header as part of the REST request message. The username and password will be routed to LoginModule for authentication and authorization.

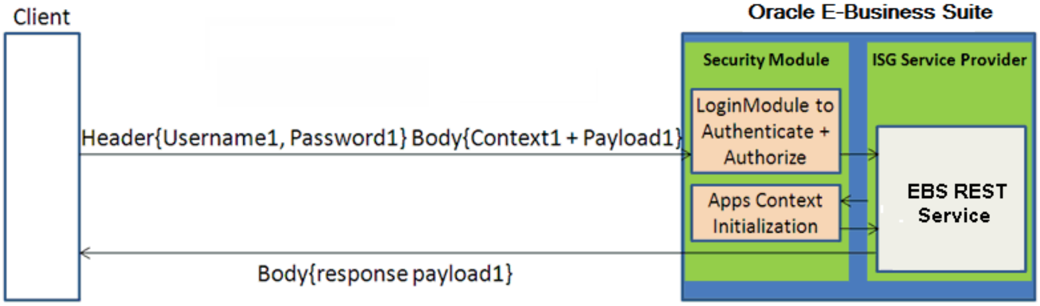
The LoginModule in turn extracts the credentials from HTTP header, authenticates user against Oracle E-Business Suite user table, and establishes identity for the authenticated user. The LoginModule will then send the response to ISG Service Provider framework.

- For the authenticated and authorized user request, the Service Provider framework invokes a security service to initialize the applications context, and then execute the REST service.

- For the unauthenticated or unauthorized user request, the Service Provider framework returns system fault to the client.

The following diagram illustrates the authentication process flow of HTTP Basic Authentication security:

Authentication Process Flow for HTTP Basic Authentication Security


Client
Oracle E-Business Suite
Security Module
ISG Service Provider
LoginModule to Authenticate + Authorize
Header{Username1, Password1} Body{Context1 + Payload1}
Apps Context Initialization
EBS REST Service
Body{response payload1}

Based on HTTP Basic Authentication defined by W3C, the HTTP client application should use the following header field to send user credentials:

```
Authorization: Basic <base64 encoded version of
username:password>
```

Please note that if it is a SSO-enabled Oracle E-Business Suite environment, user authentication should be delegated to SSO which performs authentication against information stored in Oracle Internet Directory (an LDAP server).

Token Based Authentication

Token based security authenticates users using security tokens provided by the server. When a user tries to log on to a server with multiple requests, instead of authenticating the user each time with username and password, a unique access token (such as Oracle E-Business Suite session ID) in place of password may be sent along with username in HTTP headers.

For example, when an Oracle E-Business Suite user has initially authenticated on a given username and password, after successful login, the security Login service creates an Oracle E-Business Suite user session and returns the session ID, as shown in the following:

```
<response>
<data>
<accessToken>OEj6fXXoDoo4EeubKLgYes7io7</accessToken>
<accessTokenName>myEbsInstance</accessTokenName>
<ebsVersion>12.2.0</ebsVersion>
<userName>SYSADMIN</userName>
</data>
</response>
```

The session ID that points to the user session will be passed as `Cookie` to HTTP headers of all subsequent Web service calls for user authentication.

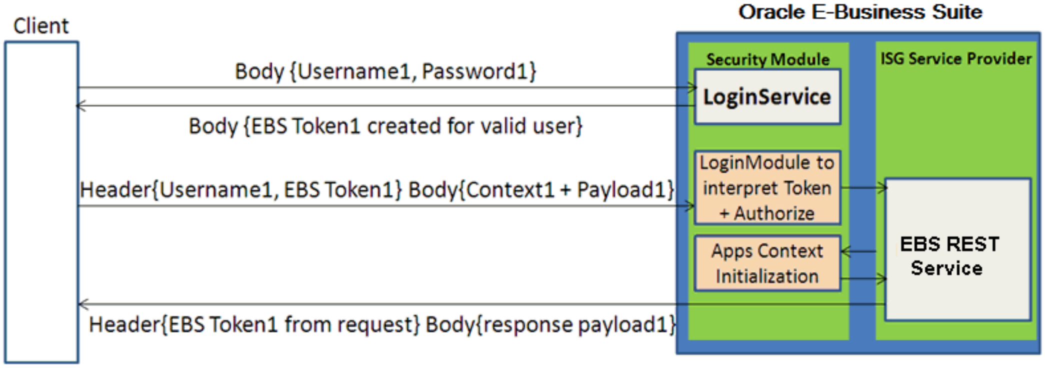
```
POST /webservices/rest/Invoice/create_invoice
Cookie: <accessTokenName>=<accessToken>
Content-Type: application/xml
```

The LoginModule will interpret and extract the token (session ID) from HTTP headers, and validate the subject or username with token, not password, in the subsequent requests for authentication.

Similar to the HTTP Basic Authentication security, if the request passes the authentication and authorization, the Service Provider framework invokes a security service to initialize the applications context, and then executes the REST service. Otherwise, system fault will be returned.

The following diagram illustrates the authentication process flow of Token Based Authentication security:

Authentication Process for Token Based Authentication Security



Client
Oracle E-Business Suite
Security Module
ISG Service Provider
Body {Username1, Password1}
LoginService
Body {EBS Token1 created for valid user}
Header{Username1, EBS Token1} Body{Context1 + Payload1}
LoginModule to interpret Token + Authorize
Apps Context Initialization
EBS REST Service
Header{EBS Token1 from request} Body{response payload1}

In this diagram, username/password information is provided and validated in the initial request. A unique token (EBS Token1) is obtained through the Login Service for the valid user. In case a different service is requested in the subsequent call, username along with the token, instead of the password, are provided in the header this time.

In this subsequent request, applications context information that may be required in initializing Oracle E-Business Suite session is also provided in the request. Security LoginModule will be used to interpret and extract the token from the header to authenticate the user and then authorize the request. Applications context session will also be initialized before invoking the REST service. After a successful service invocation, a response message will be sent along with the response payload if it is available.

Advantages of Using Token Based Security

Please note that when token based security is used, applications context information mentioned above does not have to be passed in every request. If the context values are not provided in the consecutive requests, the previously passed values will be used.

This will reduce the size of the payload included in HTTP headers and thus less data bandwidth is required. It is particularly useful for mobile data networks.

For more information on applications context in REST header, see REST Header for Applications Context, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

For more information about how to use context values to initialize or re-initialize the Oracle E-Business Suite session, see the Oracle Application Object Library REST Security Services section, *Oracle E-Business Suite Security Guide*.

7

Logging for Web Services

Overview

To extend logging support to more granular level and provide inside-out views for Web service activities, Oracle E-Business Suite Integrated SOA Gateway provides an enhanced, flexible Web service logging mechanism. An integration repository administrator can configure log settings at the integration interface level. This includes selecting a desired interface name that the logging feature should be set, enabling or disabling the design-time log, and selecting an appropriate runtime log severity level. Additionally, the Web service auditing feature can be enabled or disabled through the same logging user interface at the interface level.

With proper logging setups and configuration, you can easily monitor and audit Web service activities provided through Oracle E-Business Suite Integrated SOA Gateway. You can track log messages, and troubleshoot any issues occurred at design time and run time. Moreover, the administrator can delete existing log settings, and purge audit information through Service Monitor if needed.

> **Important:** Logging feature is supported for SOAP services only. This feature is not supported for REST services in this release.

Key Features

The enhanced Web service logging feature includes the following features:

- It provides centralized, user-friendly user interface for logging and audit configuration for Oracle E-Business Suite SOAP services.
- It allows logging and audit setups to be configured at the integration interface level.
- It lets you enable or disable the design-time and runtime logs, as well as Web service auditing feature.
- All design-time and runtime SOAP service activities within Oracle E-Business Suite

can be logged and audited if the services have logging enabled.

- It provides integrated log view allowing you to view SOAP service generation and deployment logs through Integration Repository, as well as view service processing logs through Service Monitor if the design-time and runtime logs are enabled.
- Audit information can be purged from the database tables through Service Monitor.

Design-time logs capture each stage of SOAP service generation and development life cycle activities only if the design-time log is enabled for the selected interface or service.

- If an interface has the design-time log enabled, 'Enabled' is displayed in the Log Configuration field. **View Log** appears in the interface details page for that interface allowing you to view the log details in the Log & Error Details page.

  If any errors occurred during the design-time activities, the error details are also displayed in the Log & Error Details page.
- If the design-time log is not enabled, 'Disabled' is displayed in the Log Configuration field. If errors occurred while performing the design-time activities, then **View Error** appears instead for that interface allowing you to view the error and exception details only.

Runtime logs record service processing details during the invocation of Oracle E-Business Suite services by Web service clients if the service has runtime log enabled. If a log is available for a given instance, the **Log** icon appears in Service Monitor. The administrator can view the log messages.

Audit feature allows you to monitor and track Web service activities executed through Oracle SOA Suite if the audit feature for a specific interface or service is enabled. All SOAP messages for the interface or service that has the Audit feature enabled, the associated payloads and fault messages can be saved and audited through Service Monitor.

To better understand the logging feature, the following topics are discussed in this chapter:

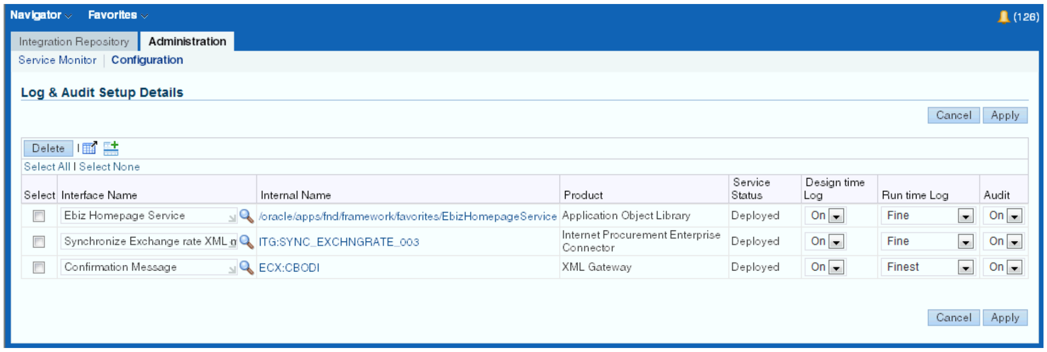
- Accessing the Logging Configuration User Interface, page 7-3
- Viewing and Searching Existing Configurations, page 7-4
- Adding a New Configuration, page 7-6
- Updating an Existing Configuration, page 7-9
- Deleting an Existing Configuration, page 7-9
- Viewing, Deleting and Exporting Log Messages, page 7-11

Accessing the Logging Configuration User Interface

To access the log and audit setup page, log on to Oracle E-Business Suite with the username who has been granted the integration repository administrator role.

Select the **Integrated SOA Gateway** responsibility from the navigation menu, and then select **Administration > Configuration**. The Administration tab appears with the Configuration subtab.

> **Note:** The **Administration** selection from the navigation menu appears only to the users who have the integration repository administrator role after logging on to Oracle E-Business Suite with the Integrated SOA Gateway responsibility.
>
> All administrative tasks performed outside the Integration Repository user interface are grouped and displayed under the **Administration** tab. These tasks include managing log and audit setups in the Configuration subtab and monitoring SOAP requests in the Service Monitor subtab.



Log & Audit Setup Details

Select	Interface Name	Internal Name	Product	Service Status	Design time Log	Run time Log	Audit
□	Ebiz Homepage Service	/oracle/apps/fnd/framework/favorites/EbizHomepageService	Application Object Library	Deployed	On	Fine	On
□	Synchronize Exchange rate XML n	ITG:SYNC_EXCHNGRATE_003	Internet Procurement Enterprise Connector	Deployed	On	Fine	On
□	Confirmation Message	ECX:CBODI	XML Gateway	Deployed	On	Finest	On

The Log & Audit Setup Details page is the entry page to perform all the following logging setup and management activities:

- Viewing and Searching Existing Configurations, page 7-4

  All existing logging and audit settings listed by interfaces are displayed in the configuration table once the Log & Audit Setup Details page appears. Each entry in the table includes interface name, internal name, product name, service status, design-time log status (On or Off), runtime log severity level, and audit feature status (On or Off).

  Clicking the Internal Name link from the table takes you to the interface details page for the selected interface in the Integration Repository.

- Adding a New Configuration, page 7-6

  To add a new log configuration for an interface, click **Add Another Row** in the Log

& Audit Setup Details page. An empty row appears allowing you to add a new configuration, including specifying log severity information for runtime logs, and enabling or disabling the design-time log and the service auditing feature for the selected interface.

- Updating an Existing Configuration, page 7-9

  From the configuration table, you can directly update an existing configuration by selecting a desired value for the log setting that you want to change. This setting includes design-time log, log severity level, and audit feature status.

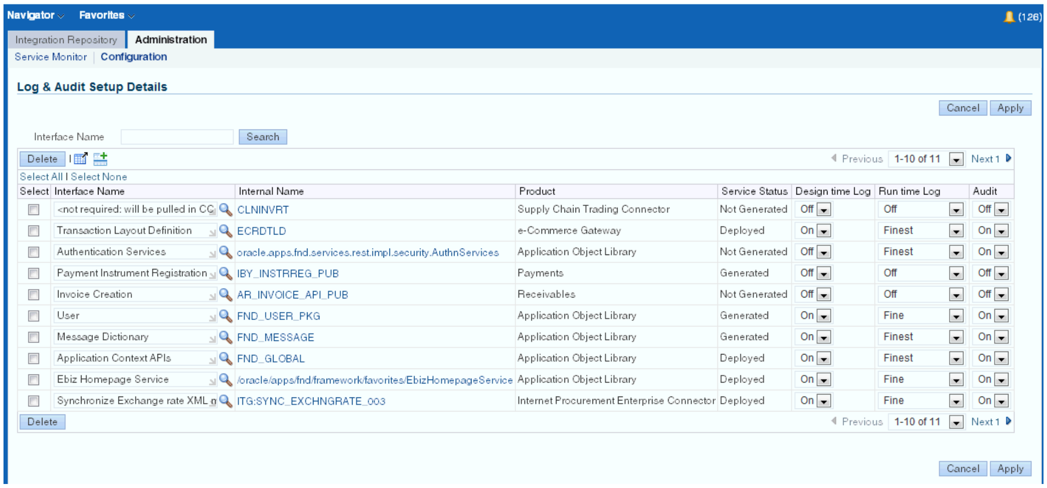
- Deleting an Existing Configuration, page 7-9

  You can delete an existing configuration by selecting an interface with log settings that you want to remove and then clicking **Delete** from the Log & Audit Setup Details page.

Viewing and Searching Existing Configurations

Logging is enabled at the integration interface level. Once an integration repository administrator logs on to Oracle E-Business Suite with the Integrated SOA Gateway responsibility and selects **Administration > Configuration** link from the navigation menu, the Log & Audit Setup Details page is displayed. All existing log configurations by interface are automatically displayed in the configuration table.

Viewing Existing Configurations



Log & Audit Setup Details

Interface Name

Select	Interface Name	Internal Name	Product	Service Status	Design time Log	Run time Log	Audit
□	<not required: will be pulled in CC	CLNINVRT	Supply Chain Trading Connector	Not Generated	Off	Off	Off
□	Transaction Layout Definition	ECRDTLD	e-Commerce Gateway	Deployed	On	Finest	On
□	Authentication Services	oracle.apps.fnd.services.rest.impl.security.AuthnServices	Application Object Library	Not Generated	Off	Finest	On
□	Payment Instrument Registration	IBY_INSTRREG_PUB	Payments	Generated	Off	Off	Off
□	Invoice Creation	AR_INVOICE_API_PUB	Receivables	Not Generated	Off	Off	Off
□	User	FND_USER_PKG	Application Object Library	Generated	On	Fine	On
□	Message Dictionary	FND_MESSAGE	Application Object Library	Generated	On	Finest	On
□	Application Context APIs	FND_GLOBAL	Application Object Library	Deployed	On	Finest	On
□	Ebiz Homepage Service	/oracle/apps/fnd/framework/favorites/EbizHomepageService	Application Object Library	Deployed	On	Fine	On
□	Synchronize Exchange rate XML m	ITG:SYNC_EXCHNGRATE_003	Internet Procurement Enterprise Connector	Deployed	On	Fine	On

Each log entry listed in the table contains interface name, internal name, product name, Web service status, design-time log status (On or Off), runtime log severity level, and audit feature status (On or Off).

Searching Existing Configurations

Search feature is available only if there are more than 10 interfaces that have log settings configured. In this situation, the Interface Name field is displayed on the top of this page allowing you to filter or search the configurations by interface name. After specifying the desired interface name (such as 'Order%') that you want to view the configuration details, click **Search** to execute the query. All interface names that match your search criteria will be displayed in the table.

If no log configuration has been defined, then an empty table with message 'No interface level logging configuration is defined.' appears.

From the configuration table, you can perform the following tasks:

- Add a new log configuration by clicking **Add Another Row**. See: Adding a New Configuration, page 7-6.
- Search the configuration list by Interface Name if there are more than 10 configurations in the table.
- View the selected interface by clicking the Internal Name link. This takes you to the interface details page in the Integration Repository.
- Update an existing configuration for a selected interface. This includes enabling or disabling the design-time log and the service auditing feature, as well as changing runtime log severity level or disabling the runtime log. See: Updating an Existing Configuration, page 7-9.
- Delete an existing configuration by clicking **Delete** for a desired log configuration.

  See: Deleting an Existing Configuration, page 7-10.

To view and search existing configurations:

1. Log on to Oracle E-Business Suite with the username that has been granted the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

   From the navigation menu, select the **Administration > Configuration** link from the menu selection. The Log & Audit Setup Details page is displayed.

2. All existing log and audit configurations are automatically displayed by interface name in the table.

3. If there are more than 10 configurations listed in the table, you can perform a search by entering interface name and click **Search** to execute the query. All matched interfaces will be displayed in the table.

4. To delete existing configurations, select desired settings that you want to delete and click **Delete** to remove them from the database.

5. To add a new configuration, click **Add Another Row** to add a new setting.

Adding a New Configuration

Oracle E-Business Suite Integrated SOA Gateway allows you to configure new log settings at the integration interface level. Click **Add Another Row** in the Log & Audit Setup Details page. An empty row is added to the end of the current configuration table letting you add a new configuration for a specific interface. This includes specifying log severity information for runtime log or disabling the runtime log by setting its value to 'Off', as well as enabling or disabling the design-time log and the service auditing feature for the selected interface.

> **Note:** Design-time logs capture only SOAP service activities recorded at design time, including Generate, Deploy, Undeploy, Reset, Retire, and Activate services, only if the design-time log is enabled for that interface or service. Without enabling the design-time log, the logs will not be written.

Perform the following tasks to add a new configuration in a new role:

- **Identify interface name that the logging will be enabled:**

  Search and select a desired interface name that you want the logging to be enabled. Once the Interface Name field is selected, the associated Internal Name, Product, and Service Status fields are automatically populated.

  > **Note:** Logging is configured at the interface or service level. Configuration at the method or operation level is not supported in this release.

  The rest of the configuration fields including Design Time Log, Run Time Log Level, and Audit fields are also displayed with default values. You can change them if needed.

- **Enable design-time log (optional)**

  Use the design-time logs to troubleshoot any issues or exceptions encountered during service generation and deployment life cycle. By default, the design-time log is turned off initially once the interface name is selected. However, you can enable the feature for the selected interface by selecting 'On' from the drop-down list.

  If the design-time log is enabled for the selected interface or service, logs can be written for the design-time actions such as Generate, Deploy, Undeploy, Reset, Retire, and Activate services. Without enabling the design-time log, the logs will not be written.

  For example, an interface 'Order Capture' has the design-time log enabled. At design time during service generation and deployment, logs specific to the selected 'Order Capture' interface can be captured through the Integration Repository user interface. The **View Log** is displayed in the interface detail page for 'Order Capture' allowing you to view log details and error details if occurred during the design-time activities.

> **Note:** If the design-time log is not enabled, and if any errors occurred while performing the design-time activities, then **View Error** appears instead for that interface. Clicking **View Error** to access and view only the error and exception details in the Log & Error Details page.

For more information on viewing design-time logs, see Viewing Generate and Deploy Time Logs, page 3-24.

- **Enable the auditing feature (optional)**

  By default, the auditing feature is turned off once the interface name is selected. You can enable the feature to create audit trail for the interface by selecting 'On' from the drop-down list.

  If the auditing feature for a specific interface or service is enabled, all SOAP messages for the interface or service that Oracle SOA Suite processes along with the associated payloads and fault messages can be saved and audited through Service Monitor.

  For more information about Service Monitor, see Monitoring and Managing SOAP Messages Using Service Monitor, page 8-1.

- **Enable the runtime log by selecting log severity (optional)**

  By default, the runtime log is turned off once the interface name is selected. You can enable it by changing the log severity level in the Run Time Log field.

  Log level is used to control logging output for the enabled service. Select a different value other than the default 'Off' from the drop-down list to enable the runtime log.

  At run time during the invocation of Oracle E-Business Suite services by Web service clients, if a service has the runtime log enabled, the associated log messages for SOAP services are captured and can be viewed through Service Monitor. Click the **Log** icon in the search result table in Service Monitor to open the Web Service Runtime Logs page where you can view logs recorded for the service against a specific instance.

  > **Important:** Runtime logging for PL/SQL, Concurrent Program, XML Gateway interface types is handled by Oracle SOA Suite; therefore, setting runtime log levels for these services in the Log & Audit Setup Details page here will display Oracle SOA Suite logs if the services are deployed in Oracle SOA Suite. Limited runtime log statements from the Oracle E-Business Suite Integrated SOA

Gateway code (identified by the package name `oracle.apps.fnd.isg`) will be displayed for these services.

Runtime logging for Business Service Object and Java API for Forms interface types is handled by Oracle E-Business Suite Integrated SOA Gateway; therefore, Service Monitor shows Oracle E-Business Suite Integrated SOA Gateway logs for these interfaces based on the log level selected here.

For more information on viewing runtime logs, see Viewing Service Processing Logs, page 8-8.

Log Level

The following table describes the available log levels used for the runtime log:

Log Level

Severity	Description
Off (default)	It is a special level that can be used to turn off logging.
Severe	It is a message level indicating a serious failure.
Warning	It is a message level indicating a potential problem.
Information	It is a message level for informational messages.
Configuration	It is a message level for static configuration messages.
Fine	It is a message level providing tracing information.
Finer	It indicates a fairly detailed tracing message.
Finest	It indicates a highly detailed tracing message.

Please note that log messages can be correlated across middle-tier and database servers. If a new configuration is added for a service that has been deployed, the newly-configured log setting including runtime log level configured for that deployed service will be added in the Oracle SOA Suite. When the configuration is deleted for a deployed service, the runtime log level would be reset at the composite level as well in Oracle SOA Suite. The same mechanism applies when an integration repository administrator updates an existing log level for a deployed service, the new log level will be updated in the database.

If a new configuration is added for a service that is not deployed, then the runtime log configuration including log level set for that service would be effective after the service is deployed.

To add a new configuration:

1. Log on to Oracle E-Business Suite with the username that has been granted with the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

   Select **Administration > Configuration** from the navigation menu. The Log & Audit Setup Details page is displayed.

2. To add a new configuration, click **Add Another Row**.

   An empty row appears allowing you to enter the following information:

   - Interface Name: Specify an appropriate interface name for the log is configured.

     Once the Interface Name field is selected, the associated Internal Name, Product, and Service Status fields are automatically populated. The rest of configuration fields such as the Design Time Log, Run Time Log, and Audit fields are also displayed with default values. You can change them if needed.

   - Design Time Log: By default, it is set to "Off'. You can enable the design-time log by selecting 'On' from the drop-down list.

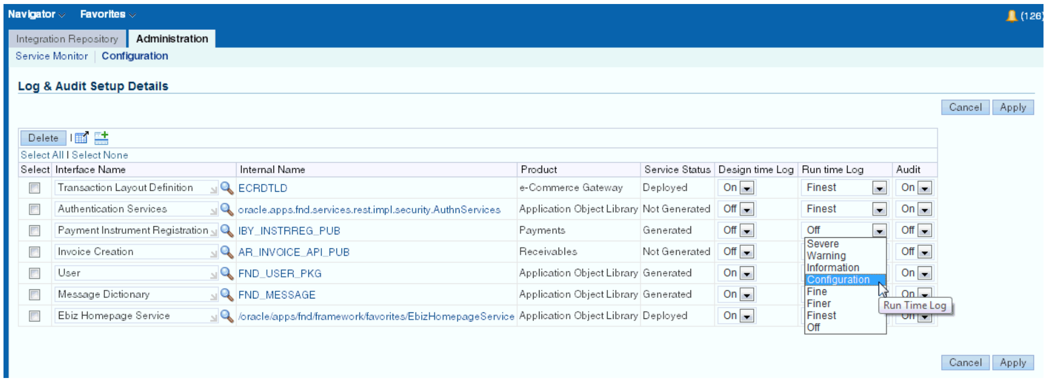
   - Run Time Log: By default, it is set to "Off' and the runtime log is turned off. You can change the default value by selecting an appropriate value from the drop-down list.

   - Audit: By default, it is set to "Off'. You can enable the auditing feature by selecting 'On' from the drop-down list.

3. Click **Apply** to save the information.

Updating an Existing Configuration

From the Log & Audit Setup Details page, you can modify an existing configuration for a selected interface including changing runtime log severity, and enabling or disabling the design-time log and the auditing feature.



Log & Audit Setup Details

Select	Interface Name	Internal Name	Product	Service Status	Design time Log	Run time Log	Audit
□	Transaction Layout Definition	ECRDTLD	e-Commerce Gateway	Deployed	On	Finest	On
□	Authentication Services	oracle.apps.fnd.services.rest.impl.security.AuthnServices	Application Object Library	Not Generated	Off	Finest	On
□	Payment Instrument Registration	IBY_INSTRREG_PUB	Payments	Generated	Off	Off	Off
□	Invoice Creation	AR_INVOICE_API_PUB	Receivables	Not Generated	Off		Off
□	User	FND_USER_PKG	Application Object Library	Generated	On		On
□	Message Dictionary	FND_MESSAGE	Application Object Library	Generated	On		On
□	Ebiz Homepage Service	/oracle/apps/fnd/framework/favorites/EbizHomepageService	Application Object Library	Deployed	On		On

To update the log settings for an interface, select appropriate values from the drop-down lists. For example, enable the runtime log for the 'Order Capture' interface and set an appropriate log level. This is achieved by changing the 'Off' value to 'Information' in the Run Time Log field. All informational messages during service invocation specific for the 'Order Capture' service will be written.

After modifying the existing settings for an interface, click **Apply** to save changes to the database and in Oracle SOA Suite if the changes applied to a service that has been deployed. Click **Cancel** to display the previous saved details.

To update an existing configuration:

1. Log on to Oracle E-Business Suite with the username that has been granted with the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

   Select **Administration > Configuration** from the navigation menu. The Log & Audit Setup Details page is displayed.

2. Update the basic log settings for an interface by selecting appropriate values from the drop-down lists for the design-time log, runtime log level, and the Audit field.

3. After the modification, click **Apply** to save the changes. Click **Cancel** to display the previous saved details.

Deleting an Existing Configuration

If an existing configuration is no longer needed, you can remove it directly from the Log & Audit Setup Details page.

To delete existing configurations, select at least one setting that you want to remove and then click **Delete**. This removes the records from the existing configuration list and database. A confirmation message appears indicating that the selected log setups have been successfully deleted. This disables the logging and audit features for the selected interfaces.

For a service that has been deployed to Oracle SOA Suite, once a configuration is deleted for that service, the runtime log level would be reset at the composite level as well in Oracle SOA Suite.

If you click **Delete** without first selecting log configurations that you want to delete, then an advice message appears indicating that you should select at least one interface level log configuration for deletion.

To delete an existing logging configuration:

1. Log on to Oracle E-Business Suite with the username that has been granted with the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

   Select **Administration > Configuration** from the navigation menu. The Log & Audit Setup Details page is displayed.

2. To delete an existing configuration, select the desired interface level setting that you want to remove and click **Delete**. The configuration for the selected interface is removed from the list and the system.

Viewing, Deleting, and Exporting Log Messages

To effectively troubleshoot or debug errors if occurred at each stage of service deployment life cycle, you can view and download log details recorded for an interface or service if it has the logging feature enabled properly.

Please note that sensitive information such as passwords, and security credentials in unencrypted plain text will not be logged.

Viewing Generate and Deploy Time Logs

At design time during service generation and deployment life cycle, logs can be captured through the Integration Repository user interface if the design-time log is enabled for a specific interface. If an interface has the design-time log enabled, **View Log** appears in the interface details page for that interface.

> **Note:** If an interface that does not have the design-time log enabled and if errors occurred during the design-time activities such as Generate, Deploy, Undeploy, Reset, Retire, and Activate, **View Error** appears instead allowing you to view only the error or exception message details. You will not find log messages recorded at the design time because the design-time log is not enabled.

Click **View Log** to open the Log & Error Details page where you can view log messages compiled in a table in the Log Details region as well as view error message details in the Error Details region only if errors occurred during the design-time activities.

Deleting and Exporting Logs in the Log Details Region

After viewing log messages retrieved for an interface in the Log Details region, you can delete them if needed by clicking **Delete Log**. A warning message appears alerting you that this will permanently delete all the logs retrieved in the region. Click **Yes** to confirm the action. An empty log table appears in the Log Details region after logs have been successfully deleted.

Before deleting the logs, you can save a backup copy by clicking **Export**. This allows you to export the records listed in the Log Details region to Microsoft Excel and use it later.

Viewing Service Processing Logs

At run time during the invocation of Oracle E-Business Suite services by Web service clients, log messages can be captured and viewed through Service Monitor. Click the **Log** icon in the search result table for a request in Service Monitor to open the Web Service Runtime Logs page where you can view the log details for the request against a specific instance.

The Web Service Runtime Logs page contains the following log regions:

- **Runtime Middle Tier Logs**: Logs in this region are retrieved from the Oracle SOA Suite server's file system for Oracle E-Business Suite integration.

- **Adapter Logs:** Logs in this region are executed for Web services on the Oracle E-Business Suite side and retrieved from the Oracle E-Business Suite table.

Deleting and Exporting Adapter Logs Retrieved from Oracle E-Business Suite Table

After viewing adapter log messages retrieved from the Oracle E-Business Suite table for a service, you can delete them if needed by clicking **Delete Log**. A warning message appears alerting you that this will permanently delete all the logs retrieved in the Adapter log table. Click **Yes** to confirm the action. An empty log table appears after adapter logs have been successfully deleted.

Before deleting the logs, you can save a backup copy by clicking **Export**. This exports the records listed in the table to Microsoft Excel.

> **Note:** Please note that the log records deleted here are instance specific, whereas the Purge program from the Service Monitor requiring you to enter specific date range in executing the concurrent request is not. The purge concurrent request will delete only the service processing logs for which the service is completed with a status of 'SUCCESS'. It does not delete the logs for the service with 'FAILURE' status.
>
> For more information on purging logs through Service Monitor, see Purging SOAP Messages, Audits, and Logs, page 8-11.

For more information on viewing logs recorded during service deployment life cycle through Integration Repository, see Viewing Generate and Deploy Time Logs, page 3-24.

For more information on viewing log messages recorded while processing service requests, see Viewing Service Processing Logs, page 8-8.

8

Monitoring and Managing SOAP Messages Using Service Monitor

This chapter covers the following topics:

- Service Monitor Overview
- Searching SOAP Requests
- Viewing SOAP Request and Response Details
- Viewing Service Processing Logs
- Purging SOAP Messages, Audits, and Logs
- Enabling Web Service Auditing Using the Configuration Subtab

Service Monitor Overview

Service Monitor, previously known as SOA Monitor, is a centralized, light-weight service execution monitoring and management tool. It fetches data and statistics for each instance of a Web service request and response and provides monitoring capability for Oracle E-Business Suite Web services.

You can view all runtime SOAP request and response data received and sent from Oracle SOA Suite through the Service Monitor user interface in Oracle E-Business Suite. Additionally, Service Monitor provides auditing records for the service execution details if the auditing feature is enabled.

> **Important:** Only SOAP services are monitored and audited through Service Monitor. Runtime REST service monitoring and auditing features are not supported in this release.

For the monitoring purpose, Service Monitor stores basic information about service execution for all the services such as instance ID, integration interface details, SOAP header, start date, end date, status and so on. Please note that it does not store SOAP request and response payloads along with the attachments unless the auditing feature is turned on.

When the auditing feature is enabled, Service Monitor saves the payloads of SOAP requests and responses, fault messages, and attachments if they are available for an instance. This auditing feature provides additional audit trails for integration repository administrators to quickly retrieve service execution details as well as identify errors or exceptions if occurred.

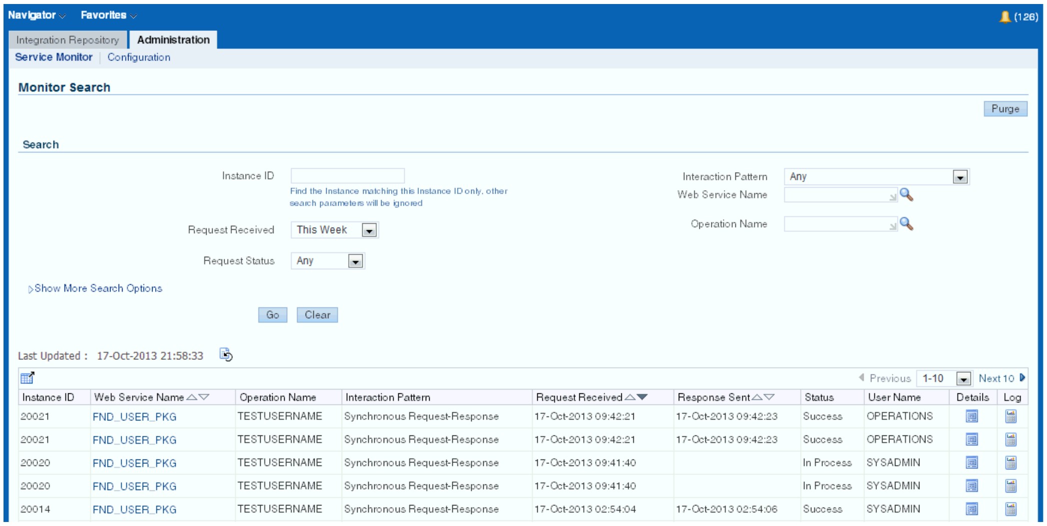
> **Important: Enabling Service Auditing Feature Using the Configuration Subtab**
>
> For the monitoring feature, Service Monitor is a permanent monitoring tool and it is enabled at all times to monitor all Oracle E-Business Suite Web services. However, its auditing feature needs to be explicitly enabled at the interface or service level through the Log & Audit Setup Details page.
>
> For more information on how to enable the auditing feature along with log configuration at the interface or service level, see Adding a New Configuration, page 7-6.

Accessing Service Monitor

To access Service Monitor, log on to Oracle E-Business Suite with the username that has been granted with the integration repository administrator role.

Select the **Integrated SOA Gateway** responsibility from the navigation menu and then select the **Administration > Service Monitor** link. The Service Monitor subtab is displayed with the Service Monitor Search page.

> **Note:** The **Administration** selection from the navigation menu appears only to the users who have the integration repository administrator role after logging on to Oracle E-Business Suite with the Integrated SOA Gateway responsibility.
>
> All administrative tasks performed outside the Integration Repository user interface are grouped and displayed under the **Administration** tab. These tasks include monitoring SOAP requests in the Service Monitor subtab and managing log and audit setups in the Configuration subtab.



Monitor Search

Search

Instance ID:
Find the Instance matching this Instance ID only, other search parameters will be ignored

Request Received: This Week

Request Status: Any

Interaction Pattern: Any

Web Service Name:

Operation Name:

Last Updated : 17-Oct-2013 21:58:33

Instance ID	Web Service Name	Operation Name	Interaction Pattern	Request Received	Response Sent	Status	User Name	Details	Log
20021	FND_USER_PKG	TESTUSERNAME	Synchronous Request-Response	17-Oct-2013 09:42:21	17-Oct-2013 09:42:23	Success	OPERATIONS		
20021	FND_USER_PKG	TESTUSERNAME	Synchronous Request-Response	17-Oct-2013 09:42:21	17-Oct-2013 09:42:23	Success	OPERATIONS		
20020	FND_USER_PKG	TESTUSERNAME	Synchronous Request-Response	17-Oct-2013 09:41:40		In Process	SYSADMIN		
20020	FND_USER_PKG	TESTUSERNAME	Synchronous Request-Response	17-Oct-2013 09:41:40		In Process	SYSADMIN		
20014	FND_USER_PKG	TESTUSERNAME	Synchronous Request-Response	17-Oct-2013 02:54:04	17-Oct-2013 02:54:06	Success	SYSADMIN		

Integration repository administrators can perform the following activities through Service Monitor:

- Searching SOAP Requests, page 8-3
- Viewing SOAP Request and Response Details, page 8-5
- Viewing Log Messages, page 8-8
- Purging SOAP Messages, Audits, and Logs, page 8-11
- Enabling Web Service Auditing Using the Configuration Subtab, page 8-12

Searching SOAP Requests

In the Search region, you can perform searches on SOAP requests received from Oracle SOA Suite based on the criteria you specified.

Service Monitor allows you to search SOAP requests by instance ID, interaction pattern, request status, Web service name, operation name, and request received time.

The Interaction Pattern field drop-down selection includes asynchronous pattern support in addition to synchronous operation. You can select an appropriate value from the selection such as 'Any', 'Synchronous Request-only', 'Asynchronous Request-only', 'Synchronous Request-Response', or 'Asynchronous Request-Response'.

The Request Received time can be selected from the list of values. Its value can be 'Any Time', 'Last 2 Weeks', 'Last 30 Days', 'Last 60 Days', 'Last 90 Days', 'This Week', and 'Today'.

> **Note:** All the list of value selections from the Request Received field

will include the requests received day of Today except 'Any Time'. For example, 'This Week' means the last 7 days inclusive of today the requests have been received, and 'Last 30 Days' means the last 30 days inclusive of today the requests have been received.

'Any Time' means a blind search of requests received regardless of the Request Received date. If this field is left blank, then 'This Week' is the default value for the Request Received time.

You can optionally enter more search criteria by clicking the **Show More Search Options** link in the Search region. These criteria include username and a selected time frame.

When the search is executed, all entries that match your search criteria will be retrieved and displayed in a table. Each entry in the result table includes the instance ID, Web service name, operation name, interaction pattern, date and time the request was received and responded, username, and request status.

If service processing log messages are available for an instance, the **Log** icon is enabled in the result table allowing you to view the log messages.

From the search result page, you can perform the following tasks:

- View the status of each monitored SOAP request and response

- View the service details in the Integration Repository by clicking a specific Web service name link

- View SOAP request and response details by clicking the **Details** icon for a given SOAP request

  See: Viewing SOAP Request and Response Details, page 8-5.

- View service processing log details by clicking the **Log** icon if log messages are available for an instance

  See: Viewing Log Messages, page 8-8.

- Purge SOAP requests and responses, audits, as well as log messages collected over a period of time by clicking **Purge**

  See: Purging SOAP Messages, Audits, and Logs, page 8-11.

Please note that the Web service auditing feature is enabled at the integration interface level through the Log & Audit Setup Details page. For more information on how to enable or disable the auditing feature, see Adding a New Configuration, page 7-6.

To perform a search:

1. Log on to Oracle E-Business Suite with the username that has been granted with the integration repository administrator role. Select the Integrated SOA Gateway

responsibility. From the navigation menu, select the **Service Monitor** link from the Administration section to open the Monitor Search page.

2. In the Search region, enter appropriate search criteria including instance ID, interaction pattern, request status, Web service name, operation name, and request received time for your search. Click **Go** to execute your search.

3. Optionally, enter more search criteria by clicking the **Show More Search Options** link to enter the following information:

   - From: Enter an appropriate search start date.

   - To: Enter an appropriate search end date.

   - Username: Search and select an appropriate username.

   Click **Go** to execute your search.

4. All SOAP requests that match your search criteria appear.

5. Click the **Details** icon for a given instance ID to view the SOAP request and response details.

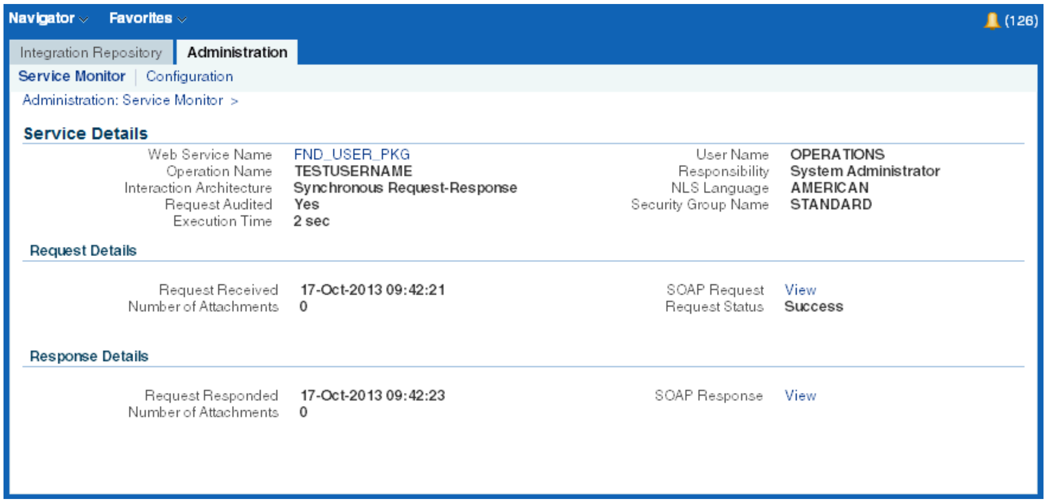
6. Click the **Log** icon, if service processing logs are available for a given instance ID, to view the log details.

7. Click **Purge** to purge SOAP requests and responses, audits, as well as log messages collected for a period of time.

Viewing SOAP Request and Response Details

After executing a search, all SOAP messages that match your criteria are retrieved. To view the SOAP request and response details, click the **Details** icon for a given instance ID listed in the search result table. The Request and Response Details page appears.

SOAP Request and Response Details Page



Service Details

Web Service Name	FND_USER_PKG	User Name	OPERATIONS
Operation Name	TESTUSERNAME	Responsibility	System Administrator
Interaction Architecture	Synchronous Request-Response	NLS Language	AMERICAN
Request Audited	Yes	Security Group Name	STANDARD
Execution Time	2 sec		

Request Details

Request Received	17-Oct-2013 09:42:21	SOAP Request	View
Number of Attachments	0	Request Status	Success

Response Details

Request Responded	17-Oct-2013 09:42:23	SOAP Response	View
Number of Attachments	0		

General SOAP request heading is displayed at the top of the page. This header information includes Web service name, operation name, interaction pattern, username, responsibility, NLS language, security group name, execution time, and whether the request is audited or not.

Clicking a Web service name link launches the interface details page for the service in Integration Repository. This lets you view the integration interface and service in details.

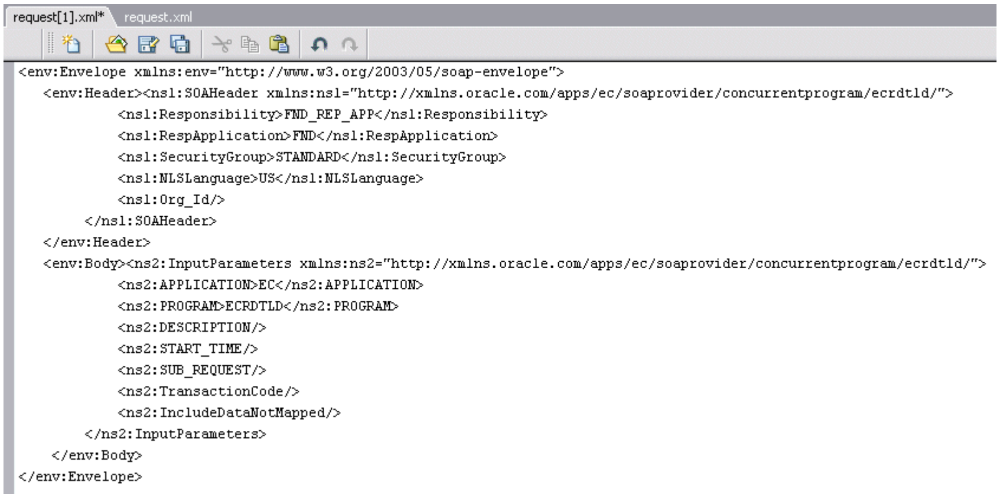
In addition to the general header, the following regions are displayed in the details page:

- **Request Details:** This region contains the SOAP request received date and time, number of attachments, request status, and the view link to view the payload of the SOAP request.

  Click the SOAP Request **View** link if available to view the actual XML file of this request.

  > **Note:** The **View** link appears only if at the time of processing that request, the auditing feature was enabled for the selected interface or service. If it was disabled at the time of processing that request, the **View** link will not appear. The same theory applies to processing SOAP responses as well.

SOAP Request XML File



```xml
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header><ns1:SOAHeader xmlns:ns1="http://xmlns.oracle.com/apps/ec/soaprovider/concurrentprogram/ecrdtld/">
            <ns1:Responsibility>FND_REP_APP</ns1:Responsibility>
            <ns1:RespApplication>FND</ns1:RespApplication>
            <ns1:SecurityGroup>STANDARD</ns1:SecurityGroup>
            <ns1:NLSLanguage>US</ns1:NLSLanguage>
            <ns1:Org_Id/>
        </ns1:SOAHeader>
   </env:Header>
   <env:Body><ns2:InputParameters xmlns:ns2="http://xmlns.oracle.com/apps/ec/soaprovider/concurrentprogram/ecrdtld/">
            <ns2:APPLICATION>EC</ns2:APPLICATION>
            <ns2:PROGRAM>ECRDTLD</ns2:PROGRAM>
            <ns2:DESCRIPTION/>
            <ns2:START_TIME/>
            <ns2:SUB_REQUEST/>
            <ns2:TransactionCode/>
            <ns2:IncludeDataNotMapped/>
        </ns2:InputParameters>
    </env:Body>
</env:Envelope>
```

Additionally, the following regions appear in the Request region if certain conditions are met:

  - **Error Information:** If the request has failure status caused by server fault, the Error Information region appears with the error description and details.

    For error messages, error codes, and possible solutions, see Error Messages, page C-1.

  - **Attachment:** If the SOAP request has attachments associated with it, the Attachment region appears with attachment details including all attachment names and MIME Type information.

- **Response Details:** This region contains the SOAP response sent date and time, number of attachments, and the view link to view the payload of the SOAP response.

  Click the SOAP Response **View** link if available to view the actual XML file of this response.

  > **Note:** The **View** link appears only if at the time of processing that response, the auditing feature was enabled for the selected interface or service. If the auditing feature was turned off at the time of processing that response, the **View** link will not appear. The same theory applies to processing SOAP requests as well.
  >
  > Additionally, if the Interaction Pattern is of type 'Request-Only', the **View** link for response payload is not shown.

To view SOAP request and response details:

1. Log on to Oracle E-Business Suite with the username that has been granted the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

   From the navigation menu, select the **Service Monitor** link from the Administration section to open the Monitor Search page.

2. Perform a search to display the search result. See: Searching SOAP Requests, page 8-3.

3. Click the **Details** icon for a given request to view the SOAP request and response details. The Request and Response Details page appears.

4. Click the SOAP Request or Response **View** link if available to view the actual XML file for the SOAP request or response message.

5. If there is any attachment associated with it, the attachment information appears in the Attachment region.

6. If the SOAP request status is 'Failed', then the Error Information region appears in the Request Details region.

Viewing Service Processing Logs

To effectively monitor SOAP messages at run time during the invocation of Oracle E-Business Suite services by Web service clients, if the runtime logging is enabled for a specific interface or service in the Log & Audit Setup Details page, log messages can be captured in Service Monitor against that instance for the specified service.

When a SOAP request is received, Service Provider generates a unique numeric instance ID based on a database sequence and passes it to Service Monitor. Therefore, each SOAP request in Service Monitor appears with instance ID and the **Log** icon letting you retrieve the log details.

Click the **Log** icon in the search result table to view log messages in the Web Service Runtime Logs page.

> **Important:** Runtime logging for PL/SQL, Concurrent Program, XML Gateway interface types is handled by Oracle SOA Suite; therefore, setting runtime log levels for these services in the Log & Audit Setup Details page will display Oracle SOA Suite logs if the services are deployed in Oracle SOA Suite. Limited runtime log statements from the Oracle E-Business Suite Integrated SOA Gateway code (identified by the package name `oracle.apps.fnd.isg`) will be displayed for these services.
>
> Runtime logging for Business Service Object and Java API for Forms

interface types is handled by Oracle E-Business Suite Integrated SOA Gateway; therefore Service Monitor shows Oracle E-Business Suite Integrated SOA Gateway logs for these interfaces based on the log level selected in the Log & Audit Setup Detail page.

The Web Service Runtime Logs page contains the following log regions:

- **Runtime Middle Tier Logs**

  Runtime Middle tier execution logs are retrieved from Oracle SOA Suite server's file system (***File Logging***) for Oracle E-Business Suite integration. Logs are compiled in a table for a selected service request.

  However, unlike the Adapter logs that are retrieved from the Oracle E-Business Suite table, you can only view the middle tier log details, but you cannot delete this type of logs directly from the Service Monitor user interface.

- **Adapter Logs**

  For Web service execution on the Oracle E-Business Suite side, runtime logs are recorded and retrieved from the log table (FND_LOG_MESSAGES) for Oracle E-Business Suite (***Database Logging***).

  These log messages are compiled and listed in the table format for the selected service in a given instance. Each entry in the table includes log sequence, log timestamp, module, severity level, and actual message.

  Deleting and Exporting Logs in the Adapter Logs Region

  In the Adapter Logs region, after viewing log messages retrieved for a request in a given instance, you can delete them if needed by clicking **Delete Log**. A warning message appears alerting you that this will permanently delete all adapter logs in the table. Click **Yes** to confirm the action. An empty log table appears after all adapter log messages have been successfully deleted.

  Before deleting the logs, you can save a backup copy by clicking **Export**. This exports the records listed in the table to Microsoft Excel and you can use it later.

File Logging and Database Logging

In general, log statements can be captured either in the Oracle SOA Suite server's file system (*File Logging*) or in the Oracle E-Busines Suite database tables (*Database Logging*). By default, log statements are captured in the database if logging is enabled from the Log & Audit Setup Details page.

Enhancing Performance for Database Logging

In comparison to file logging, database logging reduces performance of design-time operations. Performance can be improved by setting the optional parameter `<sid>.ISG_KEEP_ALIVE_DB_CONN=true` in `isgagent.properties` in Oracle E-Buiness Suite.

Enabling File Logging

File logging is enabled by setting the following properties in `isgagent.properties` in Oracle E-Buiness Suite and `isg.properties` in Oracle SOA Suite:

> **Note:** Logging mechanism should be the same across Oracle E-Business Suite and Oracle SOA Suite. If file logging is enabled in Oracle E-Business Suite, then it must be enabled in Oracle SOA Suite as well.

- `<sid>.ISG_GLOBAL_LOG=true`
- `<sid>.ISG_LOGGER=FILE`

In Oracle E-Business Suite, log files are created in the path specified in the property `<SID>.ISG_TEMP_DIRECTORY_LOCATION` in `isgagent.properties` file.

In Oracle SOA Suite server, log file is created in the path specified in the property `<SID>.ISG_TEMP_DIRECTORY_LOCATION` in `$INST_TOP/soa/isg.properties` file.

When file logging is enabled, log statements for design-time and runtime operations are not shown in the Interface Details page and Service Monitor user interfaces.

For information on log severity level and how to configure logs, see Adding a New Logging Configuration, page 7-6.

For information on how to view logs and error messages recorded during service generation and deployment at design time, see Viewing Generate and Deploy Time Logs, page 3-24.

To view log messages in Service Monitor:

1. Log on to Oracle E-Business Suite with the username that has been granted with the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

   From the navigation menu, select the **Service Monitor** link from the Administration section to open the Monitor Search page.

2. Perform a search to display the search result. See: Searching SOAP messages, page 8-3.

3. In the search result table, click the **Log** icon for a desired instance. The Web Service Runtime Logs page is displayed allowing you to view the log details.

4. In the Adapter Logs region, click **Delete Log** to delete all the logs listed in the table for a given instance if needed. Click **Yes** to confirm the action. Click **No** to return to the Web Service Runtime Logs page.

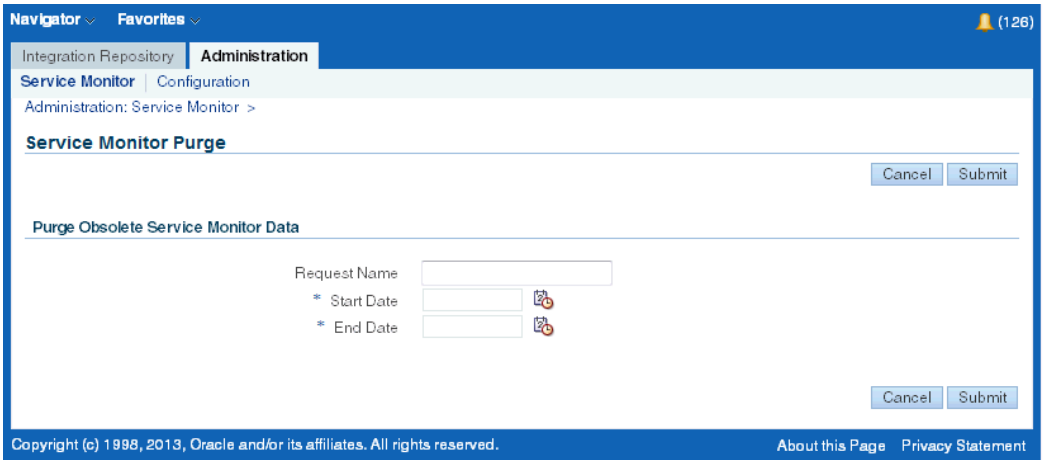
   Click **Export** to export log list table to Microsoft Excel.

Purging SOAP Messages, Audits, and Logs

Oracle E-Business Suite Integrated SOA Gateway allows you to purge SOAP messages, logs stored in the Oracle E-Business Suite database, and audit records that have been collected through Service Monitor for a period of time. Click **Purge** in the Service Monitor Search page to launch the Service Monitor Purge page.

> **Note:** For log messages retrieved from Oracle SOA Suite's server file system, these log messages cannot be purged. For log messages coming from Oracle E-Business Suite API execution stored in the log message table, these logs can be purged from the Oracle E-Business Suite database.

Service Monitor Purge Page



Enter the following purge parameters in the Service Monitor Purge page:

- **Request Name**: Specify the Request Name for your request.
- **Start Date**: Identify the start date of the date range for your purge.
- **End Date**: Identify the end date of the date range for your purge.

Click **Submit**. A request number will be automatically assigned to you for your purge request indicating that your request has been submitted for processing. When your request is executed, all SOAP requests within your specified date range will be purged.

The monitored SOAP requests and responses will be purged in the following order of sequence:

1. Purging SOAP requests

   This deletes all SOAP requests for the specified date range.

2. Purging SOAP body

   This deletes the SOAP body including payload corresponding to those SOAP requests that have been purged (for the specified date range).

3. Purging SOAP attachment

   This deletes all attachments associated with the SOAP requests and responses for the specified date range.

4. Purging log messages from the Oracle E-Business Suite database

   This deletes only the logs for which the service is completed with a status of 'SUCCESS'. This does not delete the logs for the service with 'FAILURE' status.

   The purge is based on the Completion Date of the service for the specified date range.

5. Purging composite instances from Oracle SOA Suite

   This deletes composite instances from Oracle SOA Suite for the specified date range.

To purge SOAP requests and responses:

1. Log on to Oracle E-Business Suite with the username that has been granted with the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

   From the navigation menu, select the **Service Monitor** link from the Administration section to open the Monitor Search page.

2. Click **Purge**.

3. Enter the following information in the Service Monitor Purge page:

   1. Enter the request name for your purge request.

   2. Enter the Start Date and End Date fields to specify the time range for your purge.

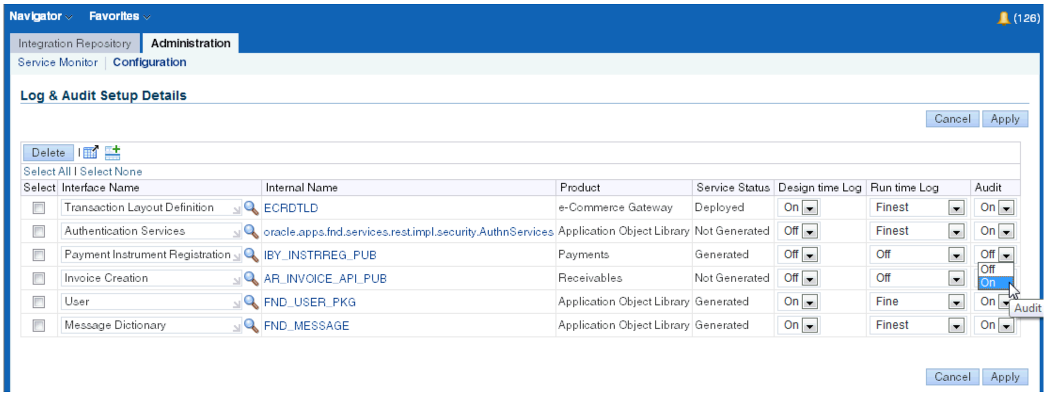
4. Click **Submit** to submit your purge request.

Enabling Web Service Auditing Using the Configuration Subtab

In addition to searching and viewing SOAP requests and responses, Service Monitor provides auditing feature allowing you to track SOAP message details such as requests, responses, and faults.

If the auditing feature for a specific service is enabled, all incoming SOAP requests and corresponding responses for the service that Oracle SOA Suite processes along with the associated payloads, and fault messages can be saved and tracked in Service Monitor.

Please note that the auditing feature is enabled at the interface level through the Log & Audit Setup Details page in the Configuration subtab.



Log & Audit Setup Details

Select	Interface Name	Internal Name	Product	Service Status	Design time Log	Run time Log	Audit
□	Transaction Layout Definition	ECRDTLD	e-Commerce Gateway	Deployed	On	Finest	On
□	Authentication Services	oracle.apps.fnd.services.rest.impl.security.AuthnServices	Application Object Library	Not Generated	Off	Finest	On
□	Payment Instrument Registration	IBY_INSTRREG_PUB	Payments	Generated	Off	Off	Off
□	Invoice Creation	AR_INVOICE_API_PUB	Receivables	Not Generated	Off	Off	Off On
□	User	FND_USER_PKG	Application Object Library	Generated	On	Fine	On
□	Message Dictionary	FND_MESSAGE	Application Object Library	Generated	On	Finest	On

To enable the auditing feature, select the interface that you want the feature to be enabled, and then select 'On' from the Audit drop-down list. Click **Apply** to save and validate the addition.

For more information on how to enable the auditing feature along with log configuration at the interface level, see Adding a New Configuration, page 7-6.

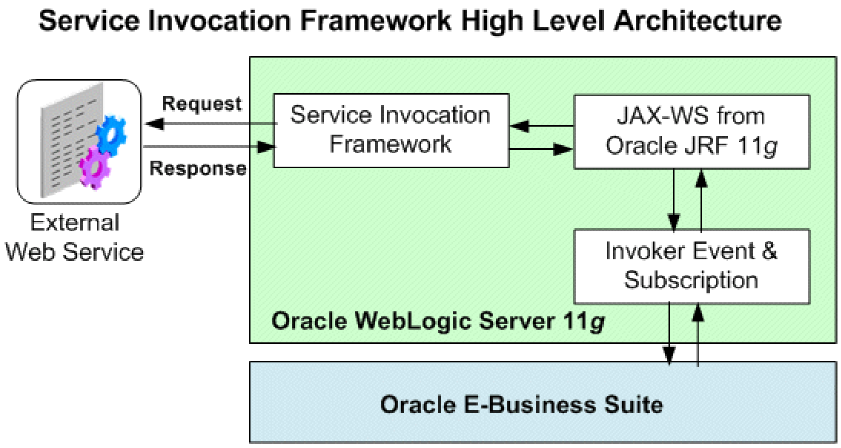
9

Implementing Service Invocation Framework

Overview

To invoke and consume Web services from Oracle E-Business Suite, Oracle E-Business Suite Integrated SOA Gateway uses service invocation framework (SIF) that leverages Oracle Workflow Java Business Event System (JBES) and a seeded Java rule function to allow services described in WSDL to be invoked.

> **Note:** Service invocation framework from Oracle E-Business Suite is enabled though Oracle Workflow Java Business Event System and is based on the JAX-WS (Java API for XML-based Web Services) Dispatch from Oracle JRF (Java Required Files) 11*g*.
>
> Previously, SIF was based on the Web Services Invocation Framework (WSIF) provided in Oracle Application Server 10*g*. In this release, SIF leverages the JAX-WS (Java API for XML-based Web Services) Dispatch from Oracle JRF 11*g* as shown in the following high level architecture diagram:


Service Invocation Framework High Level Architecture
Request
Response
External Web Service
Service Invocation Framework
JAX-WS from Oracle JRF 11*g*
Invoker Event & Subscription
Oracle WebLogic Server 11*g*
Oracle E-Business Suite

By using this service invocation framework, developers or implementers can interact with Web services through WSDL descriptions instead of working directly with SOAP APIs. This approach allows you to access Web services in a manner that is independent of protocol or location.

This invocation framework used in Oracle E-Business Suite allows updated implementations of a binding to be plugged at run time. As a result, it not only facilitates a stubless or completely dynamic Web service invocation, but also allows the calling service to defer choosing a service binding until run time. More importantly, this enhances the seamless business integration between loosely coupled applications and accelerates service execution and consumption.

Please note that the service invocation framework discussed here only supports document-based Web service invocation. Oracle E-Business Suite Integrated SOA Gateway does not support RPC (remote procedure call) style Web service invocation through this invocation framework.

> **Note:** The document-based Web service uses the form of XML with commonly agreed upon schema between the service provider and consumer as a communication protocol. While RPC-based Web service is to invoke a cross-platform remote procedure call using SOAP.

To have a better understanding on how the service invocation framework invokes Web services, the following topics are described in this chapter:

- Service Invocation Framework Architecture Overview, page 9-3
- Understanding Service Invocation Framework Major Features, page 9-6
- Implementing Service Invocation Framework, page 9-7

Service Invocation Framework Architecture Overview

Oracle Workflow is the primary process management solution within Oracle E-Business Suite; Oracle Workflow Business Event System, an essential component within Oracle Workflow, provides event and subscription features that help identify integration points within Oracle E-Business Suite.

The Business Event System consists of an Event Manager and workflow process event activities. The Event Manager lets you register subscriptions to significant events; event activities representing business events within workflow processes let you model complex business flows or logics within workflow processes.

When an event occurs, the Event Manager executes subscription to the event. Subscription processing can include executing custom code on the event information, sending event information to a workflow process, and sending event information to other agents or systems.

For example, to invoke a Web service through Oracle Workflow JBES, the description of WSDL URL representing the Web service must be consumed through the event subscription definition so that Web service metadata can be parsed and stored as subscription parameters.

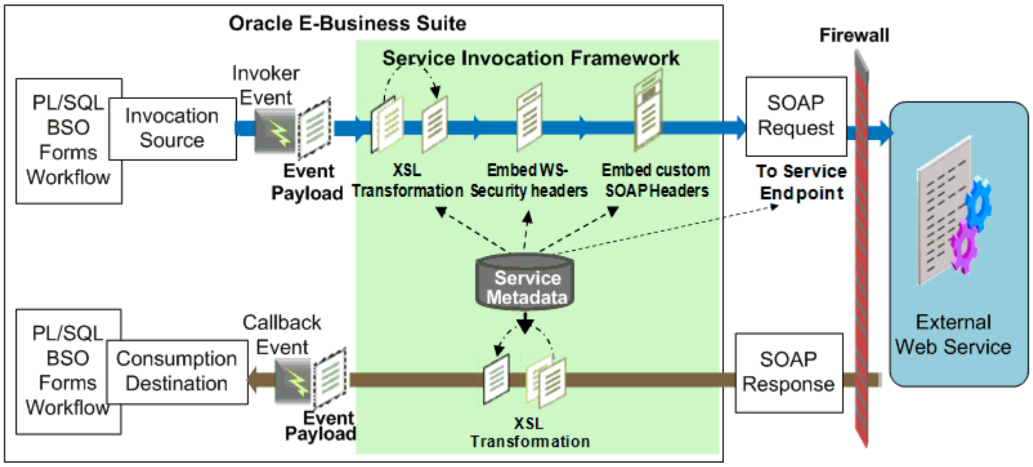
> **Note:** By leveraging Oracle Workflow Java Business Event System (JBES), service invocation framework allows almost all forms of Web services representing in WSDL URLs to be invoked from Oracle E-Business Suite.

At run time, when an invoker event is raised, the event and subscription parameters are used to invoke a Web service by sending a SOAP request message. If this request or output message requires transformation in order to communicate with an external Web service, the XSL transformation on the output message is performed before invoking the service. If it is a synchronous request - response operation and the response is available, the XSL transformation on the input message can be performed if necessary in order to communicate or callback to Oracle E-Business Suite.

> **Note:** If event parameters are passed with the same names as the subscription parameters that have been parsed and stored, the event parameter values override the subscription parameters.

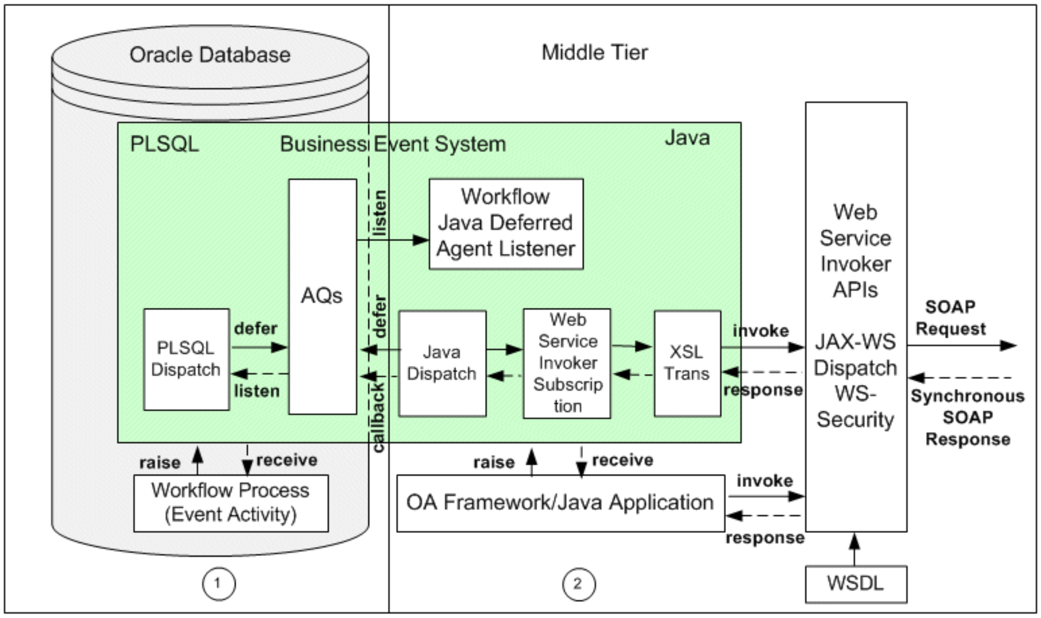
This run time service invocation process can be illustrated in the following diagram:

Service Invocation Framework Transactional Architecture



To better understand how the invocation process takes place and its relationship between Oracle Workflow components, the following architecture diagram provides the topology of various components that exchange information during the end-to-end service invocation from Oracle Workflow:

Service Invocation Framework Architecture



Oracle Workflow Business Event System is a workflow component that allows events to be raised from both PL/SQL and Java layers. Therefore, the service invocation from Oracle E-Business Suite can be from a PL/SQL or Java layer.

1. **Service Invocation from PL/SQL**

   1. Application raises a business event using PL/SQL API `WF_EVENT.Raise`.

The event data can be passed to the Event Manager within the call to the `WF_EVENT.Raise` API, or the Event Manager can obtain the event data or message payload by calling the generate function for the event if the data or payload is required for a subscription.

> **Note:** See the *Oracle Workflow API Reference* for information about `WF_EVENT.Raise` API.

2. Oracle Workflow Business Event System (BES) identifies that the event has a subscription with Java Rule Function `oracle.apps.fnd.wf.bes.WebServiceInvokerSubscription`.

3. The Business Event System enqueues the event message to the WF_JAVA_DEFERRED queue. The Java Deferred Agent Listener then dequeues and executes the subscription whose Java rule function invokes the Web service.

4. If callback event and agent parameters are mentioned, the Web service response is communicated back to Oracle E-Business Suite using the callback information. The Java Deferred Agent Listener process that runs in the Concurrent Manager (CM) tier invokes the Web service.

2. **Service Invocation from Java**

   1. Java Application raises a business event using Java method `oracle.apps.fnd.wf.bes.BusinessEvent.raise` either from OA Framework page controller/AMImpl or Java code running on the Concurrent Manager (CM) tier.

   2. Since the event is raised in Java where the subscription's seeded Java Rule Function `oracle.apps.fnd.wf.bes.WebServiceInvokerSubscription` is accessible, whether the rule function is executed inline or deferred is determined by the phase of the subscription.

      - If the invoker subscription is created with Phase >= 100, the event is enqueued to the WF_JAVA_DEFERRED queue.

      - If the invoker subscription is created with Phase < 100, the event is dispatched inline.

        If the event is raised from OA Framework page, the dispatch logic executes within `OACORE WebLogic Server`.

After an event is raised either using the PL/SQL API or Java method, the raised event can be processed in the following ways:

- If the raised event is dispatched immediately to the Java Business Event System, then the seeded Java rule function and its associated event subscription information will be retrieved and executed to invoke the Web service.

- If the raised event is enqueued to the WF_JAVA_DEFERRED queue, then Java Deferred Agent Listener running on concurrent tier will dequeue the event message and then dispatch the event to the Java Business Event System. The seeded Java rule function and its associated event subscription information will then be retrieved and executed to invoke the Web service.

While invoking the Web service, the seeded Java rule function first reads the Web service metadata created for the subscription.

If Web service input message requires transformation, the Java rule function performs XSL transformation on the request message generated during the event creation by using a PL/SQL API `ECX_STANDARD.perform_xslt_transformation`. Next, the Java rule function invokes the service.

> **Note:** For detailed information on the XSL transformation PL/SQL API, see Execution Engine APIs, *Oracle XML Gateway User's Guide*.

If it is for the synchronous request - response operation, when the response message is available and XSL transformation is required on the Web service output message, XSL transformation on the output (response) message will be performed.

If callback information is provided, perform callback by either raising a business event or by enqueuing the event to a given workflow agent with the response message as payload.

> **Note:** For the service invocation from Java code, if the Web service invoker subscription is synchronous with subscription phase < 100, then the Web service is invoked as soon as the event is raised, and if it's successful, the response is available immediately by using method `getResponseData()` on the `BusinessEvent` object.

Service Invocation Framework Major Features

Service Invocation Framework has the following features:

- It supports various service invocation sources or points from an Oracle E-Business Suite instance. This includes
  - PL/SQL Layer
    - Workflow Process
    - Any other PL/SQL code

        - Forms

    - Java Layer
        - OA Framework

        - Standalone Java Code

- It supports the Synchronous Request - Response, and One-way/Notification Only message patterns in WSDL.

- It supports SSL-based Web service invocation over HTTPS protocol.

- It supports Web Service (WS) security through UsernameToken-based Web Service authentication.

- It supports passing values for any header part that may be required to embed applications context into SOAP envelopes.

- It provides errors and exception handling, and the invocation retry feature.

- It provides the ability to test business event for service invocation.

Implementing Service Invocation Framework

This section discusses the following topics:

- Setup Tasks, page 9-7

- Setup Tasks for Invoking SSL-based Web Services Over HTTPS, page 9-9

- Implementing Service Invocation Framework, page 9-13

Setup Tasks

Web services can be invoked from any one of the following tiers:

- **OACORE WebLogic Server**: Web service invocations from OA Framework page using a synchronous event subscription (phase < 100) is executed from the OACORE WebLogic Server.

- **Concurrent Manager (CM) Tier JVM**: The following Web service invocations are executed from CM tier JVM within Java Deferred Agent Listener that runs within Workflow Agent Listener Service:
    - Invocations from PL/SQL either through synchronous or asynchronous event

    subscriptions

    - Invocations from Java/OA Framework through asynchronous event subscriptions

- **Standalone JVM**: Web service invocations from a Java process that runs outside OACORE or CM using a synchronous event subscription are executed from within that JVM.

Proxy Host and Port Setups

If a target Web service resides within the firewall and is directly accessible from an Oracle E-Business Suite server, administrators do not need to configure proxy host and port.

However, if a target Web service that is invoked resides outside the firewall and thus the request needs to be routed through the proxy, in this circumstance, administrators must set up and configure proxy host and port appropriately for the tiers that Web service invocations occur in order to perform the following activities:

- Parse and consume WSDL during subscription definition
- Invoke Web service from subscription definition

Common Proxy Setup at WebLogic Server and Concurrent Manger Tier JVM

Use common setup information to configure proxy host and port. This information is applicable to the following conditions:

- **Proxy host and port at WebLogic Server**

  For a Web service invoked from OA Framework, the JBES seeded Java rule function would run within the OACORE's WebLogic Server.

  WebLogic Server start script (`<EBSDomain>/bin/startWebLogic.sh`) should have the following system properties setup in the `JAVA_OPTIONS` in order for it to work:

  ```
  -Dhttp.proxyHost=myproxy.host.name

  -Dhttp.proxyPort=80

  -Dhttp.nonProxyHosts=*.mydomain.com|localhost
  ```

- **Proxy host and port at Concurrent Manger Tier JVM**

  For a Web service invoked from PL/SQL and Java using an asynchronous subscription, the event is raised by the application code wherever it executes and then enqueued to the WF_JAVA_DEFERRED queue by the Event Manager. The event subscription is executed from the CM tier by the Java Deferred Agent Listener.

  If a Web service is invoked by the Java Deferred Agent Listener, then the code would run within the CM tier Java service's JVM. If the Web service resides outside the firewall, proxy host and port need to be configured properly.

To configure proxy host and port for WebLogic server and CM tier JVMs, you need to update AutoConfig context file with the following entries and run AutoConfig:

```
<!-- proxy -->
  <proxyhost oa_var="s_proxyhost">myproxyhost</proxyhost>
   <proxyport oa_var="s_proxyport">80</proxyport>
   <nonproxyhosts oa_var="s_nonproxyhosts">any domain that needs to be
by-passed (such as *.us.oracle.com)</nonproxyhosts>
```

Proxy Host and Port Setup When Using Standalone Java Class

You must set the following entries:

```
java -Dhttp.proxyHost=myproxyhost -Dhttp.proxyPort=80 classname
```

Setup Tasks for Invoking SSL-based Web Services over HTTPS

Service Invocation Framework supports SSL-based Web service invocation using Server Authentication method. When a client connects to a Web server via HTTPS, the server sends back its server certificate to the client for verification. Once verified, the client sends the data, encrypted, to the server. Server Authentication allows the client to identify the server. Before invoking a Web service from a server over HTTPS (HTTP protocol over TLS/SSL), you need to perform manual setup tasks in order to read SSL-based WSDLs and invoke SSL service endpoints.

A client may receive one of the following two types of server certificates:

- Public certificate and it is issued by a Certification Authority (CA)

- Self-signed certificate or certificate is not in trusted certificate list

Perform the following two setup tasks for the Service Invocation Framework to invoke a SSL-based Web service:

- Importing Server SSL Certificate into a SIF JVM's Certificate Store, page 9-9

- Setting Up SSL Proxy Host and Port, page 9-12

Importing Server SSL Certificate into a SIF JVM's Certificate Store

Public Certificate Issued by a Certification Authority (CA)

If server certificate is a public certificate and is issued by a public CA such as VeriSign, then it is most likely available in a SIF JVM's certificate store or in a trusted certificate list.

Self-signed Certificate or Certificate is not in Trusted Certificate List

Perform the following tasks to import the server's SSL certificate into a SIF JVM's certificate store or add it to a trusted certificate list:

1. **Export** the server certificate using either one of the following methods:

   - **Use`openssl`Utility:**

     Use **`openssl`** utility to connect to the destination server with the following syntax:

     ```
     $ openssl s_client -connect <server>:<port> -showcerts
     ```

     > **Important:** If there is no port in destination, default HTTPS port 443 should be used.
     >
     > For example: `$ openssl s_client -connect host.domain.com:443 -showcerts`

     Copy the certificate content from `BEGIN CERTIFICATE` to `END CERTIFICATE` (including BEGIN CERTIFICATE and END CERTIFICATE lines as shown in the sample certificate) into a file and save the file (such as `my_cert.cer`).

     A sample output of above **`openssl`** command can be like:

     ```
     $ openssl s_client -connect host.domain.com:443 -showcerts

     ...
     Server certificate
     -----BEGIN CERTIFICATE-----
     MIIFVjCCBD6gAwIBAgIQBVWzfUyIcCa5LtuV+f9WvjANBgkqhkiG9w0BAQUFADCB
     sDELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQL
     ExZWZXJpU2lnbiBUcnVzdCBOZXR3b3JrMTswOQYDVQQLEzJUZXJtcyBvZiB1c2Ug
     YXQgaHR0cHM6Ly93d3cudmVyaXNpZ24uY29tL3JwYSAoYykwNTEqMCgGA1UEAxMh
     VmVyaVNpZ24gQ2xhc3MgMyBTZWN1cmUgU2VydmVyIENBMB4XDTA5MDQyMTAwMDAw
     MFoXDTEwMDUwNTIzNTk1OVowgckxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxp
     Zm9ybmlhMRcwFQYDVQQHFA5SZWR3b29kIFNob3JlczEbMBkGA1UEChQST3JhY2xl
     IENvcnBvcmF0aW9uMR8wHQYDVQQLFBZJbmZvcm1hdGlvbiBUZWNobm9sb2d5MTMw
     MQYDVQQLFCpUZXJtcyBvZiB1c2UgYXQgd3d3LnZlcmlzaWduLmNvbS9ycGEgKGMp
     MDUxGTAXBgNVBAMUECoub3JhY2xlY29ycC5jb20wgZ8wDQYJKoZIhvcNAQEBBQAD
     gY0AMIGJAoGBAL/EBxxt2keWTuJbo4SogWmiaJxThYDMvy8nWkpvKIp3s7OCQW0G
     t17sAirfBkUirbGRlcWi5fi0RReruGXgYxFnf12fBNAimRWVo3mjeQo8BpRBm27n
     3YcTZUlaIE77FvB3913jzD9c4sbjIe2fGpVmx+X9PZmDKSY9KPGjDbFNAgMBAAGj
     ggHTMIIBzzAJBgNVHRMEAjAAMAsGA1UdDwQEAwIFoDBEBgNVHR8EPTA7MDmgN6A1
     hjNodHRwOi8vU1ZSU2VjdXJlLWNybC52ZXJpc2lnbi5jb20vU1ZSU2VjdXJlMjAw
     NS5jcmwwRAYDVR0gBD0wOzA5BgtghkgBhvhFAQcXAzAqMCgGCCsGAQUFBwIBFhxo
     dHRwczovL3d3dy52ZXJpc2lnbi5jb20vcnBhMB0GA1UdJQQWMBQGCCsGAQUFBwMB
     BggrBgEFBQcDAjAfBgNVHSMEGDAWgBRv7K+g3Yqk7/UqEGctP1WCvNfvJTB5Bggr
     BgEFBQcBAQRtMGswJAYIKwYBBQUHMAGGGGh0dHA6Ly9vY3NwLnZlcmlzaWduLmNv
     bTBDBggrBgEFBQcwAoY3aHR0cDovL1NWUlNlY3VyZS1haWEudmVyaXNpZ24uY29t
     L1NWUlNlY3VyZTIwMDUtYWlhLmNlcjBuBggrBgEFBQcBDARiMGChXqBcMFowWDBW
     FglpbWFnZS9naWYwITAfMAcGBSsOAwIaBBRLa7kolgYMu9BSOJsprEsHiyEFGDAm
     FiRodHRwOi8vbG9nby52ZXJpc2lnbi5jb20vdnNsb2dvMS5naWYwDQYJKoZIhvcN
     AQEFBQADggEBADBi9NfljQLuD2Tnol3pmQl717rc8kKmpLYEO6u5MxIK0+L2MslV
     4NE1qbNx1dfIoW68HHXtpsF5KtKFLYk9EoOkBd7oMp7fFv31RANV3LpdAHZC9EaK
     CA/oKB2RrSu7ZmaUvoRb+3v5FdhAmgtoY6Wljk0yxMvXVf/TOeXqKl8C/r1gSzyC
     s/jVmy6N81Oeleqtozzt/aJNGu7xu/MdtP13eyu7RSEBRGJwEwTXH+rTUKK8mle0
     Kz15DgJ6ByK2XZmD4Z+O8DTUhUhIHR1OhuLR7zjGp9W7wQuCizUcTvuKEGzVf5D/
     y7orhV0U+AoXnl/5wntVMZc/Tmqr/Fkb8+g=
     -----END CERTIFICATE-----

     ...
     ```

- **Use Web Browser:**

  Access the WSDL file available through HTTPS URL (such as `https://<hostname>:<port>/webservices/SOAProvider/xmlgateway/ont__poi/?wsdl`) through a Web browser.

  1. After the WSDL file has been successfully loaded in a browser, double click on the Lock icon in the bottom right corner of the browser and export the certificate.

     For example, in Internet Explorer, double click on the Lock icon > Details > Copy to File. In Mozilla Firefox, double click on the Lock icon > Security > View Certificate > Details > Export.

  2. You can also use browser menu to access the certificate. For example, in Internet Explorer, select **Internet Options** from the **Tools** drop-down menu to open the Internet Options pop-up window. Select the Content tab, click **Certificates**. Select the Personal (or Other People) tab to select your certificate and click **Export**.

  3. You can export or save the certificate either in DER encoded binary X.509 (.CER) or in Base 64 encoded.

     > **Note:** Different browser versions may have different steps to export SSL certificates.

2. **Import** the server's SSL certificate into an appropriate SIF JVM's certificate store to add it to the list of trusted certificates.

   > **Important:** Information about where Web services are invoked through the Service Invocation Framework is described in the Setup Tasks, page 9-7.

There are many utilities available to import certificates. For example, you can use **`keytool`**, a key and certificate management utility that stores the keys and certificates in a *keystore*. This management utility is available by default with JDK to manage a *keystore* (database) of cryptographic keys, X.509 certificate chains, and trusted certificates.

The **`keytool`** commands have the following syntax:

```
keytool -import -trustcacerts -keystore <key store location>
-storepass <certificate store password> -alias <alias name>
-file <exported certificate file>
```

For example:

```
keytool -import -trustcacerts -keystore
"$AF_JRE_TOP/jre/lib/security/cacerts" -storepass password
```

```
-alias xabbott_bugdbcert -file my_cert.cer
```

Note: This must be typed as a single line. The file (`-file`) is the exported certificate file i.e. my_cert.cer.

Setting Up SSL Proxy Host and Port

If a SSL-based Web service resides outside the firewall, the JVM that invokes the Web service has to communicate through SSL proxy. Following setup tasks are required in all appropriate tiers to use SSL proxy.

Setting Up Proxy Host and Port at WebLogic Server

For a Web service invoked from OA Framework, the JBES seeded Java rule function would run within the OACORE's WebLogic Server.

WebLogic Server start script (`<EBSDomain>/bin/startWebLogic.sh`) should have the following system properties setup in the `JAVA_OPTIONS` in order for it to work:

```
-Dhttps.proxyHost=myproxy.host.name
```

```
-Dhttps.proxyPort=80
```

```
-Dhttps.nonProxyHosts=*.mydomain.com|localhost
```

AutoConfig does not support properties `https.proxyHost` and `https.proxyPort` currently. To ensure the above properties are retained during the execution of AutoConfig, the context file could be customized to add these two properties.

For information on how to customize AutoConfig-managed configurations, see *Using AutoConfig to Manage System Configurations in Oracle E-Business Suite Release 12*, My Oracle Support Knowledge Document 387859.1 for details.

Setting Up Proxy Host and Port at Concurrent Manger Tier JVM

For a Web service invoked from PL/SQL and Java using an asynchronous subscription, the event is raised by the application code wherever it executes and then enqueued to the WF_JAVA_DEFERRED queue by the Event Manager. The event subscription is executed from the CM tier by the Java Deferred Agent Listener.

If a Web service is invoked by the Java Deferred Agent Listener, then the code would run within the CM tier Java service's JVM. Workflow Agent Listener Service does not currently support Service Parameters to set SSL proxy. The SSL proxy could be set up directly to Concurrent Manager's JVM system properties in `$APPL_TOP/admin/adovars.env` using AutoConfig.

```
<oa_environment type="adovars">
 <oa_env_file type="adovars" oa_var="s_adovars_file" osd="unix">
  $APPL_TOP/admin/adovars.env</oa_env_file>
...
 <APPSJREOPTS oa_var="s_appsjreopts">="-Dhttps.proxyHost=[proxyhost]
  -Dhttps.proxyPort=[sslproxyport]</APPSJREOPTS>
...
</oa_environment>
```

Setting Up Proxy Host and Port When Using Standalone Java Class

You must set the following entries:

```
java -Dhttps.proxyHost=[proxyhost] -Dhttps.proxyPort=[sslproxyport]
<classname>
```

Implementing Service Invocation Framework

The invocation of Oracle E-Business Suite Web services using the Service Invocation Framework involves the following steps:

- Defining invocation metadata and invoking Web services through the Business Event System
- Calling back to Oracle E-Business Suite with Web service responses
- Managing errors
- Testing Web service invocation
- Extending Web service invocation

Defining Invocation Metadata and Invoking Web Services Through the Business Event System

Web service invocation metadata can be defined by using Oracle Workflow Business Event System to create events and event subscriptions. When a triggering event occurs, a Web service can be invoked through an appropriate event subscription.

Specifically, the invocation metadata can be defined through the following steps:

- Create a Web service invoker event, page 9-13
- Create a local subscription to invoke a Web service, page 9-13
- Create an error subscription to enable error processing, page 9-19
- Create a receive event (optional), page 9-19
- Create a receive event subscription (optional), page 9-20

- **Create a Web service invoker event**

  A business event that serves as a request message for a service needs to be created first.

- **Create a local subscription to invoke a Web service**

  You must subscribe to the invoker event with 'Invoke Web Service' action type.

  To create an event subscription to the Invoker event, enter basic subscription information (such as source type, phase, event filter) and select 'Invoke Web

Service' action type. Click **Next** to display the Invoke Web Service wizard where you can specify a WSDL URL as an input parameter for the event subscription. The Business Event System then parses the given WSDL and displays all services contained in the WSDL for selection.

This parsing feature allows developers to select appropriate service metadata including service port, port type, and operation for a selected service and then stores the selected information as subscription parameters that will be used later during service invocation.

Configuring UsernameToken based WS-Security

If the Web service being invoked enforces Username/Password based authentication, then the Service Invocation Framework also supports the UsernameToken based WS-Security header during Web service invocation.

> **Important:** This UsernameToken based WS-Security header is implemented during the service invocation only if the Web service provider that processes the Web service request needs this security header.

To authenticate the users invoking Web services, the UsernameToken based WS-Security model passes a *username* and an optional *password* in the SOAP Header of a SOAP request sent to the Web service provider.

After specifying needed information in the Invoke Web Service wizard during the creation of event subscription, the Web Service Security region is displayed letting you enter username and password required for authentication.

> **Note:** The security information is now entered through the Web Service Security region which replaces the security parameters (WFBES_SOAP_USERNAME, WFBES_SOAP_PASSWORD_MOD, and WFBES_SOAP_PASSWORD_KEY) used in earlier releases.

Specifying a Subscription Parameter for WS-Security Header

When creating the subscription to the Invoker event, you can specify the following parameter to set the expiration time for the security header:

- WFBES_SOAP_EXPIRY_DURATION

By default, the header is set to expire 60 seconds in the `<wsu:Timestamp>` element (with `<wsu:Created>` and `<wsu:Expires>`) after it is created. You can optionally use this parameter to set a different expiration time in seconds for the header. This helps protect the header from being reused during a replay attack.

For example, a WS-Security header can be like:

```
<wsse:Security soapenv:mustUnderstand="1"

xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
  <wsu:Timestamp wsu:Id="Timestamp-2"

xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-w
ssecurity-utility-1.0.xsd">
   <wsu:Created>2013-09-02T04:56:59.592Z</wsu:Created>
   <wsu:Expires>2013-09-02T04:57:59.592Z</wsu:Expires>
  </wsu:Timestamp>
 <wsse:UsernameToken wsu:Id="UsernameToken-1"

xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-w
ssecurity-utility-1.0.xsd">
  <wsse:Username>myUser</wsse:Username>
  <wsse:Password

Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-userna
me-token-profile-1.0#PasswordText">password</wsse:Password>
  <wsse:Nonce

EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-ws
s-soap-message-security-1.0#Base64Binary">RDyVo/jbXJdSKuVEPrQW6Q==</
wsse:Nonce>
  <wsu:Created>2013-09-02T04:56:48.597Z</wsu:Created>
 </wsse:UsernameToken>
</wsse:Security>
```

> **Note:** In the `<wsse:UsernameToken>` element, `<Nonce>` provides random string for the password which helps protect the UsernameToken security from being reused. `<Created>` indicates the creation time of the UsernameToken security.

For more information about UsernmaeToken security, see UsernameToken Based Security, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

Configuring Security Information with Customization Level

Oracle Workflow allows various levels of updates on business event and subscription based on the customization level. If the Invoke Web Service event subscription's customization level is Core or Limit, and if the username is supplied by the subscription owner, then the username cannot be updated. If the username was not already supplied, you can update it if required. Password can always be updated if it's needed regardless of the customization level.

For more information about customization level and how to configure security parameters, see Configuring Web Service Security Through Event Subscription User Interface, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

Specifying Subscription Parameters for XSL Transformation

While defining a local subscription to the Invoker event, you can specify the following message transformation parameters to support XSL transformation:

- WFBES_OUT_XSL_FILENAME
- WFBES_IN_XSL_FILENAME

Parameters to Set Values for Input Parts

Two topics are discussed in this section:

- Event Payload as SOAP Body, page 9-16
- Setting Other Web Service Input Message Parts, page 9-17

Event Payload as SOAP Body

Because the seeded Java rule function accepts SOAP body part value through business event payload, that payload can be passed in either one of the following ways:

- Event data or payload is passed through the Generate Function during the event raise.
- Event data or payload is passed along with the event itself without using the Generate function.

After the event data or payload is passed, if the XML payload is available at the time of invoking the Web service and the payload is required to be transformed into a form that complies with the input message schema, the seeded Java rule function performs XSL transformation on the XML payload and then invokes the service.

> **Note:** An input message is the XML payload that is passed to the Web service in the SOAP request. An output message is the XML document received as a response from the Web service after a successful invocation.

Message Transformation Parameters to Support XSL Transformation

For the synchronous request - response operation, when the output (response) message, an XML document, is available, if this XML document requires to be transformed to a form that is easier for Oracle E-Business Suite to understand, then XSL transformation on the output message will be performed.

The following subscription parameters are used to pass the XSL file names to the seeded Java rule function for XSL transformation:

> **Note:** The XSL file name is structured with the following format:
>
> `<filename>:<application_short_name>:<version>`
>
> For example, it can be like "`PO_XSL_OUT_2.xsl:FND:1.1`".

- WFBES_OUT_XSL_FILENAME: XSL file to perform transformation on the output (response) message

  For example, it can be like `WFBES_OUT_XSL_FILENAME=PO_XSL_OUT_2.xsl:FND:1.1`.

- WFBES_IN_XSL_FILENAME: XSL file to perform transformation on the input message

  For example, it can be like `WFBES_IN_XSL_FILENAME=PO_XSL_IN_2.xsl:FND:1.1`.

At run time, a triggering event can be raised either from a PL/SQL layer using a PL/SQL API `WF_EVENT.Raise` or from a Java layer using a Java method `oracle.apps.fnd.wf.bes.BusinessEvent.raise` through the Business Event System.

If event parameters are passed with the same names, then the event parameters override the subscription parameters. For example, the event parameters are passed as follows:

- `BusinessEvent.setStringProperty("WFBES_OUT_XSL_FILENAME", "PO_XSL_OUT_2.xsl:FND:1.1");`

- `BusinessEvent.setStringProperty("WFBES_IN_XSL_FILENAME", "PO_XSL_IN_2.xsl:FND:1.1");`

For more information on Web service security and message payload, see the *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

Setting Other Web Service Input Message Parts

Apart from passing the SOAP body part as event payload, Service Invocation Framework supports passing values for other parts that are defined for the Web service operation's input message.

For example, consider the operation `PROCESSPO` in Oracle E-Business Suite XML Gateway service (`http://<hostname>:<port>/webservices/SOAProvider/xmlgateway/ont__poi/?wsdl`) as described below.

```
<definitions targetNamespace="ONT__POI"
targetNamespace="http://xmlns.oracle.com/apps/ont/soaprovider/xmlgat
eway/ont__poi/">
<type>
   <schema elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/apps/ont/soaprovider/xmlgat
eway/ont__poi/">
    <include
schemaLocation="http://<hostname>:<port>/webservices/SOAProvider/xml
gateway/ont__poi/PROCESS_PO_007.xsd"/>
   </schema>
...
<message name="PROCESSPO_Input_Msg">
  <part name="header" element="tns:SOAHeader"/>
   <part name="body" element="tns1:PROCESS_PO_007"/>
</message>
...
<binding name="ONT__POI_Binding" type="tns:ONT__POI_PortType">
<soap: binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="PROCESSPO">
  <soap:operation
soapAction="http://<hostname>:<port>/webservices/SOAProvider/xmlgate
way/ont__poi/"/>
  <input>
   <soap:header message="tns:PROCESSPO_Input_Msg" part="header"
use="literal"/>
   <soap:body parts="body" use="literal"/>
   </input>
  </operation>
</binding>
...
</definitions>
```

The operation PROCESSPO requires input message PROCESSPO_Input_Msg, which has two parts:

- Body: The value of PROCESS_PO_007 type to be set as SOAP body is sent as business event payload.

- Header: The value of SOAHeader type to be sent in the SOAP header which is required for Web Service authorization can be set by using the business event parameter with the following format:

  WFBES_INPUT_<partname>

  <partname> is the same part name in the input message definition in WSDL.

For example, the header part for above example is passed to business event as parameter WFBES_INPUT_header during the invoker event raise. The following code snippet shows the header part that is used to pass username, responsibility, responsibility application, and NLS language elements for Web service authorization:

```
String headerPartMsg = "<SOAHeader
xmlns=\"http://xmlns.oracle.com/xdb/SYSTEM\" " +
            "env:mustUnderstand=\"0\"
xmlns:env=\"http://schemas.xmlsoap.org/soap/envelope/\"> \n" +
         " <MESSAGE_TYPE>XML</MESSAGE_TYPE>\n" +
         " <MESSAGE_STANDARD>OAG</MESSAGE_STANDARD>\n" +
         " <TRANSACTION_TYPE>PO</TRANSACTION_TYPE>\n" +
         " <TRANSACTION_SUBTYPE>PROCESS</TRANSACTION_SUBTYPE>\n" +
         " <DOCUMENT_NUMBER>123</DOCUMENT_NUMBER>\n" +
         " <PARTY_SITE_ID>4444</PARTY_SITE_ID>\n" +
         "</SOAHeader>\n";
businessEvent.setStringProperty("WFBES_INPUT_header",
headerPartMsg);
```

> **Note:** Please note that this WFBES_INPUT_`<partname>` parameter can only be passed at run time during the event raise, not through the event subscription. Several constants are defined in interface `oracle.apps.fnd.wf.bes.InvokerConstants` for use in Java code.

If the Web service input message definition has several parts, value for the part that is sent as SOAP body is passed as event payload. Values for all other parts are passed as event parameters with parameter name format WFBES_INPUT_ `<partname>`. If the value for a specific input message part is optional to invoke the Web service, you still have to pass the parameter with null value so that invoker subscription knows to which part the event payload should be set as SOAP body. For example, pass the following parameter with null value:

```
businessEvent.setStringProperty("WFBES_INPUT_myheader", null);
```

- **Create an error subscription to enable error processing**

  To enable error processing in the Business Event System that communicates with SYSADMIN user about error conditions during subscription execution, you must subscribe to the event with 'Launch Workflow' action type for error processing.

- **Create a receive event (optional)**

  If a Web service has an output or a response message to communicate or callback to Oracle E-Business Suite, and the invoker event is raised from Java code with the subscription phase >= 100 or if the event is raised from PL/SQL, then you should create a receive event for callback feature to complete the invocation process. Additionally, create an external subscription to the receive event to pass the Web service response.

  > **Note:** If it is raised from Java with subscription phase < 100, then the Web service is invoked immediately and response is available to the calling program using `BusinessEvent.getResponseData()` method after calling `BusinessEvent.raise()`. In this case, the response may not

have to be communicated back to Oracle E-Business Suite using callback event.

If a Web service does not require a response, then there is no need to create a receive event.

- **Create a receive event subscription (optional)**

  If you have a receive event created, you must also create an external event subscription to pass the Web service response.

  Please note that the subscription to the receive event does not have to be with "Launch Workflow" action type. It can be created with any action type that the system integration developer wants.

To create an event, log on to Oracle Workflow with the Workflow Administrator Web Applications responsibility and select the Business Event link and click **Create**.

To access the business event subscription page, log on to Oracle Workflow with the same Workflow Administrator Web Applications responsibility and select the Business Event link > Subscriptions. Click **Create Subscription** to access the event subscription page.

For detailed instructions on how to create business events and event subscriptions to invoke Web services, see the *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

Calling Back to Oracle E-Business Suite With Web Service Response

As mentioned earlier, if a Web service has an output or a response message to communicate or callback to Oracle E-Business Suite, then a receive event and the local subscription to the receive event must be created first in the Business Event System.

To accomplish this synchronous request - response process, the Service Invocation Framework uses the *callback* mechanism to communicate the response back to Oracle E-Business Suite through the Business Event System. As a result, a new or waiting workflow process can be started or executed. The following callback subscription parameters are used to support the callback mechanism:

- WFBES_CALLBACK_EVENT

  This subscription parameter can have a valid business event to be raised upon completion of the Web service with the service output message as payload.

  For example, it can be like:

  ```
  WFBES_CALLBACK_EVENT=oracle.apps.wf.myservice.callback
  ```

- WFBES_CALLABACK_AGENT

  This subscription parameter can have a valid business event system agent to which the event with the service response message as payload can be enqueued.

For example, it can be like:

```
WFBES_CALLBACK_AGENT=WF_WS_JMS_IN
```

> **Note:** `WF_WS_JMS_IN` is a standard default inbound agent for Web service messages. If desired, a custom agent can also be created to enqueue Web service responses. Additionally, if an agent listener is not available, you need to create one. See the *Oracle Workflow Developer's Guide* for details.

If event parameters are passed with the same names as the subscription parameters that have been parsed and stored, the event parameter values take precedence over subscription parameters. For example, the event parameters are passed as follows:

- `BusinessEvent.setStringProperty("WFBES_CALLBACK_EVENT", "oracle.apps.wf.myservice.callback");`
- `BusinessEvent.setStringProperty("WFBES_CALLBACK_AGENT", "WF_WS_JMS_IN");`

To process Web service responses from inbound workflow agent, make sure that you have agent listener set up properly.

Detailed information about these callback subscription parameters, see the *Oracle E-Busines Suite Integrated SOA Gateway Developer's Guide*.

Managing Errors

If there is a runtime exception when invoking the Web service by raising the Invoker event with synchronous subscription (phase <100), the exception thrown to the calling application. It is the responsibility of the calling application to manage the exception.

If there is a runtime exception when the Workflow Java Deferred Agent Listener executes event subscription to invoke the Web service, the event is enqueued to the WF_JAVA_ERROR queue. If the event has an Error subscription defined to launch Error workflow process `WFERROR:DEFAULT_EVENT_ERROR2`, the Workflow Java Error Agent Listener executes the error subscription which sends a notification to a user (`SYSADMIN`) with Web service definition, error details and event details. The `SYSADMIN` user can correct the error and then invoke the Web service again from the notification if necessary.

For more information on error handling during Web service invocation, see the *Oracle E-Busines Suite Integrated SOA Gateway Developer's Guide*.

Testing Web Service Invocations

To validate whether Web services can be successfully invoked from concurrent manager and OACORE WebLogic Server, system integration developers can run a test case through Oracle Workflow Test Business Event page. Use this test to check the basic operation of Business Event System by raising a test event from Java or from PL/SQL and executing synchronous and asynchronous subscriptions to that event.

By using 'Raise in Java' option to raise the Invoker event with synchronous subscription (phase <100), Web service invocation within OACORE WebLogic Server can be tested. If there is a runtime exception when invoking the Web service using synchronous subscription, the exception message is shown on the Test Business Event page.

The following event parameters may be specified when raising the event from the Test Business Event page to invoke a Web service:

- Message transformation: XSL transformation for Web service input message and output message
    - WFBES_OUT_XSL_FILENAME
    - WFBES_IN_XSL_FILENAME
- Input Message part value: Pass values for any part that may be required to embed applications context into SOAP envelopes
    - WFBES_INPUT_`<partname>`
- WS-Security: Information required to add UsernameToken header to a SOAP request as event parameters

    The Web service security information is entered in the Web Service Security region of the event subscription page after the Invoke WSDL wizard. See: Create a local subscription to invoke a Web service, page 9-13.

    > **Note:** As described here that security information is now entered through the event subscription user interface to replace the security parameters used in Oracle E-Business Suite Release 12.1.
    >
    > These WS-Security parameters (WFBES_SOAP_USERNAME, WFBES_SOAP_PASSWORD_MOD, and WFBES_SOAP_PASSWORD_KEY) are now maintained internally by service invocation framework.
- Callback: Callback to Oracle E-Business Suite with Web service responses
    - WFBES_CALLBACK_EVENT
    - WFBES_CALLBACK_AGENT
- SOAP Body:
    - XML Input message (Required)

Detailed information on how to test Web service invocations, see the *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

Extending Web Service Invocation

Oracle E-Business Suite Integrated SOA Gateway allows developers to extend the invoker subscription seeded rule function `oracle.apps.fnd.wf.bes.WebServiceInvokerSubscription` using Java coding standards for more specialized processing.

Developers could extend the seeded rule function to override following methods for custom processing:

- preInvokeService
- postInvokeService
- addWSSecurityHeader
- setInputParts
- addCustomSOAPHeaders

For more information on these methods, see the *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

Implementation Limitation and Consideration

While implementing the Service Invocation Framework, consider the following limitations:

- WFBES_INPUT_`<partname>` Parameter Can Only be Passed at the Event Raise

  The Service Invocation Framework uses event parameter WFBES_INPUT_ `<partname>` to support passing values for any header part that may be required to embed applications context into SOAP envelopes. However, unlike other parameters that can be defined while subscribing to the Invoker event, this event parameter can only be defined during the event raise.

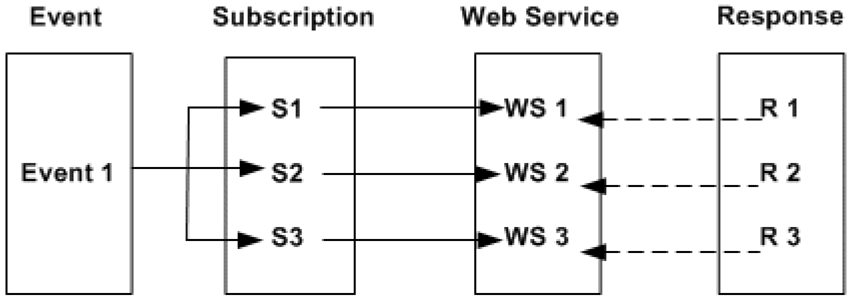
- Support Document Style Web Services Only

  The Service Invocation Framework supports invoking only document-based Web services. The RPC (remote procedure call) style remote Web service invocation is not supported in this release.

- Support One-to-One Relationship of Event Subscriptions

  To successfully invoke Web services, each event should only have one subscription (with 'Invoker Web Service' action type) associated with it. This one-to-one relationship of event subscription is especially important in regards to synchronous request - response service invocation.

  For example, if there are three event subscriptions (S1, S2, and S3) for the same event (Event 1), when a triggering event occurs at run time, the services associated with each subscription can be invoked three times (WS1, WS2, and WS3) respectively. The scenario is illustrated in the following diagram:



- If callback parameters are not passed, `getResponseData()` method on the `BusinessEvent` object returns the output (response) message in the same session after the Invoker event raise. The R2 overrides the R1; R3 overrides the R2. As a result, you will only get R3 message back.

- If callback parameters are passed, since there are three different instances of the receive event with the same event key, it is difficult to match the response to the corresponding Invoke Web Service subscription.

A

Oracle E-Business Suite Integrated SOA Gateway Diagnostic Tests

Overview

The installation of Oracle E-Business Suite Integrated SOA Gateway requires number of manual setup tasks on both Oracle E-Business Suite and Oracle SOA Suite. To effectively identify any issues, Oracle E-Business Suite Integrated SOA Gateway provides a suite of diagnostic tests executed through a backend script to help determine specific causes or issues with installation steps.

This diagnostic test suite includes multiple tests with various test functions to check on both Oracle E-Business Suite and Oracle SOA Suite instances. For example, certain tests validate if correct versions of required software and libraries are installed, some tests check if needed patches are applied, or if issues are functional related, such as Generate, Deploy, or other design-time activities.

At the end of test run, a report will be generated which may contain corrective actions mostly regarding the installation of Oracle E-Business Suite Integrated SOA Gateway.

To have better understanding on the diagnostic tests and how the test suite is executed, the following topics are included in this chapter:

- Understanding the Usage of Backend Script $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml, page A-1
- Running Diagnostic Tests, page A-3

Understanding the Usage of Backend Script $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml

Oracle E-Business Suite Integrated SOA Gateway uses an Ant script `$JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml` to run the diagnostic tests through backend processing.

Note: Please note that `$JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml` is a multipurpose script. It can also be used to execute the design-time activities such as generate, regenerate, deploy, undeploy, activate, retire, and reset services as well as to upgrade or postclone services from command line, or download the configuration file from the instance.

For more information on how to use the script to perform design-time activities or download the configuration file, see Managing Web Service Life Cycle Activities Using An Ant Script, page 3-27.

Usage

1. Enter `ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml`. It will prompt for the arguments.

   Note: Do not enclose any input between double quotes.

2. Enter the arguments in the following ways:

   -
     ```
     ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
     -Dactions=<comma separated list of operations>
     -DserviceType=SOAP -DirepNames=<comma separated list of
     API Names> -Dverbose=<ON|OFF>
     ```

     While passing actions and `irepNames` using this method, be aware of the following conditions:

     - If more than one action or `irepNames` is passed as command line argument, enclose them between double quotes. For example,

       `-Dactions="method1, method2,.."`

       `-DirepNames="ECRDTLD,FND_USER_PKG[func1:SY;func2:AS;..],..."`

     - If only one action or `irepNames` is passed as command line argument, then there is no need to enclose between double quotes.

   -
     ```
     ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
     -Dfile=<absolute path of service descriptor file>
     -Dverbose=<ON|OFF>
     ```

Information on argument descriptions for the script `isgDesigner.xml`, see Managing Web Service Life Cycle Activities Using An Ant Script, page 3-27.

Running Diagnostic Tests

Use the backend script `isgDesigner.xml` to run complete diagnostic tests on both Oracle E-Business Suite and Oracle SOA Suite with the following syntax:

```
ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
DiagnoseISGSetup
```

Alternatively, you can use the same script supplying with different targets to run the configuration checks for various purposes. For example, use the test to check only on the Oracle E-Business Suite side or the Oracle SOA Suite side, or to test the design-time operations of Oracle E-Business Suite Integrated SOA Gateway for all types of interfaces.

Use the following commands to run diagnostic tests depending on your needs:

-
  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
  DiagnoseAGENTSetup
  ```

  This command runs configuration checks on the Oracle E-Business Suite side.

-
  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
  DiagnoseAPPSetup
  ```

  This command runs configuration checks on the Oracle SOA Suite side.

-
  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
  DiagnoseISGSetup
  ```

  This command runs complete diagnostic tests on both Oracle E-Business Suite and Oracle SOA Suite.

-
  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/ant/isgDesigner.xml
  DiagnoseISGFunctionality
  ```

  This command runs all design-time operations for all types of interfaces in Oracle E-Business Suite Integrated SOA Gateway.

After each test run, a report DiagnosticsReport.xml will be generated as a result. The generated report will have test name, status, and message if test is failed. Message will convey the information that what type of error occurred and what is the error and corresponding actions if available.

B

Synchronous and Asynchronous Web Services

Synchronous and Asynchronous Web Services

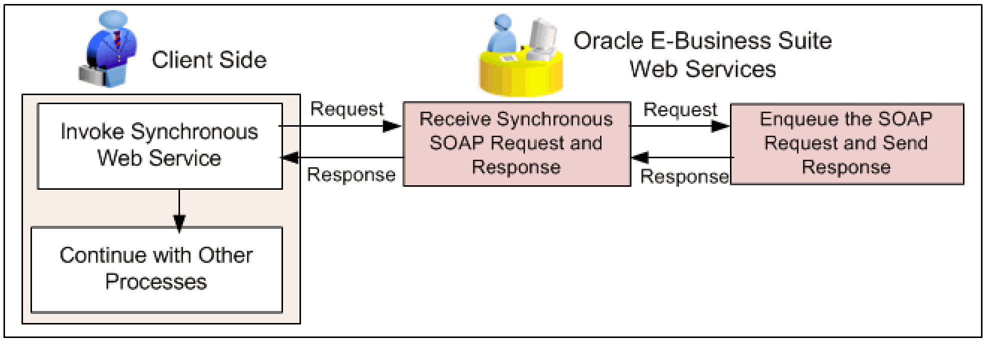
Oracle E-Business Suite Integrated SOA Gateway supports both synchronous and asynchronous service processing and execution for SOAP-based services.

Interfaces exposed as REST services can be generated with the support for synchronous interaction pattern only. Asynchronous pattern for REST services is not supported in this release.

- **Synchronous Web Services**

  This type of service execution provides an immediate response to a query. In this situation, the client will wait until the server sends back the response message. The advantage of using the synchronous service is that the client application knows the status of the Web service operation in a very short time.

  Synchronous Web Service Transaction

  
  Client Side
Oracle E-Business Suite Web Services
Invoke Synchronous Web Service
Request
Response
Receive Synchronous SOAP Request and Response
Request
Response
Enqueue the SOAP Request and Send Response
Continue with Other Processes

  When a Web service client sends a synchronous SOAP request to Oracle E-Business Suite service, the SOAP response will be sent back to the client as soon as the process completes.

- **Asynchronous Web Services (SOAP Web Services Only)**

  This type of service execution may require a significant amount of time to process a request. However, the client that invoked the Oracle E-Business Suite Web service can continue with other processing in the meantime rather than wait for the response.

  Asynchronous operation is extremely useful for environments in which a service, such as a loan processor, can take a long time to process a client request.

  In this release, asynchronous operation pattern is supported for SOAP-based Web services only.

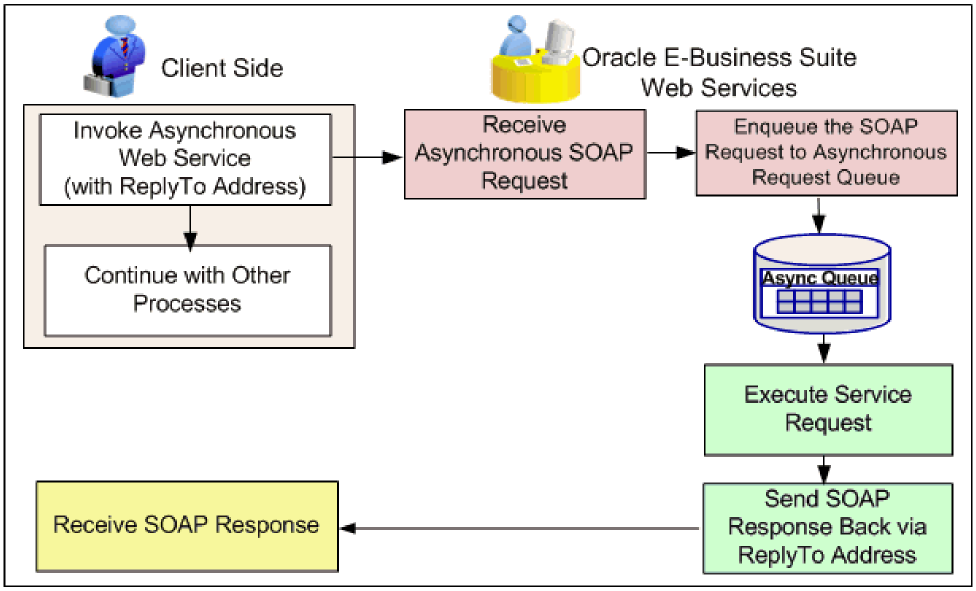
  Callback without Acknowledgement

  Callback pattern is a very important communication method in asynchronous services - a request is made to the service provider and a response is sent back to the requester when it is ready. This pattern can be used in conjunction with acknowledgement to recognize the receipt of a request sent by a requester. Only *callback without acknowledgement* pattern is supported in this release.

  In callback without acknowledgment pattern, a SOAP Callback header becomes necessary when the Web service is asynchronous and the client contact information is unknown at deployment time. Callback header allows the client to specify how to contact the client (`ReplyTo` address) in the request for service. Therefore, client must publish a listener or a receive service. In other words, the structure of the WSDL dictates how the client will receive the response.

  A Web service client must provide `MessageID`, and an appropriate callback endpoint address (`ReplyTo` and `FaultTo`) using WS-Addressing in SOA Headers for the asynchronous request callback pattern.

Asynchronous Web Service Transaction


Client Side
Oracle E-Business Suite Web Services
Invoke Asynchronous Web Service (with ReplyTo Address)
Continue with Other Processes
Receive Asynchronous SOAP Request
Enqueue the SOAP Request to Asynchronous Request Queue
Async Queue
Execute Service Request
Send SOAP Response Back via ReplyTo Address
Receive SOAP Response

> When a Web service client sends a SOAP request to Oracle E-Business Suite service, on completion of service execution, the SOAP response (service response payload) is sent to ReplyTo address of the client. This pattern does not expect acknowledgment from client as it is a fire-and-forget message exchange pattern for callback.

SOAP services, depending on specified interaction patterns, can be generated synchronously, asynchronously, or both synchronously and asynchronously to meet your business needs. REST services can be generated with synchronous operation only.

Once a SOAP service has been generated and deployed to an Oracle SOA Suite WebLogic managed server, service consumers or Web service clients can send request messages through Oracle SOA Suite. After security checks on the inbound requests, Oracle E-Business Suite Web services can be invoked synchronously or asynchronously.

For information on how to specify interaction patterns for a given interface, see Generating SOAP Web Services, page 3-4.

C

Error Messages

Error Messages and Solutions

The following table describes error message if occurs during the design-time activities through the Integration Repository user interface, as well as during service runtime execution from service provider.

The error codes and corresponding solutions are also listed in the table for possible solutions.

Error Type	HTTP Status Code	Code	Error Message	Resolution
Server Side	500	ISG-011, ISG-012, ISG-013, ISG-014, and ISG-015	Error in Service Generation	Contact Oracle Support to get fix for this issue.
Server Side	500	ISG-016	Error in creating database connection	Make sure that Oracle E-Business Suite database is running and verify the setup tasks.
Client Side	401	ISG-051	Username is not valid	Error in Service SCA Composite. Contact Oracle Support to get fix for this issue.

Error Type	HTTP Status Code	Code	Error Message	Resolution
Client Side	403	ISG-052	Responsibility Key does not exist in Oracle E-Business Suite	Verify that Responsibility Key is passed in the SOAHeader.
Client Side	403	ISG-053	Responsibility application short code id does not exist in Oracle E-Business Suite	Verify that Application Short Code is passed in SOAHeader.
Client Side	403	ISG-054	Language code does not exist in Oracle E-Business Suite	Verify that NLS Language Code is passed in SOAHeader.
Client Side	403	ISG-055	Security group key does not exist in Oracle E-Business Suite	Verify that Security Group Key is passed in SOAHeader.
Client Side	403	ISG-056	Org Id does not exist in Oracle E-Business Suite	Verify that Org Id is passed in SOAHeader.
Server Side	500	ISG-057	Exception occurred while setting up Applications Context	Verify all the elements in SOAHeader.
Server Side	500	ISG-059	Language code is not valid	Verify if NLS Language passed in SOAHeader is installed on server.

Error Type	HTTP Status Code	Code	Error Message	Resolution
Client Side	403	ISG-060	Responsibility is not assigned to user	Ensure that Responsibility passed in SOAHeader is assigned to Username in wsse security header.
Server Side	500	ISG-062	Exception occurred while setting up Application Context	Contact Oracle Support to get fix for this issue.
Server Side	500	ISG-070	Invalid Internal Name	Contact Oracle Support to get fix for this issue.
Client Side	403	ISG-100	XML Gateway Header Properties not found	Verify that XML Gateway Header Properties are present in the SOAP Request.
Client Side	403	ISG-101	XML Gateway Transaction Type is missing	Verify that Transaction Type is present in ECXMSG of SOAHeader.
Client Side	403	ISG-102	XML Gateway Transaction Subtype is missing	Verify that Transaction SubType is present in ECXMSG of SOAHeader.
Client Side	403	ISG-103	XML Gateway Party Site ID is missing	Verify that Party Site ID is present in ECXMSG of SOAHeader.

Error Type	HTTP Status Code	Code	Error Message	Resolution
Server Side	500	ISG-201	Unknown error during service execution	Unknown error during service execution. View runtime logs for more information.
Server Side	500	ISG-203	Unable to commit service transaction	View runtime logs for more information.
Server Side	500	ISG-205	Unable to rollback service transaction	View runtime logs for more information.
Server Side	500	ISG-231	Error occurred while parsing Business Event Data	View runtime logs for more information.
Server Side	500	ISG-251	Database Adapter is not found	Contact Oracle Support to get fix for this issue.
Server Side	500	ISG-252	AQ Adapter is not found	Contact Oracle Support to get fix for this issue.
Client Side	403	ISG-301, ISG-303, and ISG-305	User is not authorized to execute service	Contact your System Administrator and verify that grant to execute this service is present.
Server Side	500	ISG-499	Unknown error during service execution	View runtime logs for more information.

Glossary

Agent

A named point of communication within a system.

Agent Listener

A type of service component that processes event messages on inbound agents.

Asynchronous Operation

Unlike the synchronous service execution to obtain the result immediately, asynchronous operations may require a significant amount of time to process a request.

However, the client that invoked the Oracle E-Business Suite Web service can continue with other processing in the meantime rather than wait for the response.

BPEL

Business Process Execution Language (BPEL) provides a language for the specification of executable and abstract business processes. By doing so, it extends the services interaction model and enables it to support business transactions. BPEL defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to-business spaces.

Business Event

See Event.

Callback Pattern

Callback pattern is an important communication method in asynchronous services. An asynchronous callback means that a request is made to the service provider and a response (callback) is sent back to the requester when it is ready. This pattern can be used in conjunction with acknowledgement to recognize the receipt of a request sent by a requester.

Concurrent Manager

An Oracle E-Business Suite component that manages the queuing of requests and the operation of concurrent programs.

Concurrent Program

A concurrent program is an executable file that performs a specific task, such as posting a journal entry or generating a report.

Event

An occurrence in an internet or intranet application or program that might be significant to other objects in a system or to external agents.

Event Activity

A business event modelled as an activity so that it can be included in a workflow process.

Event Data

A set of additional details describing an event. The event data can be structured as an XML document. Together, the event name, event key, and event data fully communicate what occurred in the event.

Event Key

A string that uniquely identifies an instance of an event. Together, the event name, event key, and event data fully communicate what occurred in the event.

Event Message

A standard Workflow structure for communicating business events, defined by the datatype `WF_EVENT_T`. The event message contains the event data as well as several header properties, including the event name, event key, addressing attributes, and error information.

Event Subscription

A registration indicating that a particular event is significant to a system and specifying the processing to perform when the triggering event occurs. Subscription processing can include calling custom code, sending the event message to a workflow process, or sending the event message to an agent.

Function

A PL/SQL stored procedure that can define business rules, perform automated tasks within an application, or retrieve application information. The stored procedure accepts standard arguments and returns a completion result.

Integration Repository

Oracle Integration Repository is the key component or user interface for Oracle E-Business Suite Integrated SOA Gateway. This centralized repository stores native packaged integration interface definitions and composite services.

Integration Repository Parser

It is a standalone design-time tool used by the integration repository administrator to validate annotated custom interface definitions against the annotation standards and generate an Integration Repository loader file (iLDT). This generated iLDT file can be uploaded to Integration Repository where custom interfaces can be exposed to all users.

Interface Type

Integration interfaces are grouped into different interface types.

JSON

JSON (JavaScript Object Notation) is a text-based open standard designed for human-readable data interchange. The JSON format is often used with REST services to transmit structured data between a server and Web application, serving as an alternative to XML.

Loose Coupling

Loose coupling describes a resilient relationship between two or more systems or organizations with some kind of exchange relationship. Each end of the transaction makes its requirements explicit and makes few assumptions about the other end.

Lookup Code

An internal name of a value defined in a lookup type.

Lookup Type

A predefined list of values. Each value in a lookup type has an internal and a display name.

Message

The information that is sent by a notification activity. A message must be defined before it can be associated with a notification activity. A message contains a subject, a priority, a body, and possibly one or more message attributes.

Message Attribute

A variable that you define for a particular message to either provide information or prompt for a response when the message is sent in a notification. You can use a predefine item type attribute as a message attribute. Defined as a 'Send' source, a message attribute gets replaced with a runtime value when the message is sent. Defined as a 'Respond' source, a message attribute prompts a user for a response when the message is sent.

Notification

An instance of a message delivered to a user.

Notification Worklist

A Web page that you can access to query and respond to workflow notifications.

Operation

An abstract description of an action supported by a service.

Port

A port defines an individual endpoint by specifying a single address for a binding.

Port Type

A port type is a named set of abstract operations and abstract messages involved.

Process

A set of activities that need to be performed to accomplish a business goal.

REST

Representational State Transfer (REST) is an architecture principle in which the Web services are viewed as resources and can be uniquely identified by their URLs. The key characteristic of a REST service is the explicit use of HTTP methods (GET, POST, PUT, and DELETE) to denote the invocation of different operations.

Please note that only POST method is supported in this release.

SAML Token (Sender-Vouches)

This type of security model authenticates Web services relying on sending a username only through Security Assertion Markup Language (SAML) assertion.

SAML is an XML-based standard for exchanging authentication and authorization data between security domains, that is, between an identity provider and a service provider. SAML Token uses a sender-vouches method to establish the correspondence between a SOAP message and the SAML assertions added to the SOAP message.

See Username Token.

Service

A service is a collection of related endpoints.

Service Component

An instance of a Java program which has been defined according to the Generic Service Component Framework standards so that it can be managed through this framework.

Service Monitor

It is the monitoring and auditing tool in Oracle E-Business Suite allowing you to view runtime messages for web services provided by Oracle E-Business Suite Integrated SOA

Gateway.

It is known as SOA Monitor in earlier releases.

SOA

Service-oriented Architecture (SOA) is an architecture to achieve loose coupling among interacting software components and enable seamless and standards-based integration in a heterogeneous IT ecosystem.

SOA Composite (SCA Composite)

It is a new set of specifications that define a new way of assembling SOA-enabled applications. It is developed and deployed as a single service that includes all the components it assembles to form the application implementation. In Oracle SOA Suite 11*g*, it may contain one or more cooperating component types such as Mediator component, BPEL process component, and so on.

SOAP

Simple Object Access Protocol (SOAP) is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols.

Subscription

See Event Subscription.

Synchronous Operation

Synchronous operation provides an immediate response to a query. In this situation, the client connection remains open from the time the request is submitted to the server. The client will wait until the server sends back the response message.

Username Token

A type of security model based on username and password to authenticate SOAP requests at run time.

See SAML Token (Sender-Vouches).

WADL

Web Application Description Language (WADL) is designed to provide a machine-processable description of HTTP-based Web applications. It models the resources provided by a service and the relationships between them.

Web Services

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in WSDL. Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Workflow Engine

The Oracle Workflow component that implements a workflow process definition. The Workflow Engine manages the state of all activities for an item, automatically executes functions and sends notifications, maintains a history of completed activities, and detects error conditions and starts error processes. The Workflow Engine is implemented in server PL/SQL and activated when a call to an engine API is made.

WSDL

Web Services Description Language (WSDL) is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint.

WS-Addressing

WS-Addressing is a way of describing the address of the recipient (and sender) of a message, inside the SOAP message itself.

WS-Security

WS-Security defines how to use XML Signature in SOAP to secure message exchanges, as an alternative or extension to using HTTPS to secure the channel.

Index