Oracle® E-Business Suite

Integrated SOA Gateway Implementation Guide

Release 12.1

Part No. E12169-13

February 2022

Oracle E-Business Suite Integrated SOA Gateway Implementation Guide, Release 12.1

Part No. E12169-13

Primary Author: Melody Yang

Contributor: Rekha Ayothi, Sudipto Chakraborty, Neeraj Chauhan, Bhaskar Ghosh, Vardhan Kale, Anil Kemisetti, Surya Narayana Nellepalli, Aditya Rao, Nadakuditi Ravindra, Dilbaghsingh Sardar, Vijayakumar Shanmugam, Shivdas Tomar, Abhishek Verma

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.

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

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

5 Administering Custom Integration Interfaces and Services

6 Securing Web Services

7 Logging for Web Services

8 Monitoring and Managing SOAP Messages Using SOA Monitor

9 Implementing Service Invocation Framework

A Oracle E-Business Suite Integrated SOA Gateway Diagnostic Tests

Glossary

Index

Send Us Your Comments

Oracle E-Business Suite Integrated SOA Gateway Implementation Guide, Release 12.1

Part No. E12169-13

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

Structure

Related Information Sources

This book is included in the Oracle E-Business Suite Documentation Library. If this guide refers you to other Oracle E-Business Suite documentation, use only the latest Release 12.1 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.

- **Oracle E-Business Suite Documentation Library** - This library, which is included in the Oracle E-Business Suite software distribution, provides PDF documentation as of the time of each release.

- **Oracle E-Business Suite Documentation Web Library** - This library, available on the Oracle Help Center (https://docs.oracle.com/cd/E18727_01/index.htm), provides the latest updates to Oracle E-Business Suite Release 12.1 documentation. Most documents are available in PDF and HTML formats.

- **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 from 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 Cloud Using the Oracle E-Business Suite Adapter with Oracle Integration Cloud

This guide describes how to set up and use Oracle E-Business Suite Adapter connections in Oracle Integration to access supported Oracle E-Business Suite interfaces and REST services as inbound or outbound integrations from Oracle E-Business Suite.

As part of the integration documents in Oracle Cloud Platform as a Service (PaaS), this guide is available in Oracle Cloud Documentation.

Oracle E-Business Suite Concepts

This book is intended for all those planning to deploy Oracle E-Business Suite Release 12, 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 E-Business Suite CRM System Administrator's Guide

This manual describes how to implement the CRM Technology Foundation (JTT) and use its System Administrator Console.

Oracle E-Business Suite Developer's Guide

This guide contains the coding standards followed by the Oracle E-Business Suite development staff. It describes the Oracle Application Object Library components needed to implement the Oracle E-Business Suite user interface described in the *Oracle E-Business Suite User Interface Standards for Forms-Based Products*. It provides information to help you build your custom Oracle Forms Developer forms so that they integrate with Oracle E-Business Suite. In addition, this guide has information for customizations in features such as concurrent programs, flexfields, messages, and logging.

Oracle Application Framework Developer's Guide

This guide contains the coding standards followed by the Oracle E-Business Suite development staff to produce applications built with Oracle Application Framework. This guide is available in PDF format on My Oracle Support and as online documentation in Oracle JDeveloper 10g with Oracle Application Extension.

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, or as part of an upgrade from Release 11i to Release 12. The book also describes the steps needed to install the technology stack components only, for the special situations where this is applicable.

Oracle Fusion Middleware Adapter for Oracle Applications User's Guide

This book covers the use of Oracle E-Business Suite Adapter (formerly known as "Adapter for Oracle Applications" in Oracle Applications Server 10g or Oracle Fusion Middleware 11g releases) in developing integrations between Oracle E-Business Suite and trading partners.

This book is available in the Oracle Application Server 10g Documentation Library and Oracle Fusion Middleware 11g Documentation Library.

Oracle E-Business Suite System Administrator's Guide Documentation Set

This documentation set provides planning and reference information for the Oracle E-Business Suite System Administrator. *Oracle E-Business Suite System Administrator's Guide - Configuration* contains information on system configuration steps, including defining concurrent programs and managers, enabling Oracle Applications Manager features, and setting up printers and online help. *Oracle E-Business Suite System Administrator's Guide - Maintenance* provides information for frequent tasks such as monitoring your system with Oracle Applications Manager, administering Oracle E-Business Suite Secure Enterprise Search, managing concurrent managers and reports, using diagnostic utilities including logging, managing profile options, and using alerts. *Oracle E-Business Suite System Administrator's Guide - Security* describes User Management, data security, function security, auditing, and security configurations.

Oracle E-Business Suite User's Guide

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

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 Mobile Apps Administrator's Guide, Release 12.1 and 12.2

This guide describes how to set up an Oracle E-Business Suite instance to support connections from Oracle E-Business Suite mobile apps. It also describes common administrative tasks for configuring Oracle E-Business Suite mobile apps and setup tasks for enabling push notifications for supported mobile apps. Logging and troubleshooting information is also included in this book.

Oracle E-Business Suite Mobile Apps Developer's Guide, Release 12.1 and 12.2

This guide describes how to develop enterprise-distributed mobile apps by using mobile application archive (MAA) files and how to implement corporate branding. It also explains required tasks on implementing push notifications for supported mobile apps. In addition, it includes how to implement Oracle E-Business Suite REST services to develop custom mobile apps by using the Login component from Oracle E-Business Suite Mobile Foundation or using any mobile app development framework if desired.

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 runtime.

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 User's Guide

This guide describes the functionality of Oracle e-Commerce Gateway and the necessary setup steps in order for Oracle E-Business Suite to conduct business with trading partners through Electronic Data Interchange (EDI). It also contains how to run extract programs for outbound transactions, import programs for inbound transactions, and the relevant reports.

Oracle e-Commerce Gateway Implementation Manual

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 Developer's Guide

This manual describes how to build, test, and deploy Oracle iSetup Framework interfaces.

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 Web Applications Desktop Integrator Implementation and Administration Guide

Oracle Web Applications Desktop Integrator brings Oracle E-Business Suite functionality to a spreadsheet, where familiar data entry and modeling techniques can be used to complete Oracle E-Business Suite tasks. You can create formatted spreadsheets on your desktop that allow you to download, view, edit, and create Oracle

E-Business Suite data, which you can then upload. This guide describes how to implement Oracle Web Applications Desktop Integrator and how to define mappings, layouts, style sheets, and other setup options.

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 Developer's Guide

This guide explains how to define new workflow business processes and customize existing Oracle E-Business Suite-embedded workflow processes. It also describes how to define and customize business events and event subscriptions.

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 Workflow Client Installation Guide

This guide describes how to install the Oracle Workflow Builder and Oracle XML Gateway Message Designer client components for Oracle E-Business Suite.

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 Oracle 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.

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

This chapter covers the following topics:

- Oracle E-Business Suite Integrated SOA Gateway Overview
- Native Service Enablement Architecture Overview

Oracle E-Business Suite Integrated SOA Gateway Overview

Building on top of Oracle Fusion Middleware and service-oriented architecture (SOA) technology, Oracle E-Business Suite Integrated SOA Gateway (ISG) is a complete set of service infrastructure to provide, consume, and administer Oracle E-Business Suite Web services.

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

SOAP-based services are described in WSDLs and are deployed to the application server for service consumption. REST services described in WADLs are used for user-driven applications such as Oracle E-Business Suite mobile applications.

Oracle E-Business Suite Integrated SOA Gateway provides Service Invocation Framework to invoke and consume Web services provided by other applications.

For more information about each integration interface and service, see the *Oracle E-Business Suite Integrated SOA Gateway User's Guide*; for more information about Web service invocation and performing service integration activities, see the *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

Major Features

Oracle E-Business Suite Integrated SOA Gateway contains the following features:

- 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 interaction pattern for both SOAP-based and REST-based web services

  > **Note:** In this release, only PL/SQL APIs, Concurrent Programs, and Business Service Objects can be exposed as both SOAP and REST services. Java Bean Services, Application Module Services, Open Interface Tables, and Open Interface Views can be exposed as REST services only.

- 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 user interface for logging configuration
- Audit and monitor Oracle E-Business Suite service operations from native SOA Monitor
- Leverage Oracle Workflow Business Event System to enable web service invocation from Oracle E-Business Suite

Major Components Features and Definitions

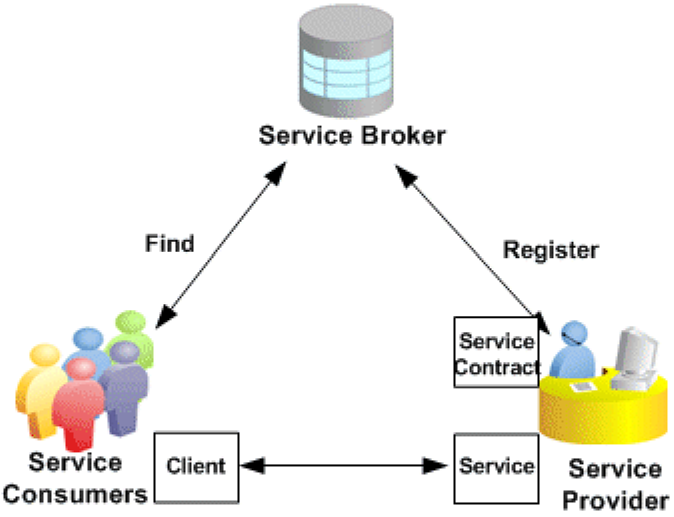
The better understand Oracle E-Business Suite Integrated SOA Gateway and its key components, this section describes some key features and the definition of each component.

Native Service Enablement

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 residing in Oracle Integration Repository to be further transformed into Web services that comply with Web standards. Additionally, these services can be deployed from the Integration Repository to the application server allowing more consumptions over the Web.

To understand the basic concept of Web services and how the service works, the following diagram illustrates the essential components for service enablement:

Major Components for Service Enablement



A Service Provider is the primary engine underlying the Web services. It facilitates the service enablement for various types of interfaces.

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

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

Composite Services

Composite services use the native service as building blocks to construct the sequence of business flows. Basically, this interface type orchestrates the invocation sequence of discrete Web services into a meaningful end-to-end business process through a Web service composition language BPEL (business process execution language).

For example, use Oracle BPEL Process Manager (BPEL PM) to integrate the Order-to-Receipt business process that contains sales order entry, item availability check, pack and ship, and invoice to Accounts Receivable sub processes handled by various applications. This approach effectively tightens up the control of each individual process and makes the entire business flow more efficiently.

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.

To effectively manage all integration interfaces and services incurred within the Oracle E-Business Suite, Oracle E-Business Suite Integrated SOA Gateway now supports complex business processes or composite services, Web service generation and deployment, as well as business event subscriptions through the centralized Integration Repository.

You can browse these interface definitions and services through the Oracle Integration Repository user interfaces. Users with administrator privileges can further perform administrative tasks through the same interfaces.

Oracle Integration Repository supports the following interface types:

- PL/SQL
- XML Gateway
- Concurrent Programs
- Business Events
- Open Interface Tables and Open Interface Views
- EDI
- Business Service Object (Service Beans)
- Java

  Apart from normal Java APIs, Java interface includes the following subcategories:

  - Application Module Services

    > **Note:** Application Module Implementation class is a Java class that provides access to business logic governing the OA Framework-based components and pages. Such Java classes are called Application Module Services and are categorized as a subtype of Java interface.

  - Java Bean Services

    > **Note:** Java APIs whose methods use parameters of either simple data types or serializable Java Beans are categorized as Java Bean Services. Such Java APIs can be exposed as REST-based Web services.

  - Security Services

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

Please note that Java APIs for Forms (Forms-based Web services) are desupported in Oracle E-Business Suite Release 12.2. If you are planning to use this type of interfaces as Web services, you are advised to use alternate serviceable interfaces, such as PL/SQL and Business Service Objects interfaces, which can be deployed as Web services. Refer to My Oracle Support Knowledge Document 966982.1 for the suggested alternatives to the existing Java APIs for Forms services.

- Composite Services

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, Concurrent Program, Business Service Object, Application Module Services, Java Bean Services, Open Interface Tables, and Open Interface Views.

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.

Manage 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.

SOA Monitor

SOA Monitor is a centralized, light-weight service execution monitoring and management tool. It not only monitors all the SOAP requests that SOA Provider and Web Service Provider process, but also provides auditing feature for the SOAP messages if the auditing feature is enabled.

With SOA Monitor, the integration repository administrator can effectively manage and identify errors incurred during the service deployment life cycle and take necessary actions to expedite the interaction between services.

Service Invocation Framework

To invoke all integration services from Oracle E-Business Suite, Oracle E-Business Suite Integrated SOA Gateway uses the Service Invocation Framework (SIF) that leverages Oracle Workflow Java Business Event System (JBES) and a seeded Java rule function to allow any WSDL-described service to be invoked.

By using this service invocation framework, developers or implementors can interact with Web services through WSDL descriptions instead of working directly with SOAP APIs, the usual programming model. This approach lets you use WSDL as a normalized description of disparate software, and allows you to access this software in a manner that is independent of protocol or location.

Since this feature is the major development framework in invoking Web services within the entire Oracle E-Business Suite, detailed implementation information is described in a separate chapter in this book.

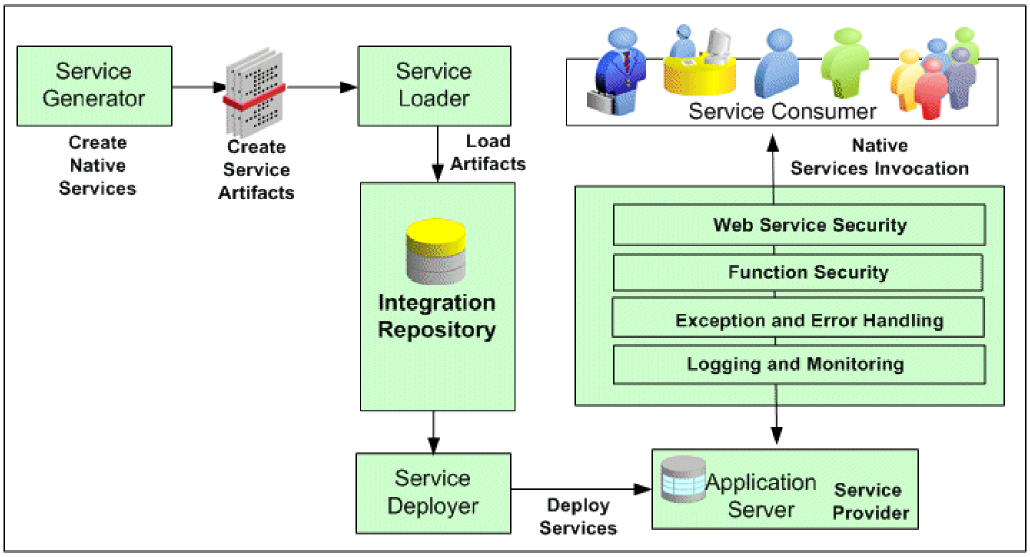
See Implementing Service Invocation Framework, page 9-1.

Native Service Enablement Architecture Overview

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

The following diagram illustrates the integration architecture flow between each SOA component:

Native Service Enablement Architecture Flow Diagram



All the native packaged public integration interfaces are published in the Oracle Integration Repository by default. Users who have the Integration Repository Administrator role can then transform these native integration interfaces into web services by service generator. Service loader uploads service artifacts to Oracle Integration Repository. Service deployer deploys service artifacts from the Integration Repository to the application server where services can be exposed to customers through service provider.

Service provider identifies and processes inbound SOAP requests from service consumers or web service clients, reinforces function security and web service security, as well as passes all SOAP request and response messages to SOA Monitor (if the monitoring feature is enabled) for further monitoring to ensure the seamless service invocations throughout the entire service 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 provides 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.

SOAP Service Enablement at Design Time

All the native packaged public integration interfaces are published in Oracle Integration Repository by default.

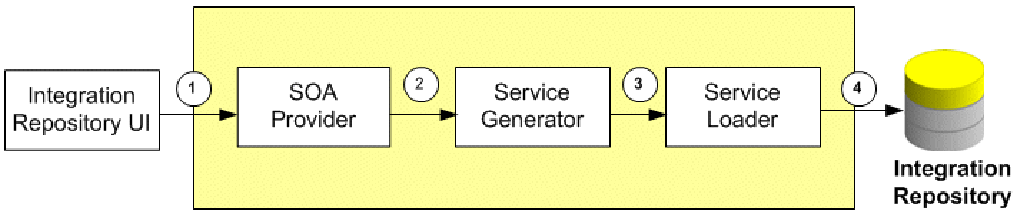
At design time, an integration repository administrator performs the web service generation task through the Integration Repository user interface. This sends a request for service generation and invokes the Service Generator to create service artifacts and stores them in the application server file system. Service loader then uploads these artifacts to Oracle Integration Repository.

These tasks are completed through the Oracle Integration Repository user interfaces and they are actually executed by SOA Provider behind the scenes.

> **Important:** XML Gateway maps can be supported by both Web Service Provider in Release 12.0 and by SOA Provider in this release. For backward compatibility in supporting the XML Gateway Map service enablement, a profile option *FND: XML Gateway Map Service Provider (FND_SOA_SERVICE_PROVIDER)* is used to let you select an appropriate service provider in enabling services for XML Gateway Map interfaces. Based on the selected profile value, you may find the Web Service - SOA Provider region, or Web Service - Web Service Provider region, or both regions displayed in the XML Gateway Interface Details Page. See XML Gateway Map Web Service Region, *Oracle E-Business Suite Integrated SOA Gateway User's Guide*.

The following diagram illustrates the service generation function flows at design time:

Design Time Functional Flow



1. Integration Repository UI sends a request for a service generation to SOA Provider.

2. SOA Provider passes the request to Service Generator.

3. Service Generator generates the service artifact.

   > **Note:** Service artifacts are generated in the application server file system at location specified by system property `SOA_SERVER_TEMP_DIRECTORY_LOCATION`.
   >
   > These artifacts are located in `oc4j.property` of `OAFM container`. The file system storage structure on server (may use `System.getProperty (APPLRGF)`) can be as follows:

```
TEMP_LOCATION
      |_Type (such as PL/SQL, concurrent program, etc.)
             |_ClassID
                        (contains all SQL Wrapper
packages)
```

4. Service Loader uploads the service artifact to Oracle Integration Repository.

Integration repository administrators as defined by the Integration Repository Administrator role can further deploy the services from Oracle Integration Repository to the application server where services can be exposed to customers through service provider.

> **Note:** For information about the Integration Repository Administrator role, see Role-Based Access Control (RBAC) Security for Oracle E-Business Suite Integrated SOA Gateway, page 6-4.

REST Service at Design Time

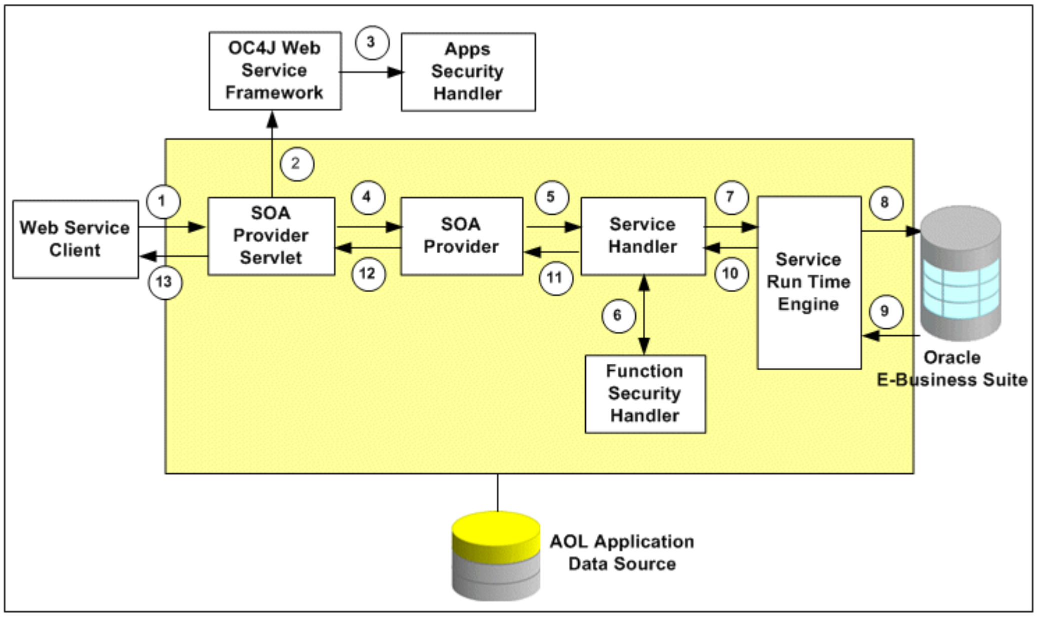
REST services are developed based on Oracle E-Business Suite technology infrastructure. At design time, an integration repository administrator can deploy a selected interface as a REST service. Additionally, the administrator can undeploy the service if needed.

SOAP Service Enablement Runtime

At runtime, a service consumer or web service client sends a SOAP request message. SOA Provider receives the request message and references integration services and data from Oracle Integration Repository in processing the request that invokes web service. Function security is enforced at this time to secure the web service content from unauthorized access. After passing security checks on the inbound request, SOA Provider then sends the SOAP response message out back to the web service client.

The following diagram illustrates the web service process flows between a web service client and Oracle E-Business Suite through SOA Provider at runtime:

Service Enablement Flow at Run Time



1. Web service client sends a SOAP request to WSDL URL that is redirected to SOA Provider Servlet.

2. The inbound SOAP message is passed to OC4J Web Service Framework.

3. The OC4J Web Service Framework authenticates the SOAP message based on the `wsse:security` headers. To validate username and password, the Framework calls Application Security Handler.

4. On authentication of the SOAP message user, the Framework hands over the message along with its context to SOA Provider.

5. SOA Provider hands over the request to Service Handler.

6. Service Handler calls the Function Security Handler to decide whether the user is authorized to execute the particular interface.

7. After passing authorization check, the request is passed on to Service Run Time Engine.

8. Service Run Time Engine executes the interface on Oracle E-Business Suite.

9. Response is returned back to the Service Run Time Engine.

10. Response is converted to a SOAP response and returned back to Service Handler.

11. Service Handler returns the SOAP response back to SOA Provider.

12. SOA Provider returns the SOAP response back to SOA Provider Servlet.

13. SOA Provider Servlet returns the SOAP response back to web service client.

Web Service Clients

To enable integration interfaces become web services, Oracle E-Business Suite Integrated SOA Gateway uses the following technologies or tools for web service enablement:

- Apache Axis

- .NET Web Service Client

- Oracle JDeveloper

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

- Oracle BPEL Process Manager (Oracle BPEL PM)

  Business process execution language (BPEL) is particularly used in orchestrating composite web services.

- Oracle Enterprise Service Bus (ESB)

  Similar to the composite service creation through BPEL, composite services can also be created using Oracle ESB.

2

Setting Up Oracle E-Business Suite Integrated SOA Gateway

Setup Overview

After successfully installing Oracle E-Business Suite Integrated SOA Gateway, implementors or integration repository administrators need to perform the following necessary setup tasks to enable its functions and establish the connection to application database schema at run time:

- Enabling ASADMIN User, page 2-1
- Creating a New Oracle E-Business Suite User Account, page 2-3
- Setting Profile Options, page 2-6

For detailed information on how to install and enable Oracle E-Business Suite Integrated SOA Gateway upgraded from earlier releases, see the *Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12*, My Oracle Support Knowledge Document 556540.1.

For troubleshooting information on potential problem symptoms and corresponding solutions for Oracle E-Business Suite Integrated SOA Gateway, see the *Oracle E-Business Suite Integrated SOA Gateway Troubleshooting Guide, Release 12*, My Oracle Support Knowledge Document 726414.1.

Enabling ASADMIN User

Before enabling `ASADMIN` user, make sure that all workflow agent listeners, service components, background engines, and notification mailers are all up and running. You can verify the information in Oracle Workflow Manager. Otherwise, the 'Apps Schema Connect Role' (`UMX|APPS_SCHEMA_CONNECT`) assignment will not be reflected correctly to the `ASADMIN` user after it is enabled.

> **Note:** Ensure that the APPLSYS.WF_ERROR queue is enabled in order for the Workflow Deferred Notification Agent Listener to run.

Use the following steps to enable ASADMIN user:

1. Log in to Oracle E-Business Suite as a user who has the Integration Repository Administrator role and choose the User Management responsibility in the Navigator.

2. Click the Users link from the navigation menu to open the User Maintenance window.

3. Enter information in the search area to locate the ASADMIN user.

4. Click the **Update** icon next to the ASADMIN user to open the Update User window.

5. Remove the Active To date field and click **Apply**.

6. Click the **Reset Password** icon next to the ASADMIN user to open the Reset Password window.

7. Enter new password twice and click **Submit**.

After the `ASADMIN` user is enabled from Oracle E-Business Suite, you must perform the following tasks:

- Verify if the `ASADMIN` user has the 'Apps Schema Connect Role' (UMX|APPS_SCHEMA_CONNECT) role in `wf_user_roles`.

  If the 'Apps Schema Connect Role' role is not present in the `wf_user_roles` for the `ASADMIN` user, then run the 'Workflow Directory Services User/Role Validation' concurrent program to grant the role.

- Reset the `ASADMIN` password in the file system.

  Update the file as shown below to reset the password:

  $INST_TOP/ora/10.1.3/j2ee/oafm/config/system-jazn-data.xml

  ```
  <user>
  <name>ASADMIN</name>
  <display-name>Default Apps SOA User</display-name>
  <description>Used by SOAProvider for DB connection</description>
  <credentials>!password</credentials>
  </user>
  ```

  > **Note:** The password should be preceded by an exclamation mark ('!') so that when OAFM is started, the password gets encrypted. Ensure that you enter the exclamation mark before the actual password value in the `<credentials>` tag. In addition, the

password of ASADMIN should be synchronized between the `system-jazn-data.xml` file and the database through the application user interfaces.

- Bounce the application tier and retry the generation process.

Creating a New Oracle E-Business Suite User Account

An appropriate Oracle E-Business Suite user account should be created to establish the Applications database connection at run time. If you do not want to use the default user ASADMIN who is enabled during the installation for database connection, alternatively you can create another Oracle E-Business Suite user account for establishing the connection.

Use the following steps to create an Oracle E-Business Suite user account in Oracle User Management and then configure user in technology stack:

1. Creating an Oracle E-Business Suite User Account, page 2-3

2. Granting 'Apps Schema Connect Role' to the User, page 2-5

3. Configuring User in Technology Stack, page 2-5

Creating an Oracle E-Business Suite User Account

Use Oracle User Management to create an Oracle E-Business Suite user account in addition to the default user `ASADMIN` that has been enabled during the installation.

You can create an account using either one of the following ways:

- Create a new user account without using an existing user, page 2-3

  This approach lets you create a new user account and add the newly created user information to the application schema and TCA schema simultaneously. And this new user will automatically become an Oracle E-Business Suite user.

- Create a user account by using an existing user, page 2-4

  If you have an existing Oracle E-Business Suite user that you want to use, then use this approach by first locating the user, and then creating an account associated with it.

To create a new user account without using an existing user:

1. Log in to Oracle E-Business Suite as a user who has the Integration Repository Administrator role and choose the User Management responsibility in the Navigator.

2. Click the Users link from the navigation menu to open the User Maintenance window.

3. To register or create a new person, select 'External Organization Contacts' from the Register drop-down list and click **Go**.

4. In the Register Business Contact window, enter appropriate information for the new user for whom you want to have an account created:

   - E-mail: Enter an appropriate e-mail address

   - First Name: Enter the first name of the person

   - Last Name: Enter the last name of the person

   - Organization: Select an appropriate organization

   - Phone Number: Optionally enter the person's contact number

   - In the Account Information region, select one of the following options for the account password:

     - Generate Automatically: This allows the system to automatically generate the password.

     - Enter Manually: The system prompts you to enter the password and a confirmation of the password.

5. Click **Submit**. A confirmation message appears indicating that the new Oracle E-Business Suite user account has been created.

To create a user account by using an existing user:

1. Log in to Oracle E-Business Suite as a user who has the Integration Repository Administrator role and choose the User Management responsibility in the Navigator.

2. Click the Users link from the navigation menu to open the User Maintenance window.

3. Enter information in the search area to locate the appropriate user for whom you wish to create an account.

   Please note that only those users who are part of the application schema and present in either HR database or TCA database will be displayed in the search result.

4. Click the **Create User** icon next to the person's name if the account does not exist. This opens the Create User Account window.

5. Enter the appropriate information in the Create User Account window including e-mail address, active dates, and password. Click **Submit**.

Granting Apps Schema Connect Role to the User

After creating an Oracle E-Business Suite user account, you will grant the 'Apps Schema Connect Role' (UMX|APPS_SCHEMA_CONNECT) user role to the user. Thus, the user will have the privilege to access the Applications database schema at run time.

To grant a user role to the user:

1. Log in to Oracle E-Business Suite as a user who has the Integration Repository 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 either an existing user account or the user account that you just created in Creating an Oracle E-Business Suite User Account, page 2-3. 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 the 'Apps Schema Connect Role' (UMX|APPS_SCHEMA_CONNECT). Choose this 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.

Configuring User in Technology Stack

Once a new user account is created and assigned with the 'Apps Schema Connect Role' role, you must configure the user in technology stack. This configuration allows the new user to be used by SOA Provider for the application database connection at runtime.

To configure the user in technology stack:

1. Set the context variable `s_soaprovider_user` to the new user that you just created in Creating an Oracle E-Business Suite User Account, page 2-3.

2. Use the following steps to modify `$INST_TOP/ora/10.1.3/j2ee/oafm/config/system-jazn-data.xml`:

   Note that this file will not be overwritten when AutoConfig is run; therefore, user name and password are preserved.

1. Change ASADMIN user to the new user that you just created in Creating an Oracle E-Business Suite User Account, page 2-3.

2. Change the password for the ASADMIN user to the password which was provided while creating the new user.

   > **Note:** The password should be preceded by an exclamation mark ('!') so that when OAFM oc4j is started, the password gets encrypted. Ensure that you enter the exclamation mark before the actual password value in the <credentials> tag.
   >
   > The password of ASADMIN should be synchronized between system-jazn-data.xml file and the database through the application user interfaces.
   >
   > The following example shows the information details on user name and password change:
   >
   > Existing Values:
   >
   > ```
   > <user>
   >     <name>ASADMIN</name>
   >     <display-name>Default Apps SOA User</display-
   > name>
   >     <description>Used by SOAProvider for DB
   > connection</description>
   >     <credentials>!password</credentials>
   > ```
   >
   > New Values:
   >
   > ```
   > <user>
   >     <name>New_User</name>
   >     <display-name>Default Apps SOA User</display-
   > name>
   >     <description>Used by SOAProvider for DB
   > connection</description>
   >     <credentials>!password</credentials>
   > ```

3. Run AutoConfig.

Setting Profile Options

Oracle E-Business Suite Integrated SOA Gateway uses profile options to set necessary parameters in determining appropriate service providers for XML Gateway Map service enablement and enabling SOA auditing feature.

Specifically, these profiles determine the following features:

- Appropriate service providers that are used in enabling services for XML Gateway Maps.

- The availability of the SOA Monitor auditing feature.

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 Service Provider	Use this profile option to select an appropriate service provider in enabling services for XML Gateway Map interface type. Based on the selected profile value, the interface details page displays an appropriate Web Service region or more than one region. You can select one of the following three profile values: • WSP - Web Service Provider If this profile value is selected, then the Web Service - Web Service Provider region will be displayed in the XML Gateway Map interface details page. • SOAP - SOA Provider If this profile value is selected, then the Web Service - SOA Provider region will be displayed in the XML Gateway Map interface details page. • BOTH - Both	Yes	SOAP (SOA Provider) **Important:** If you do not start from this release and your system has Web Service Provider based service integration for enabling generic XML Gateway services, set the profile option to 'Both' (Web Service Provider and SOA Provider) instead. This allows the Web Service - Web Service Provider region and the Web Service - SOA Provider region to be displayed simultaneously if Web services are available. Otherwise, Web Service Provider will be disabled and any invocations of generic XML Gateway Web services will return a fault message.

Profile Option	Description	Required	Default Value
	Web Service Provider and SOA Provider If this profile value is selected, then both the Web Service - Web Service Provider region and Web Service - SOA Provider region will be displayed in the XML Gateway Map interface details page.		
SOA: Web Service Audit	Use this profile option to enable/disable the SOA auditing feature. If it is enabled, SOAP request and response messages are audited in SOA Monitor.	Yes	ON Please note that this profile value can be overridden by clicking on the **Turn Off Audit** or **Turn On Audit** button in the SOA Monitor main page. For example, clicking **Turn Off Audit** will override the default value and disable the SOA Monitor auditing feature.

For information on how to set profile options, see the *Oracle E-Business Suite System Administrator's Guide - Maintenance*.

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 can perform administrative tasks in managing service lifecycle activities and as well as managing security grants.

To better understand how to administer and manage both SOAP and REST services, the following topics are included in this chapter:

- Administering SOAP Web Services, page 3-1
- Administering REST Web Services, page 3-19
- Managing Service Life Cycle and Security Grants Through Backend Processing, page 3-43

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)

> **Note:** The service for XML Gateway Map can be enabled by both Web Service Provider in release 12.0 and SOA Provider in this release. For backward compatibility, a profile option 'FND: XML Gateway Map Service Provider' is used to let you select an appropriate service provider in enabling services for XML Gateway Map interface type. For more information on service enablement for XML Gateway Map, see XML Gateway Map Web Service Region, *Oracle E-Business Suite Integrated SOA Gateway User's Guide*.

- Concurrent Program

> **Important:** Oracle Integration Repository supports REST service enablement for Open Interface Tables and Views. If a concurrent program is associated with an open interface table or view, this concurrent program can be viewed and displayed under the Open Interface type and can be available as a REST service.

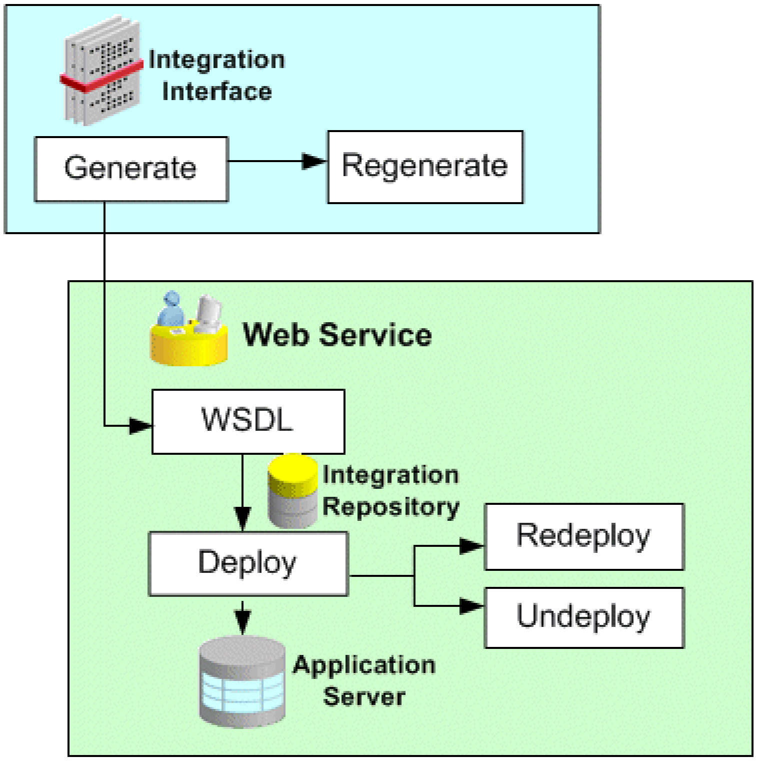
- Business Service Object

Note that Java APIs for Forms web services are desupported in Oracle E-Business Suite Release 12.2. If you are planning to use this type of interfaces as web services, you are advised to use alternate serviceable interfaces, such as PL/SQL and Business Service Objects interfaces, which can be deployed as web services. Refer to My Oracle Support Knowledge Document 966982.1 for the suggested alternatives to the existing Java APIs for Forms services.

Managing SOAP Service Lifecycle Activities

Only integration repository administrators (defined by the Integration Repository Administrator role) 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 3-5

  Oracle Integration Repository provides a capability of transforming interface definitions to a machine-processable format that complies with web standards using WSDL. Once the WSDL file is generated successfully, an appropriate Web Service region becomes visible in the interface details page for XML Gateway interface.

  For interfaces with the support for both REST and SOAP services, service generation is managed in the SOAP Web Service tab of the interface details page.

- **Deploying, Undeploying, and Redeploying SOAP Web Services**, page 3-10

  Integration repository administrators can further deploy the generated SOAP service to Oracle Application Server.

  If the SOAP service is successfully deployed, the administrators can undeploy the service if the service is no longer required for integration, or redeploy the service if needed.

- **Subscribing to Business Events**, page 3-14

  This task allows the administrators to subscribe to selected business events and create subscriptions for the selected events.

- **Creating Grants**, page 3-15

  This allows the administrators to create security grants by authorizing the access permission for a selected interface method, or a procedure or function to an appropriate user, a user group, or all users.

- **Viewing Generate and Deploy Time Logs for SOAP Services**, page 3-16

  This allows the administrators to view and download the logs recorded during service generation and deployment for specific services or all services that have the logging enabled at the Site level only.

Note that the administrators can also manage SOAP service lifecycle activities and create security grants using manual steps, see:

- Managing SOAP Service Lifecycle Activities Using Manual Steps, page 3-43

  This allows the administrators to generate SOAP services and perform postclone activities through backend scripts.

- Managing Security Grants Using an Ant Script, page 3-47

  This allows the administrators to create security grants through a backend script.

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, next to the **Integration Repository** tab. The **Administration** tab is specifically for the administrators to perform additional administrative tasks outside the Integration Repository user interface. This tab contains the following administrative and management features:

> **Note:** All Integration Repository Administration functions are grouped under Integration Repository Administrator permission set (FND_REP_ADMIN_PERM_SET) and performed by the users with the Integration Repository Administrator role.
>
> For more information about the Integration Repository Administrator permission set, see Role-Based Access Control (RBAC) Security, page 6-3.

- SOA Monitor: The SOA Monitor subtab allows the administrators to monitor and audit all SOAP messages in and out through SOA Provider and view the message details.

  For information about how to use SOA Monitor, see Monitoring and Managing SOAP Messages Using SOA Monitor, page 8-1.

- Log Configuration: The Log subtab allows the administrators to configure and manage log setups.

For information about log configuration, see Logging for Web Services, page 7-1.

Generating SOAP Web Services

Oracle E-Business Suite Integrated SOA Gateway allows users who have the Integration Repository Administrator role to transform interface definitions to SOAP services.

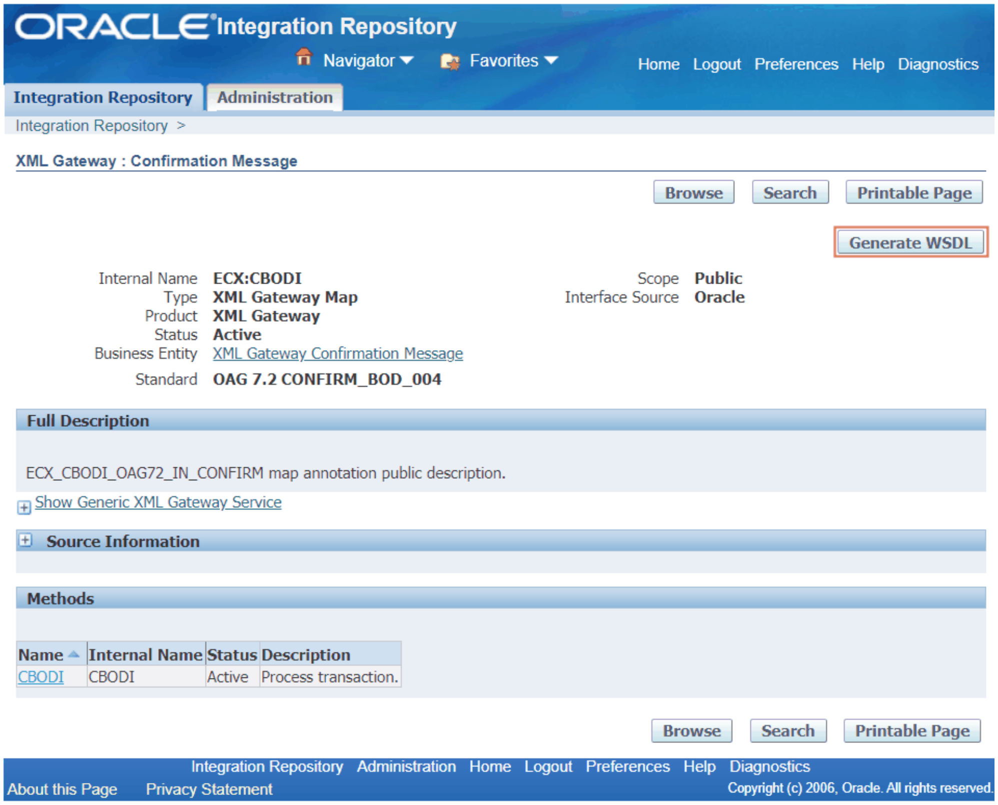
To accomplish this goal, these interface definitions will be transformed to a machine-processable format that complies with web standards using Web Services Description Language (WSDL). The WSDL code contains operations or messages that can be bound to a concrete network protocol and message format to define web services.

Generating SOAP Services

- For interfaces with the support for SOAP services only, such as XML Gateway maps, service activities are managed in the Web Service region. Click **Generate WSDL** to generate the service for a selected interface. Once the SOAP service is generated, service deployment activities are managed in the appropriate Web Service region.

> **Note:** A composite service consists of multiple native services which have WSDL files generated already; therefore, there is no service generation button shown in the composite service details page.

XML Gateway Interface Details Page with "Generate WSDL" Button Highlighted


ORACLE Integration Repository
Navigator
Favorites
Home Logout Preferences Help Diagnostics
Integration Repository
Administration
Integration Repository >
Browse
Search
Printable Page
Generate WSDL
Show Generic XML Gateway Service
Browse
Search
Printable Page
Integration Repository Administration Home Logout Preferences Help Diagnostics
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

XML Gateway : Confirmation Message

Internal Name **ECX:CBODI**
Type **XML Gateway Map**
Product **XML Gateway**
Status **Active**
Business Entity XML Gateway Confirmation Message
Standard **OAG 7.2 CONFIRM_BOD_004**

Scope **Public**
Interface Source **Oracle**

Full Description

ECX_CBODI_OAG72_IN_CONFIRM map annotation public description.

Source Information

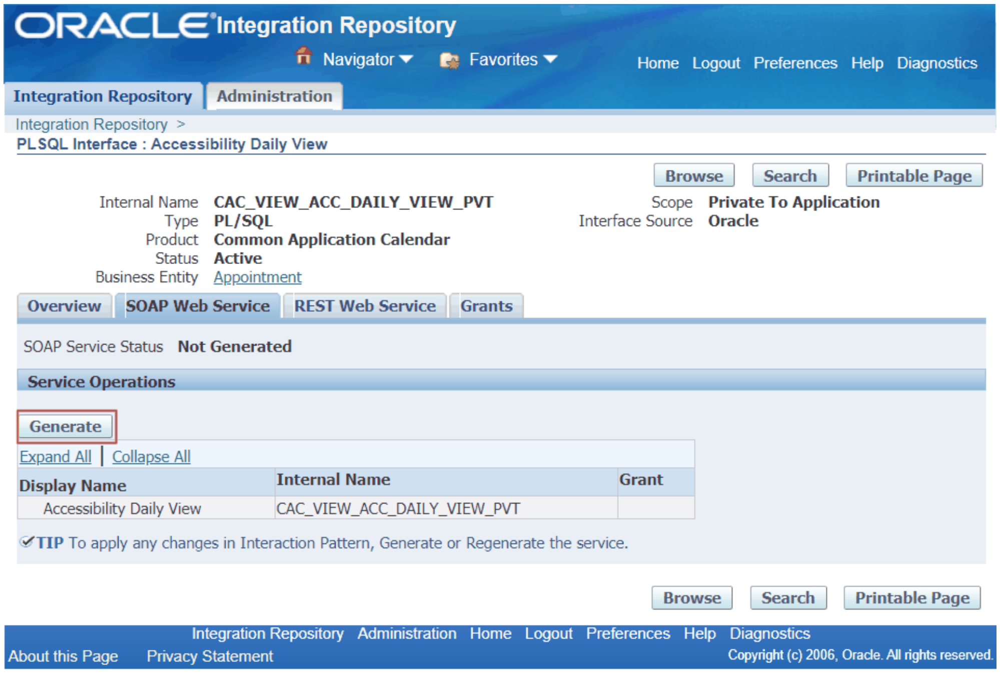
Methods

Name	Internal Name	Status	Description
CBODI	CBODI	Active	Process transaction.

- For interfaces with the support for both REST and SOAP services, such as PL/SQL, Concurrent Programs, and Business Service Objects, service generation and deployment activities are managed in the SOAP Web Service tab of the interface details page.

  To generate the SOAP service for a selected interface, click **Generate** (or **Generate WSDL** for a Business Service Object) in the SOAP Web Service tab.

SOAP Web Service Tab with "Generate" Button Highlighted



PLSQL Interface : Accessibility Daily View

Internal Name	CAC_VIEW_ACC_DAILY_VIEW_PVT	Scope	Private To Application
Type	PL/SQL	Interface Source	Oracle
Product	Common Application Calendar		
Status	Active		
Business Entity	Appointment		

SOAP Service Status **Not Generated**

Service Operations

Display Name	Internal Name	Grant
Accessibility Daily View	CAC_VIEW_ACC_DAILY_VIEW_PVT	

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

Copyright (c) 2006, Oracle. All rights reserved.

After SOAP Service Generation

For XML Gateway interfaces with the support for SOAP services only, once the SOAP service of a selected XML Gateway interface has been successfully generated, the Web Service region appears.

> **Note:** For XML Gateway Map interface type, you might find more than one Web Service Region available. See: Generating Web Service Region (s) for XML Gateway Map Interfaces, page 3-8.

For interfaces with the support for both SOAP and REST services, service generation information appears in the SOAP Web Service tab. For example, "Generated" is shown as the SOAP Service Status field in the SOAP Web Service tab.

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

  > **Important:** Multiple requests to generate services for an integration interface are not allowed. While a web service is getting generated, the status of the service is changed to 'Generating' and the **Generate WSDL** or **Generate** button is disabled.

- **View WSDL link:** Click this link to view the generated WSDL code.

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

- **Authentication Type:** Prior to deploying the generated service, an integration repository administrator must select at least one authentication type for the selected interface. The selected authentication type(s) will be used by SOA Provider to authenticate the users who want to access the generated SOAP service throughout the service deployment cycle.

  The supported authentication types are Username Token and SAML Token (Sender Vouches). For more information, see Deploying and Undeploying Web Services, page 3-10.

- **The Deploy Button:** This button appears allowing you to deploy the generated service.

  > **Note:** Composite services such as BPEL files are typically not deployed within Oracle E-Business Suite like other service enabled interface types. Instead, you need a separate BPEL PM (SOA Suite or third party BPEL PM server) to deploy the BPEL composite services. Therefore, you will not be able to find **Deploy** in the composite service details page.

  For more information, see Deploying and Undeploying SOAP Web Services, page 3-10.

Regenerating the SOAP Service If the Definition is Changed

If the interface definition is changed, the administrators need to regenerate the service from the Interface Details page by clicking **Regenerate WSDL** in the appropriate Web Service region or **Regenerate** in the SOAP Web Service tab. Upon regeneration, the service definition will also change to reflect the modification in the interface. The administrators will have to modify its web service clients based on the new service definition.

If interface definition is not changed, then regenerating the service would not change the service definition. The administrators can continue to use the existing web service clients, if any, with the new service definition.

Generating Web Service Region(s) for XML Gateway Map Interfaces

In supporting the Web Service Provider-based and SOA Provider-based service enablement for XML Gateway Map, if a SOAP service is successfully generated, depending on the profile value set in the *FND: XML Gateway Map Service Provider* profile option, you can find one of the following Web Service regions displayed in the XML Gateway interface details page:

- Web Service - Web Service Provider region

  In Release 12.0, XML Gateway Map interface type is deployed by default through

Web Service Provider. The administrators can find both the standard WSDL URL (`http://<hostname>:<port>/webservices/AppsWSProvider/oracle/apps/fnd/XMLGateway?wsdl`) and deployed ones available in this region.

- Web Service - SOA Provider region (default)

  The administrators will find a WSDL file along with **Deploy**, **Undeploy**, or **Redeploy**.

  > **Important:** If you do not start from this release and your system is upgraded from Oracle E-Business Suite Release 12.0, then the default value should be set to 'Both' (Web Service Provider and SOA Provider). This is because Web Service Provider could have already been used in enabling services for XML Gateway Maps in the Release 12.0. To continue to support service enablement in this release using SOA Provider and to support backward compatibility, both service providers should be enabled in transforming XML Gateway Map interface definitions into web services. The Web Service - Web Service Provider region and the Web Service - SOA Provider region can both be displayed simultaneously in the interface details page if web services are available.
  >
  > If you start with this release, then the default value remains the same which is SOAP (SOA Provider). Therefore, the Web Service Provider that generates the generic XML Gateway web services should be disabled at runtime.

- Both the Web Service - Web Service Provider region and Web Service - SOA Provider region

To generate a web service:

1. Log in to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility and 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. To generate a SOAP service for a selected interface:

   - If the selected interface is an XML Gateway map that can be exposed as a SOAP service only, click **Generate WSDL** to generate the service. Once the SOAP service is successfully generated, an appropriate Web Service region becomes available. The Web Service Status field marked as 'Generated' also appears which indicates that this selected interface has WSDL description available.

- If the selected interface can be exposed as both REST and SOAP services, click **Generate** in the SOAP Web Service tab to generate the service.

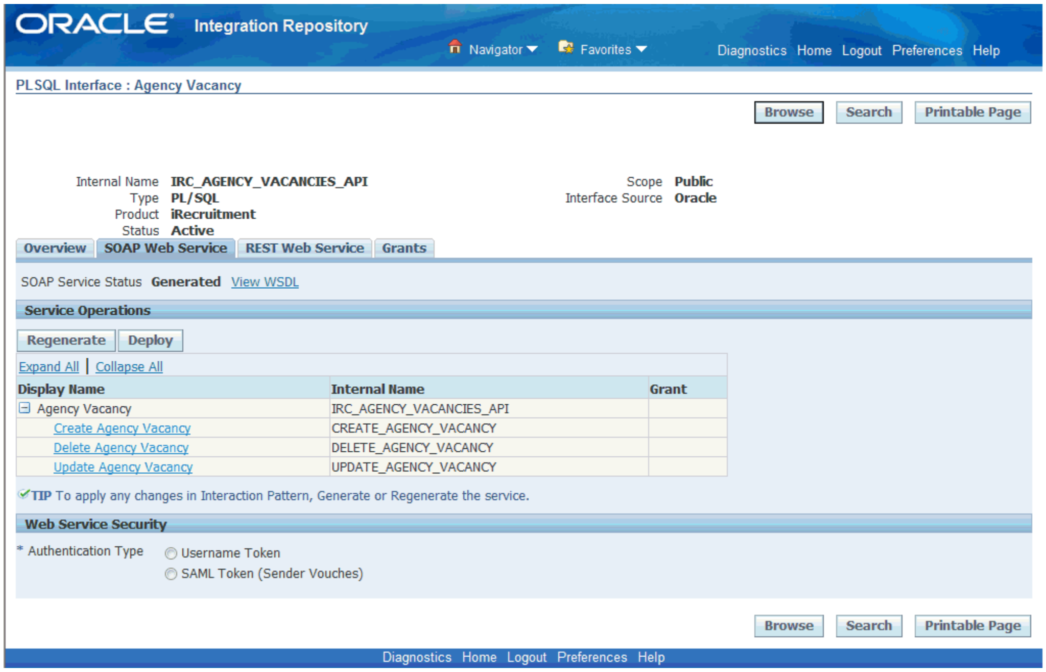
6. Click the **View WSDL** link to view the WSDL description.

7. Click **Regenerate WSDL** or **Regenerate** in the SOAP Web Service tab or Web Service region 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 appropriate Web Service region or the SOAP Web Service tab if the interface can be exposed as both SOAP and REST services.

Interface Details Page with SOAP Web Service Tab


ORACLE Integration Repository
Navigator
Favorites
Diagnostics Home Logout Preferences Help
Browse
Search
Printable Page
Overview
SOAP Web Service
REST Web Service
Grants
Regenerate
Deploy
Expand All | Collapse All
Diagnostics Home Logout Preferences Help

PLSQL Interface : Agency Vacancy

Internal Name	IRC_AGENCY_VACANCIES_API	Scope	Public
Type	PL/SQL	Interface Source	Oracle
Product	iRecruitment		
Status	Active		

SOAP Service Status **Generated** View WSDL

Service Operations

Display Name	Internal Name	Grant
⊟ Agency Vacancy	IRC_AGENCY_VACANCIES_API	
Create Agency Vacancy	CREATE_AGENCY_VACANCY	
Delete Agency Vacancy	DELETE_AGENCY_VACANCY	
Update Agency Vacancy	UPDATE_AGENCY_VACANCY	

✓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: Unlike native services that they are deployed through the Web Service region of an interface type detail page, composite services are typically not deployed within Oracle E-Business Suite like those of other service enabled interface types. You need a separate BPEL PM

(SOA Suite or third party BPEL PM server) to deploy the BPEL composite services. For example, a composite service - BPEL type can be deployed through Oracle JDeveloper to a BPEL server in Oracle SOA Suite BPEL PM (Process Manager) or a third party BPEL PM in a J2EE environment. This deployed composite service - BPEL project can update Oracle Applications if necessary.

Deploying Web Services with Authentication Types

To secure web service content, Oracle E-Business Suite Integrated SOA Gateway supports multiple authentication types for inbound service requests. Prior to deploying or redeploying a SOAP service generated, an integration repository administrator must first select at least one of the following authentication types:

- Username Token

  This authentication type provides username and password information in the security header for a web service provider to use in authenticating the users who request the Oracle E-Business Suite services. 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.

After you identify the preferred authentication type(s) for a web service, clicking the **Deploy** button deploys the web service with selected authentication type(s) from Oracle Integration Repository to Oracle Application Server. When SOA Provider receives inbound SOAP requests, the web service framework will authenticate the user who sends the SOAP request message based on the specified authentication type(s).

If no authentication type is identified for the service, then an validation error occurs requesting you to select an appropriate authentication type.

Once the web service has been successfully deployed, the Web Service Status field is changed to from "Generated" to "Deployed", along with selected authentication type check box(es). This indicates that the selected service has been successfully deployed to the application server.

For more information on supported authentication types for web service security, see Managing Web Service Security, page 6-9.

Reviewing Deployed WSDL

Once the web service has been successfully deployed, click the **View WSDL** link to view the deployed web service WSDL description. The following example shows the deployed WSDL code:

```
<?xml version="1.0"encoding="UTF-8" ?>
<definitions name="FNDWF_MOVE_MSGS_EXCEP2NORMAL"
targetNamespace="http://xmlns.oracle.
com/apps/owf/soaprovider/concurrentprogram/fndwf_move_msgs_excep2normal/
"
xmlns:tns="http://xmlns.oracle.
com/apps/owf/soaprovider/concurrentprogram/fndwf_move_msgs_excep2normal/
"
xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.
xmlsoap.org/wsdl/soap/"
xmlns:tns1="http://xmlns.oracle.
com/apps/owf/soaprovider/concurrentprogram/fndwf_move_msgs_excep2normal/
">
<types>
<schema xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="
qualified"
 targetNamespace="http://xmlns.oracle.
com/apps/owf/soaprovider/concurrentprogram/fndwf_move_msgs_excep2normal/
">
  <include schemaLocation="http://<hostname>:
<port>/webservices/SOAProvider/concurrentprogram/fndwf_move_msgs_excep2n
ormal/APPS_ISG_CP_REQUEST_CP_SUBMIT.xsd" />
</schema>
<schema xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="
qualified"
 targetNamespace="http://xmlns.oracle.
com/apps/owf/soaprovider/concurrentprogram/fndwf_move_msgs_excep2normal/
">
  <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>
   . . .
```

Additionally, the following buttons are available in the Web Service region or the SOAP Web Service tab if the interface can be exposed as both SOAP and REST services:

- **Redeploy**: It allows the administrator to redeploy the deployed web service for the following reasons:
  - The service was regenerated from a new interface definition.
  - The original service was corrupt.
  - The Authentication Type field was modified for a deployed service.

  Click **Redeploy** to update the deployed service with the current system values.

- **Undeploy**: It allows the administrator to undeploy the web service from Oracle Application Server back to Oracle Integration Repository if necessary.

  Any previously selected authentication type(s) for a service will be deselected by default.

If the administrator undeploys a native service that is not used, Oracle Integration Repository will undeploy the native service from the server.

> **Note:** The **Deploy** and **Redeploy** buttons appear only for users who have the Integration Repository Administrator role.

To deploy, undeploy, redeploy a web service:

1. Log in to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility and 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 SOAP Web Service tab (or Web Service region for XML Gateway interface type), select at least one of the following authentication type check boxes:

   - Username Token

   - SAML Token (Sender Vouches)

   Click **Deploy** to deploy the service from Oracle Integration Repository to Oracle Application Server.

   > **Note:** If this selected interface definition has never been generated as a web service, generate the service first and then deploy it. For information on how to generate a SOAP service, see Generating SOAP Web Services, page 3-5.

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

7. Redeploy the service if needed by clicking **Redeploy**.

   > **Note:** If changes are made to the Authentication Type field for a deployed web service, you must redeploy the service.

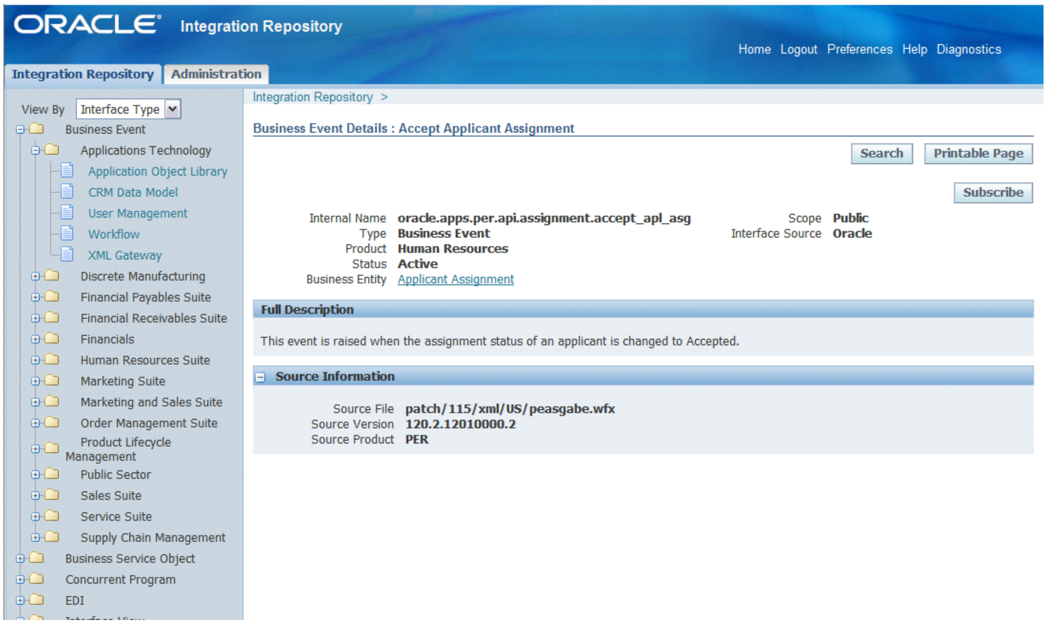
8. Click **Undeploy** to undeploy the service if necessary.

Subscribing to Business Events

Subscribing to Business Events

For business events, users who have the Integration Repository Administrator role can find **Subscribe** available in the interface details page which allows the administrators to subscribe to selected business events and create subscriptions for the selected events.

Business Event Details Page



Business Event Details : Accept Applicant Assignment

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

Scope Public
Interface Source Oracle

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

Source File patch/115/xml/US/peasgabe.wfx
Source Version 120.2.12010000.2
Source Product PER

Clicking the **Subscribe** button lets you subscribe to the selected business event. 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 WF_BPEL_Q, then the event message will be enqueued in WF_EVENT_T structure to Advanced Queue WF_BPEL_Q.

Unsubscribing to Business Events

The **Unsubscribe** button becomes available in the business event details page if the selected event has been subscribed. Clicking the **Unsubscribe** button removes the event subscription from 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*.

To subscribe to a business event:

1. Log in to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility and 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.

To remove the subscribed event, click **Unsubscribe** to remove or delete the event subscription if needed.

Managing Security Grants for SOAP Web Services Only

By leveraging Oracle User Management function security and data security, Oracle E-Business Suite Integrated SOA Gateway provides a security mechanism which only allows users who have authorized privileges to access or execute certain API methods exposed through Oracle Integration Repository. This protects application data from unauthorized access without security checks.

In this release, XML Gateway (inbound) is the only interface type that can be exposed as SOAP services only. To manage user security for XML Gateway interfaces, you need to log in to Oracle XML Gateway user interface. See Managing XML Gateway User Security in the Trading Partner User Setup Form, page 3-15.

> **Note:** For interfaces that can be exposed as REST services, security grants are managed in the Grants tab of the selected interface details page. For example, PL/SQL APIs, Concurrent Programs, and Business Service Objects can be exposed as both SOAP and REST services; Java Bean Services, Application Module Services, Open Interface Tables, and Open Interface Views can be exposed as REST services only.
>
> Note that when a method access permission is authorized to a grantee, if the selected method can be exposed as both SOAP and REST service operations, then this grants the permission to the associated SOAP and REST services simultaneously. For information on managing security grants in the Grants tab, see Managing Security Grants for SOAP and REST Web Services, page 3-38.

Managing XML Gateway User Security in the Trading Partner User Setup Form

For XML Gateway interfaces, user security is managed in the Oracle XML Gateway user interface through the Trading Partner User Setup form where the administrator needs to associate users with a trading partner. Only these authorized users can perform XML Gateway inbound transactions with the trading partner. Specifically, the administrator needs to:

- Set the "ECX: Enable User Check for Trading Partner" profile option to "Yes" to enable the trading partner specific security feature

- Associate users with a trading partner

  Log in to Oracle E-Business Suite as a user who has the XML Gateway responsibility. Navigate to Setup and then select Define Trading Partners from the navigation menu. In the Define Trading Partner Setup form, click the User Setup button to access the Trading Partner User Setup form.

For more information about trading partner user security, refer to Trading Partner Setup, XML Gateway Setup chapter, *Oracle XML Gateway User's Guide*.

Viewing Generate and Deploy Time Logs

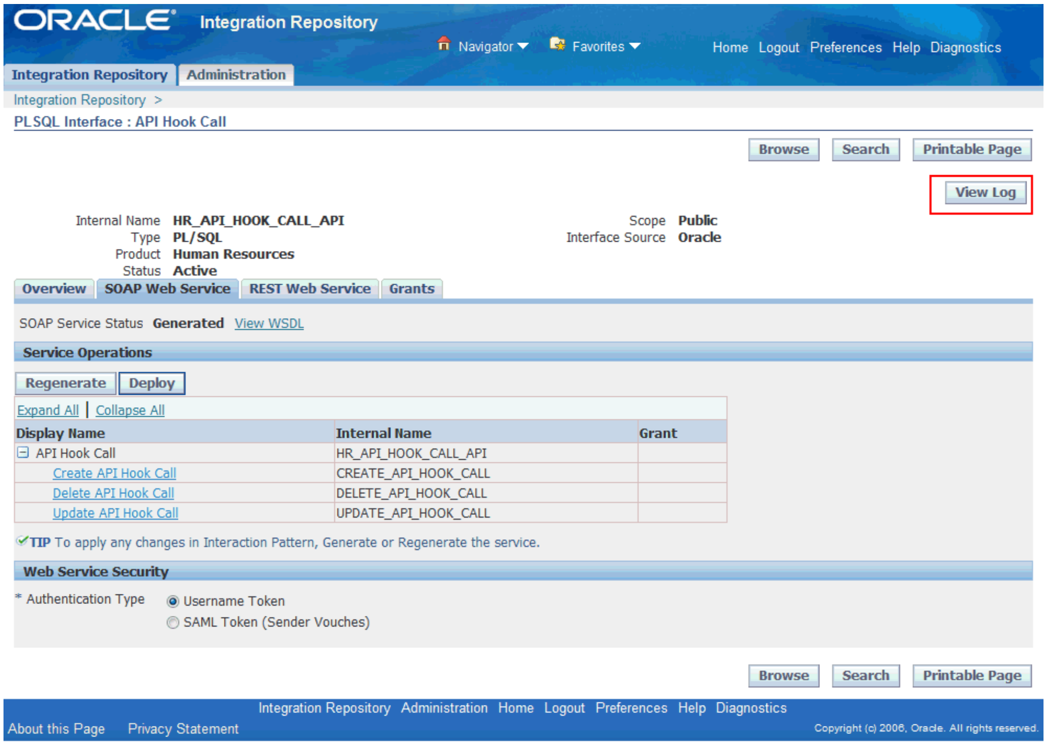
To effectively troubleshoot any issues or exceptions encountered at design time during service generation and deployment, error messages and service activity information can be logged and viewed through the Interface Details page if logging is enabled for specific services or all services at the Site level only. Administrators can find **View Log** displayed in the Interface Details page.

> **Note:** Logging is supported for SOAP services only.

Note that if logging is enabled for 'All Services' at the Site level, then **View Log** will be shown in the Interface Details page for all interfaces that can be service enabled. If the logging is enabled at the Site level for specific operations, then there will be no log generated and you will not find **View Log** in the Interface Details page. Generate and Deploy time log is only available if logging is enabled for specific services or all services at the Site level.

> **Note:** You will not find **View Log** available in the Interface Details page for a given service if the logging is enabled at the user level. Only site level logging configuration with specific services or all services will have the Generate and Deploy time logs captured.
>
> For information on how to configure logging at the site level for enabling specific services, see Enabling Logging at the Service Level, page 7-9.

SOAP Web Service Tab with 'View Log' Highlighted



Internal Name **HR_API_HOOK_CALL_API**
Type **PL/SQL**
Product **Human Resources**
Status **Active**

Scope **Public**
Interface Source **Oracle**

SOAP Service Status **Generated** View WSDL

Display Name	Internal Name	Grant
⊟ API Hook Call	HR_API_HOOK_CALL_API	
Create API Hook Call	CREATE_API_HOOK_CALL	
Delete API Hook Call	DELETE_API_HOOK_CALL	
Update API Hook Call	UPDATE_API_HOOK_CALL	

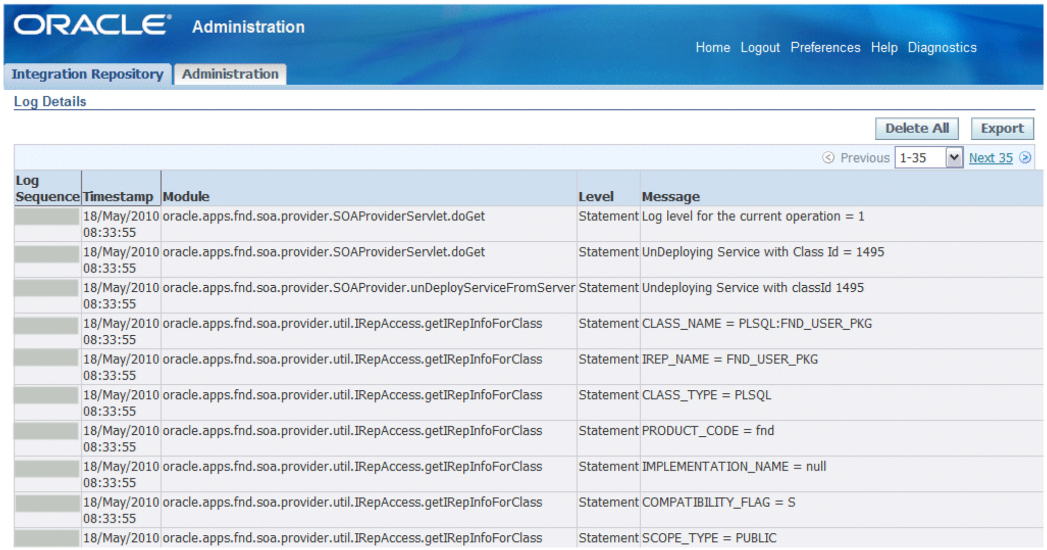
✓TIP To apply any changes in Interaction Pattern, Generate or Regenerate the service.

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

Click **View Log** to open the Log Details page. All 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.

> **Note:** Generate and Deploy time logs might be present for a service that has the logging enabled at the site level even if no error occurs while generating (regenerating) and deploying (redeploying or undeploying) the service. For example, if log setup is done at the log level of Statement, then statement level log messages can be written and listed in the log table.

Log Details Page



Log Sequence	Timestamp	Module	Level	Message
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.SOAProviderServlet.doGet	Statement	Log level for the current operation = 1
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.SOAProviderServlet.doGet	Statement	UnDeploying Service with Class Id = 1495
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.SOAProvider.unDeployServiceFromServer	Statement	Undeploying Service with classId 1495
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	CLASS_NAME = PLSQL:FND_USER_PKG
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	IREP_NAME = FND_USER_PKG
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	CLASS_TYPE = PLSQL
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	PRODUCT_CODE = fnd
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	IMPLEMENTATION_NAME = null
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	COMPATIBILITY_FLAG = S
[redacted]	18/May/2010	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	SCOPE_TYPE = PUBLIC

Deleting and Exporting Logs Listed in the Log Details Page

After viewing log messages retrieved for a service in the Log Details page, you can delete them if needed by clicking **Delete All**. A warning message appears alerting you that this will permanently delete all the logs retrieved in the page. Click **Yes** to confirm the action. An empty log table appears in the page after logs are 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 page to Microsoft Excel and save them to a designated directory and use it later.

For more logging information, see Logging for Web Services, page 7-1. For information on how to add a new log configuration, see Adding a New Log Configuration, page 7-5.

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

To view Generate and Deploy time log messages:

1. Log in to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility and 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 logs are available for this selected service that has logging enabled properly, you will find **View Log** available in the interface details page.

6. Click **View Log** to open the Log Details page where you can view the log details.

7. Click **Delete All** to delete all the logs listed in the table if needed. Click **Yes** to confirm the action. Click **No** to return back to the Log Details page.

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

Administering REST Web Services

In addition to supporting SOAP-based service generation and deployment, Oracle E-Business Suite Integrated SOA Gateway allows supported interface types to become REST-based services. REST services can be used for user-driven applications such as mobile, tablet, or handheld devices. In this release, PL/SQL APIs, Concurrent Programs, and Business Service Objects can be exposed as both SOAP and REST services; Java Bean Services and Application Module Services, Open Interface Tables, and Open Interface Views can be exposed as REST services only.

> **Note:** Security services are also REST services; however, unlike other service-enabled interfaces, they are predefined and predeployed REST services from Oracle Application Object Library. This type of services provides security related features for mobile applications. See: Supporting Security Services - Predeployed REST Services, page 3-20.

REST 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 supported interface can be exposed as a REST service through a 'Deploy' action. This deploys a REST service to an Oracle E-Business Suite application 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.

Note that the administrator can also manage these REST service activities through manual steps. See: Managing REST Service Lifecycle Activities Using Manual Steps, page 3-45.

The administrator can manage security grants in the Grants tab of the interface details page or through a backend script. It assigns grants to specific users to access or invoke the deployed REST services. For information on managing security grants using a script, see Managing Security Grants Using An Ant Script, page 3-47.

Supporting Security Services - Predeployed REST Services

In addition to exposing a supported interface as a REST service, Oracle E-Business Suite Integrated SOA Gateway supports Oracle Application Object Library's Authentication and Authorization services as REST security services. Security services are used for mobile applications to validate or invalidate user credentials, initialize user sessions with applications context, and authorize users.

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 3-28 .

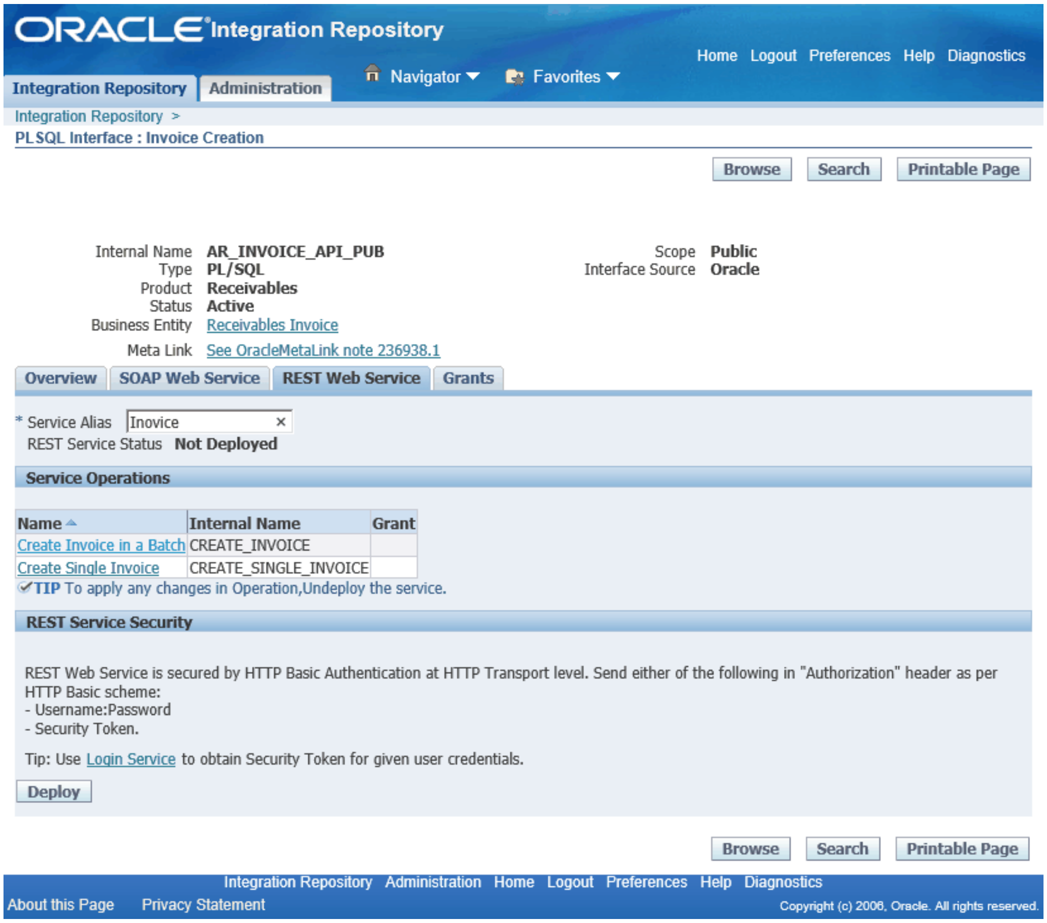
To better understand each administrative task, the following topics are included in this section:

- Deploying REST Web Services, page 3-20
- Undeploying REST Web Services, page 3-36
- Managing Grants for Interfaces with Support for SOAP and REST Services, page 3-38

Deploying REST Web Services

Oracle E-Business Suite Integrated SOA Gateway allows the administrator to deploy interface definitions as REST services. These interfaces are PL/SQL APIs, Concurrent Programs, Business Service Objects, Java Bean Services, Application Module Services, Open Interface Tables, and Open Interface Views. Among these interfaces, only PL/SQL APIs, Concurrent Programs, and Business Service Objects can be exposed as both SOAP and REST services.

Interface Details Page with REST Web Service Tab



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

* Service Alias: Invoice

REST Service Status: **Not Deployed**

Service Operations

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.

Deploy
Browse
Search
Printable Page
Integration Repository Administration Home Logout Preferences Help Diagnostics
About this Page
Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

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 is 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 WADL and schema for a selected service operation CREATE_INVOICE as follows:

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

  Guidelines for Entering Service Alias

  - Use simple and meaningful name to represent the service, such as "person", "employee", and so on.

  - Do not use "rest", "soap", and "webservices" as the alias.
  - Do not start with a number or a special character, such as #, $, %, _, `-` and more.
  - Do not end with a special character.
  - Characters such as `.`, _, and `-` are allowed in a service alias.

- **Select Desired HTTP Verbs for Java Bean Services, Application Module Services, Business Service Objects, Open Interface Tables, and Open Interface Views**

  For Java Bean Services, Application Module Services, Business Service Objects, Open Interface Tables, and Open Interface Views, in addition to selecting desired methods to be exposed as REST service operations, the administrator needs to select HTTP method check boxes for the desired methods.

  The following table lists the interfaces that can be exposed as REST services and their supported HTTP methods:

REST-based Interfaces with Supported HTTP Methods

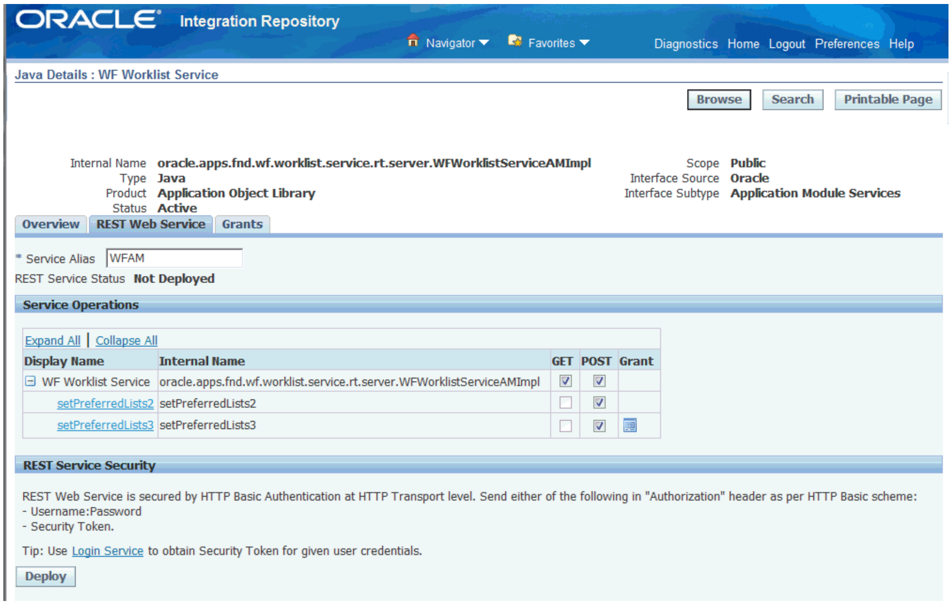
Interface Type	Supported HTTP Method(s)
PL/SQL API	POST only
Concurrent Program	POST only
Business Service Object	POST and GET
Java Bean Service	POST and GET
Application Module Service	POST and GET
Open Interface Table (Inbound)	POST, GET, PUT, and DELETE
Open Interface Table (Outbound)	GET only
Open Interface View	GET only

> **Note:** PL/SQL APIs and Concurrent Programs can be exposed as REST services with the POST HTTP method only; therefore, there is

no need to further specify the HTTP method for these two interface types.

- *For Java Bean Services and Application Module Services*

REST Web Service Tab for Application Module Services



Java Details : WF Worklist Service

Internal Name oracle.apps.fnd.wf.worklist.service.rt.server.WFWorklistServiceAMImpl
Type Java
Product Application Object Library
Status Active
Scope Public
Interface Source Oracle
Interface Subtype Application Module Services

* Service Alias WFAM
REST Service Status **Not Deployed**

Service Operations

Display Name	Internal Name	GET	POST	Grant
⊟ WF Worklist Service	oracle.apps.fnd.wf.worklist.service.rt.server.WFWorklistServiceAMImpl	☑	☑	
setPreferredLists2	setPreferredLists2	□	☑	
setPreferredLists3	setPreferredLists3	□	☑	▤

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.

If the Java or Application Module method is annotated (`rep:httpverb`) with a specific HTTP method, then the corresponding HTTP method check box is preselected for that method in the table.

- If the `GET` HTTP method is not annotated, then the `GET` check box becomes inactive or disabled for further selection. This means that the Java or Application Module method will never be deployed as a REST service operation with the `GET` method.

- If the `POST` HTTP method is not annotated, unlike the `GET` method, the `POST` check box is still active or enabled by default. This allows the administrator to select the `POST` check box if needed for the Java or Application Module method as a REST service operation before deploying the service.

For example, if "Add Grant" method within the "REST Service Locator" is annotated only with the `POST` HTTP method, then the `POST` method check box is preselected for the method. The `GET` method check box that is not annotated for the "Add Grant" method is shown as inactive or disabled which cannot be chosen for that method before deploying the service.

The administrator can modify the desired HTTP methods before deploying the REST service. For example, uncheck the preselected `POST` check box if the "Add Grant" method will not be exposed as a REST service operation with the `POST` method.

For information about the `rep:httpverb` annotation, see rep:httpverb, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*. For more Java Bean Services annotation guidelines, see Annotations for Java Bean Services, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

For more Application Module Services annotation guidelines, see Annotations for Application Module Services, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

For example, if "Add Grant" method within the "REST Service Locator" Java Bean Service is annotated only with the POST HTTP method, then the POST method check box is preselected for the method. The GET method check box that is not annotated for the "Add Grant" method is shown as inactive or disabled which cannot be chosen for that method before deploying the service.

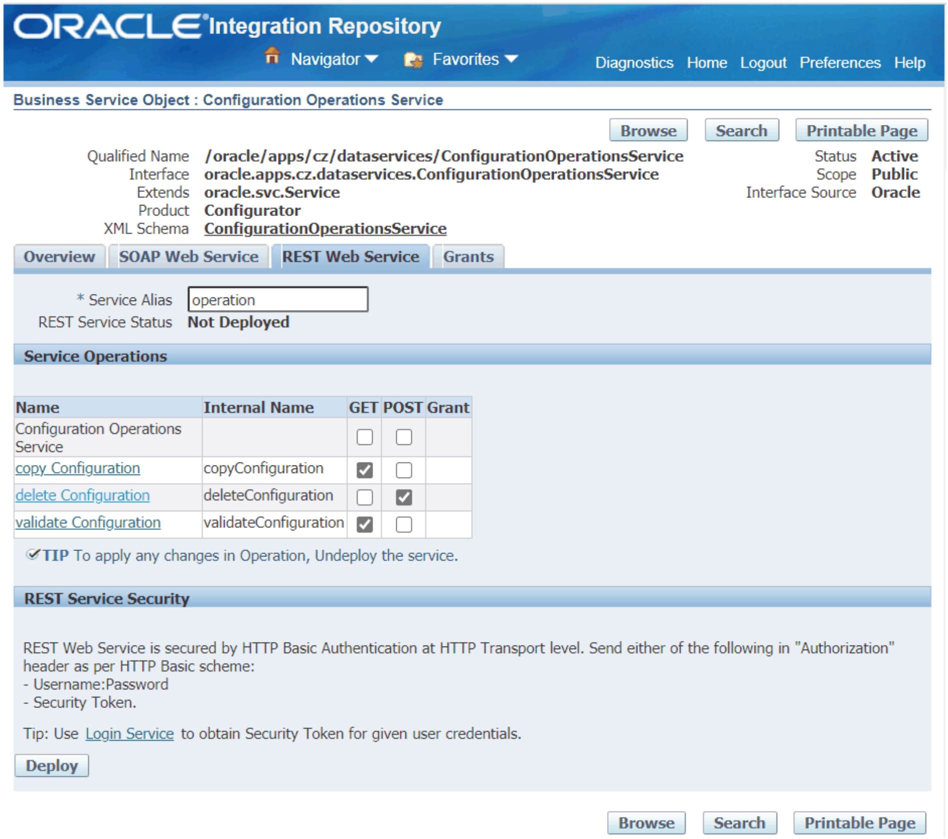
The administrator can modify the desired HTTP methods before deploying the REST service. For example, uncheck the preselected POST check box if the "Add Grant" method will not be exposed as a REST service operation with the POST method.

For information about the `rep:httpverb` annotation, see rep:httpverb, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*. For more Java Bean Services annotation guidelines, see Annotations for Java Bean Services, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

For more Application Module Services annotation guidelines, see Annotations for Application Module Services, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

- ***For Business Service Object***

REST Web Service Tab for Business Service Objects with GET and POST Methods



Business Service Object : Configuration Operations Service

Qualified Name	/oracle/apps/cz/dataservices/ConfigurationOperationsService	Status	Active
Interface	oracle.apps.cz.dataservices.ConfigurationOperationsService	Scope	Public
Extends	oracle.svc.Service	Interface Source	Oracle
Product	Configurator		
XML Schema	ConfigurationOperationsService		

* Service Alias	operation
REST Service Status	Not Deployed

Service Operations

Name	Internal Name	GET	POST	Grant
Configuration Operations Service		☐	☐	
copy Configuration	copyConfiguration	☑	☐	
delete Configuration	deleteConfiguration	☐	☑	
validate Configuration	validateConfiguration	☑	☐	

✓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.

A Business Service Object interface can be exposed as a REST service operation with the support of the GET and POST methods.

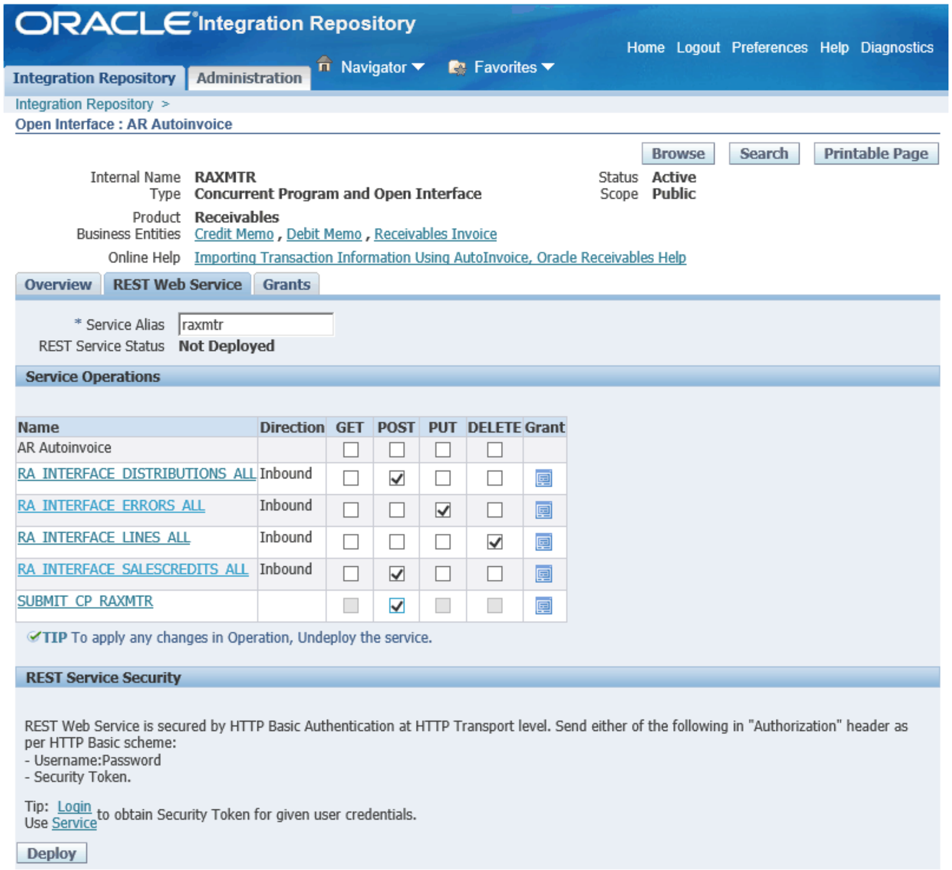
- The GET check box is enabled only if input parameters of the selected interface are of simple data types (String, Number, etc.). The check box is disabled if input parameters consist of complex data object types (AccountMergeRequest, etc.).
- There is no annotation required for enabling or using the GET and POST methods.

The administrator can select desired methods for an operation before deploying the Business Service Object interface as a REST service.

- ***For Open Interface Tables and Open Interface Views***

  For **open interface tables**, the supported HTTP methods are determined by the direction of the open interfaces.

REST Web Service Tab for Open Interface Table



Open Interface : AR Autoinvoice

Internal Name	RAXMTR	Status	Active
Type	Concurrent Program and Open Interface	Scope	Public
Product	Receivables		
Business Entities	Credit Memo , Debit Memo , Receivables Invoice		
Online Help	Importing Transaction Information Using AutoInvoice, Oracle Receivables Help		

* Service Alias: raxmtr

REST Service Status: Not Deployed

Service Operations

Name	Direction	GET	POST	PUT	DELETE	Grant
AR Autoinvoice		□	□	□	□	
RA_INTERFACE_DISTRIBUTIONS_ALL	Inbound	□	☑	□	□	▤
RA_INTERFACE_ERRORS_ALL	Inbound	□	□	☑	□	▤
RA_INTERFACE_LINES_ALL	Inbound	□	□	□	☑	▤
RA_INTERFACE_SALESCREDITS_ALL	Inbound	□	☑	□	□	▤
SUBMIT_CP_RAXMTR		□	☑	□	□	▤

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

- An open interface table with `Inbound` direction
  - All four HTTP methods (`GET`, `POST`, `PUT`, and `DELETE`) are available for selection.
    - `GET` - It reads or selects one or more records from the open interface table or view.
    - `POST` - It creates or inserts one or more records to the open interface table.
    - `PUT` - It updates or edits one or more records in the open interface table.
    - `DELETE` - It deletes or removes one or more records from the open interface table.
  - An additional method called `SUBMIT_CP_<internal name of the`

`associated concurrent program>` appears as the last entry of the method table with the `POST` HTTP method only.

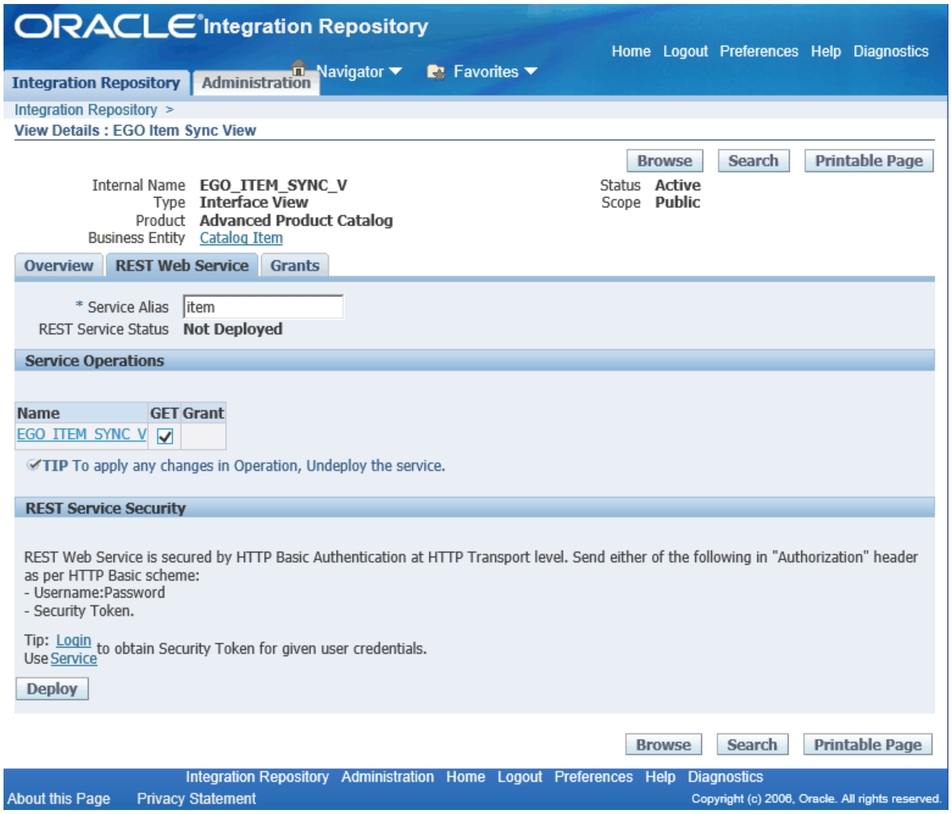
Please note that open interface is a combination of a concurrent program and associated open interface tables. Therefore, all these components including each open interface table and the concurrent program contained in a selected open interface table should be service enabled if desired. You can submit the associated concurrent program through this SUBMIT_CP `POST` service operation which is internally mapped to the "process" method of the associated concurrent program.

- An open interface table with `Outbound` direction

  For open interface tables with `Outbound` direction, only the `GET` method is supported.

For **open interface views** which are always with `Outbound` direction, only the `GET` method is supported.

REST Web Service Tab for Open Interface View



ORACLE Integration Repository
Home Logout Preferences Help Diagnostics
Integration Repository
Administration
Navigator
Favorites

Integration Repository >

View Details : EGO Item Sync View

Internal Name	EGO_ITEM_SYNC_V	Status	Active
Type	Interface View	Scope	Public
Product	Advanced Product Catalog		
Business Entity	Catalog Item		

* Service Alias: item

REST Service Status: Not Deployed

Service Operations

Name	GET Grant
EGO_ITEM_SYNC_V	☑

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.

Deploy
Browse
Search
Printable Page
Integration Repository Administration Home Logout Preferences Help Diagnostics
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

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 as `Cookie` in HTTP header. 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 6-9.

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

After Service Deployment

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

- **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** (PL/SQL APIs and Concurrent Programs Only): This field appears only if the selected interface is a PL/SQL API or concurrent program.

  'POST' appears by default because it is the only supported HTTP method for PL/SQL APIs and Concurrent Programs.
- **Service Operations**: This table displays the list of methods (or procedures and functions) contained in the selected interface.
  - If the selected interface is a PL/SQL API or concurrent program, then the Included Operations column will be checked for all the methods contained in the selected interface.

    By default, all methods in a PL/SQL API appear with the POST HTTP method. A concurrent program contains only one method which also appears with the POST method.

- If the selected interface is an interface type of Business Service Objects, Java Bean Services, or Application Module Services, then the GET and POST columns will be displayed with the included operation marks indicating which HTTP methods have been used to assist the REST service operations.

- If the selected interface is an open interface table with Inbound direction, then all four HTTP methods (`GET`, `POST`, `PUT`, and `DELETE`) will appear with the included operation marks indicating which HTTP methods have been used to assist the REST service operations.

- If the selected interface is an open interface table with Outbound direction or an open interface view, then only the `GET` column will appear with the included operation marks for the methods that have been exposed as REST service operations.

- 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:** 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.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<application xmlns:tns="http://xmlns.oracle.
com/apps/ar/soaprovider/plsql/rest/ar_invoice_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>
```

If the deployed REST service is an interface type of Java Bean Services or Application Module Services, then both GET and POST can be shown as the supported methods. For example, the following WADL description shows two Java methods contained in the Employee Information service. The `getAllReports` operation is implemented with the GET method, and the `getPersonInfo` operation is implemented with both the POST and GET HTTP methods.

```
<xml version="1.0" encoding="UTF-8">
<application name="EmployeeInfo" targetNamespace="http://xmlns.oracle.
com/apps/per/soaprovider/pojo/employeeinfo/"
 xmlns:tns="http://xmlns.oracle.
com/apps/per/soaprovider/pojo/employeeinfo/"
 xmlns="http://wadl.dev.java.net/2009/02" xmlns:xsd="http://www.w3.
org/2001/XMLSchema"
 xmlns:tns1="http://xmlns.oracle.
com/apps/fnd/rest/empinfo/getallreports/"
 xmlns:tns2="http://xmlns.oracle.
com/apps/fnd/rest/empinfo/getdirectreports/"
 xmlns:tns3="http://xmlns.oracle.
com/apps/fnd/rest/empinfo/getpersoninfo/">

<grammars>
   ...
</grammars>
<resources base="http://<hostname>:<port>/webservices/rest/empinfo/">
  <resource path="/getAllReports/">
 <method id="getAllReports" name="GET">
     <request>
     <param name="ctx_responsibility" type="xsd:string" style="query"
required="false" />
           <param name="ctx_respapplication" type="xsd:string" style="
query" required="false" />
    <param name="ctx_securitygroup" type="xsd:string" style="query"
required="false" />
    <param name="ctx_nlslanguage" type="xsd:string" style="query"
required="false" />
    <param name="ctx_orgid" type="xsd:int" style="query" required="
false" />
   </request>
   <response>
     <representation mediaType="application/xml" type="tns1:
getAllReports_Output" />
     <representation mediaType="application/json" type="tns1:
getAllReports_Output" />
    </response>
   </method>
 </resource>
 ...
 <resource path="="/getPersonInfo/ {personId}/">
  <param name="personId" style="template" required="true" type="xsd:int"
/>
 <method id="getPersonInfo" name="GET">
     <request>
     <param name="ctx_responsibility" type="xsd:string" style="query"
required="false" />
           <param name="ctx_respapplication" type="xsd:string" style="
query" required="false" />
    <param name="ctx_securitygroup" type="xsd:string" style="query"
required="false" />
    <param name="ctx_nlslanguage" type="xsd:string" style="query"
required="false" />
    <param name="ctx_orgid" type="xsd:int" style="query" required="
false" />
   </request>
   <response>
     <representation mediaType="application/xml" type="tns3:
getPersonInfo_Output" />
     <representation mediaType="application/json" type="tns3:
getPersonInfo_Output" />
    </response>
   </method>
  </resource>
 <resource path="/getPersonInfo/">
```

```
<method id="getPersonInfo" name="POST">
    <request>
       <representation mediaType="application/xml" type="tns3:
getPersonInfo_Input" />
     <representation mediaType="application/xml" type="tns3:
getPersonInfo_Output" />
    </request>
      <response>
       <representation mediaType="application/xml" type="tns3:
getPersonInfo_Input" />
     <representation mediaType="application/xml" type="tns3:
getPersonInfo_Output" />
           </response>
   </method>
  </resource>
  </resource path>
</application>
```

If the deployed REST service is an open interface table with Inbound direction, then the service operation table displays all four HTTP methods. In the following WADL example for the `AR Autoinvoice` open interface table (associated concurrent program internal name `RAXMTR`), the `RA_INTERFACE_LINES_ALL` operation is implemented with all four HTTP methods, and the associated concurrent program `SUBMIT_CP_RAXMTR` is implemented with the POST method.

- Each open interface table name contained in the selected open interface "AR Autoinvoice" is shown in one resource entry (`<resource path>`) with the selected HTTP method(s). For example, table name `RA_INTERFACE_LINES_ALL` in this example is shown with four selected methods (`GET`, `POST`, `PUT`, and `DELETE`) in one resource entry, and the associated concurrent program `SUBMIT_CP_RAXMTR` with `POST` is contained in another resource entry.

- For the `GET` and `DELETE` methods, application context values, including Responsibility, Responsibility Application, Security Group, NLS Language, and Organization ID complex types, are passed as query strings in the RESTHeader element.

```
 <?xml version = '1.0' encoding = 'UTF-8'?>
<application name="RAXMTR" targetNamespace="http://xmlns.oracle.
com/apps/ar/rest/autoinvoice" xmlns="http://wadl.dev.java.net/2009/02"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns1="http://xmlns.
oracle.com/apps/ar/rest/autoinvoice/RA_INTERFACE_LINES_ALL">
 <grammars>
  <include href="http://<hostname>:<port>/webservices/rest/autoinvoice/?
XSD=RA_INTERFACE_LINES_ALL_post.xsd" xmlns="http://www.w3.
org/2001/XMLSchema"/>
  <include href="http://<hostname>:<port>/webservices/rest/autoinvoice/?
XSD=RA_INTERFACE_LINES_ALL_get.xsd" xmlns="http://www.w3.
org/2001/XMLSchema"/>
  <include href="http://<hostname>:<port>/webservices/rest/autoinvoice/?
XSD=RA_INTERFACE_LINES_ALL_put.xsd" xmlns="http://www.w3.
org/2001/XMLSchema"/>
  <include href="http://<hostname>:<port>/webservices/rest/autoinvoice/?
XSD=RA_INTERFACE_LINES_ALL_delete.xsd" xmlns="http://www.w3.
org/2001/XMLSchema"/>
  <include href="http://<hostname>:<port>/webservices/rest/autoinvoice/?
XSD=SUBMIT_CP_RAXMTR_post.xsd" xmlns="http://www.w3.org/2001/XMLSchema"
/>
 </grammars>
  <resources base="http://<hostname>:<port>/webservices/rest/autoinvoice
/"><resource path="RA_INTERFACE_LINES_ALL/">
 <method id="RA_INTERFACE_LINES_ALL_get" name="GET">
     <request>
    <param name="ctx_responsibility" type="xsd:string" style="query"
required="false"/>
           <param name="ctx_respapplication" type="xsd:string" style="
query" required="false" />
    <param name="ctx_securitygroup" type="xsd:string" style="query"
required="false" />
    <param name="ctx_nlslanguage" type="xsd:string" style="query"
required="false" />
    <param name="ctx_orgid" type="xsd:int" style="query" required="
false" />
    <param name="select" type="xsd:string" style="query" required="
false"/>
    <param name="filter" type="xsd:string" style="query" required="
false"/>
    <param name="sort" type="xsd:string" style="query" required="false"
/>
    <param name="offset" type="xsd:string" style="query" required="
false"/>
    <param name="limit" type="xsd:string" style="query" required="false"
/>
   </request>
   <response>
    <representation mediaType="application/xml" type="tns1:
RA_INTERFACE_LINES_ALL_Output"/>
    <representation mediaType="application/json" type="tns1:
RA_INTERFACE_LINES_ALL_Output" />
    <representation mediaType="text/csv" type="tns1:
RA_INTERFACE_LINES_ALL_Output"/>
   </response>
 </method>
 <method id="RA_INTERFACE_LINES_ALL_post" name="POST">
     <request>
    <representation mediaType="application/xml" type="tns1:
RA_INTERFACE_LINES_ALL_Input"/>
    <representation mediaType="application/json" type="tns1:
RA_INTERFACE_LINES_ALL_Input" />
    <representation mediaType="text/csv" type="tns1:
RA_INTERFACE_LINES_ALL_Input"/>
   </request>
   <response>
```

```
<representation mediaType="application/xml" type="tns1:
RA_INTERFACE_LINES_ALL_Output"/>
    <representation mediaType="application/json" type="tns1:
RA_INTERFACE_LINES_ALL_Output" />
    <representation mediaType="text/csv" type="tns1:
RA_INTERFACE_LINES_ALL_Output"/>
   </response>
 </method>
 <method id="RA_INTERFACE_LINES_ALL_put" name="PUT">
     <request>
    <representation mediaType="application/xml" type="tns1:
RA_INTERFACE_LINES_ALL_Input"/>
    <representation mediaType="application/json" type="tns1:
RA_INTERFACE_LINES_ALL_Input" />
    <representation mediaType="text/csv" type="tns1:
RA_INTERFACE_LINES_ALL_Input"/>
   </request>
   <response>
    <representation mediaType="application/xml" type="tns1:
RA_INTERFACE_LINES_ALL_Output"/>
    <representation mediaType="application/json" type="tns1:
RA_INTERFACE_LINES_ALL_Output" />
    <representation mediaType="text/csv" type="tns1:
RA_INTERFACE_LINES_ALL_Output"/>
   </response>
 </method>
 <method id="RA_INTERFACE_LINES_ALL_delete" name="DELETE">
     <request>
    <param name="ctx_responsibility" type="xsd:string" style="query"
required="false"/>
           <param name="ctx_respapplication" type="xsd:string" style="
query" required="false" />
    <param name="ctx_securitygroup" type="xsd:string" style="query"
required="false" />
    <param name="ctx_nlslanguage" type="xsd:string" style="query"
required="false" />
    <param name="ctx_orgid" type="xsd:int" style="query" required="
false" />
    <param name="filter" type="xsd:string" style="query" required="
false"/>
   </request>
   <response>
    <representation mediaType="application/xml" type="tns1:
RA_INTERFACE_LINES_ALL_Output"/>
    <representation mediaType="application/json" type="tns1:
RA_INTERFACE_LINES_ALL_Output" />
    <representation mediaType="text/csv" type="tns1:
RA_INTERFACE_LINES_ALL_Output"/></response>
 </method>
 </resource><resource path="SUBMIT_CP_RAXMTR/">
 <method id="SUBMIT_CP_RAXMTR_post" name="POST">
      <request>
    <representation mediaType="application/xml" type="tns1:
SUBMIT_CP_RAXMTR_Input"/>
    <representation mediaType="application/json" type="tns1:
SUBMIT_CP_RAXMTR_Input" />
    <representation mediaType="text/csv" type="tns1:
SUBMIT_CP_RAXMTR_Input"/>
   </request>
   <response>
    <representation mediaType="application/xml" type="tns1:
SUBMIT_CP_RAXMTR_Output"/>
    <representation mediaType="application/json" type="tns1:
SUBMIT_CP_RAXMTR_Output" />
    <representation mediaType="text/csv" type="tns1:
SUBMIT_CP_RAXMTR_Output"/>
```

```
</response></resource>
 </resources>
</application>
```

For more information about WADL description, see Reviewing WADL Element Details, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

To deploy a REST web service:

1. Log in to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility and 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, specify service alias information.

   If the selected interface is an interface type of Business Service Objects, Java Bean Services, Application Module Services, Open Interface Tables, or Open Interface Views, select the desired HTTP method check boxes for the desired methods to be exposed as REST service operations.

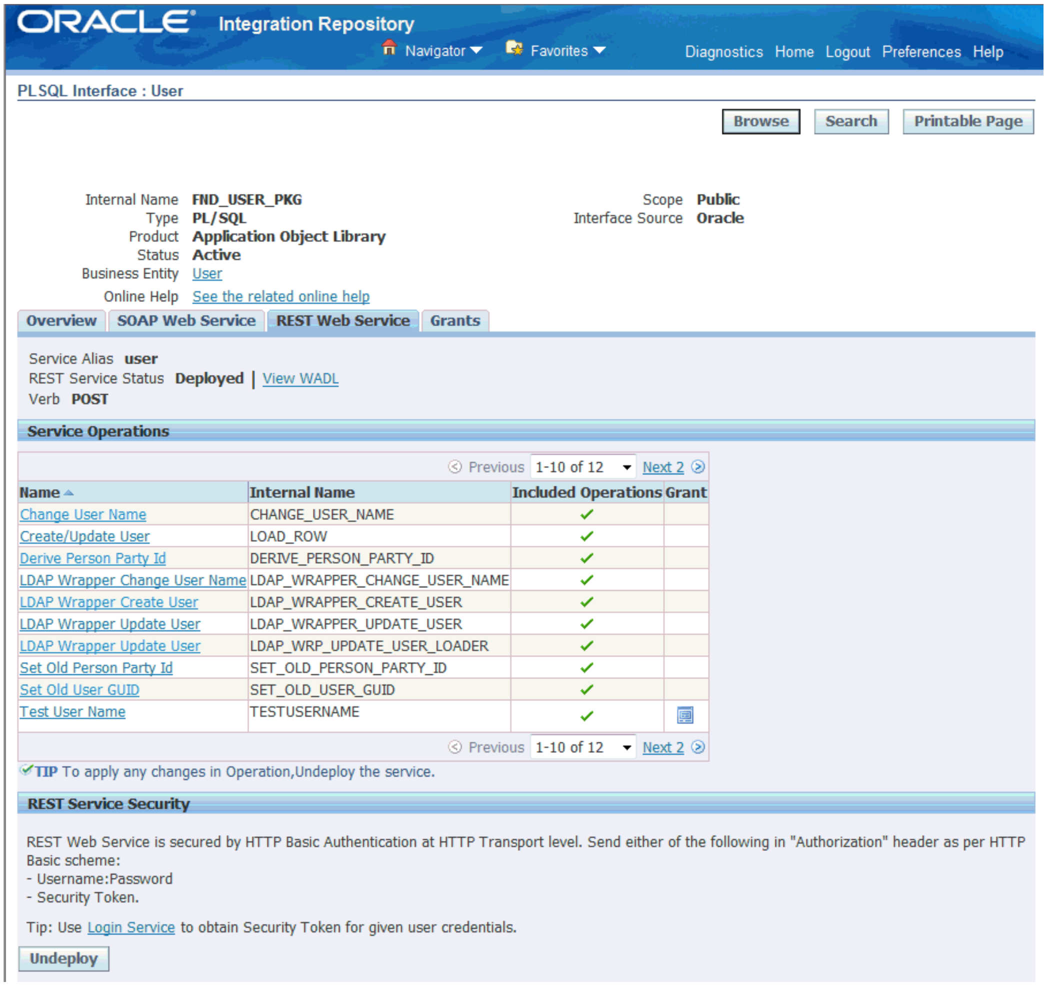
6. Click **Deploy** to deploy the service to an Oracle E-Business Suite environment.

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".

Interface Details Page with a Deployed REST Service



PLSQL Interface : User

Internal Name	**FND_USER_PKG**	Scope	**Public**
Type	**PL/SQL**	Interface Source	**Oracle**
Product	**Application Object Library**		
Status	**Active**		
Business Entity	User		
Online Help	See the related online help		

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

Service Operations

Previous 1-10 of 12 Next 2

Name	Internal Name	Included Operations	Grant
Change User Name	CHANGE_USER_NAME	✓	
Create/Update User	LOAD_ROW	✓	
Derive Person Party Id	DERIVE_PERSON_PARTY_ID	✓	
LDAP Wrapper Change User Name	LDAP_WRAPPER_CHANGE_USER_NAME	✓	
LDAP Wrapper Create User	LDAP_WRAPPER_CREATE_USER	✓	
LDAP Wrapper Update User	LDAP_WRAPPER_UPDATE_USER	✓	
LDAP Wrapper Update User	LDAP_WRP_UPDATE_USER_LOADER	✓	
Set Old Person Party Id	SET_OLD_PERSON_PARTY_ID	✓	
Set Old User GUID	SET_OLD_USER_GUID	✓	
Test User Name	TESTUSERNAME	✓	▤

Previous 1-10 of 12 Next 2

✓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.

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.

To undeploy a REST service:

1. Log in to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility and 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 SOAP and REST Web Services

Users who have 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.

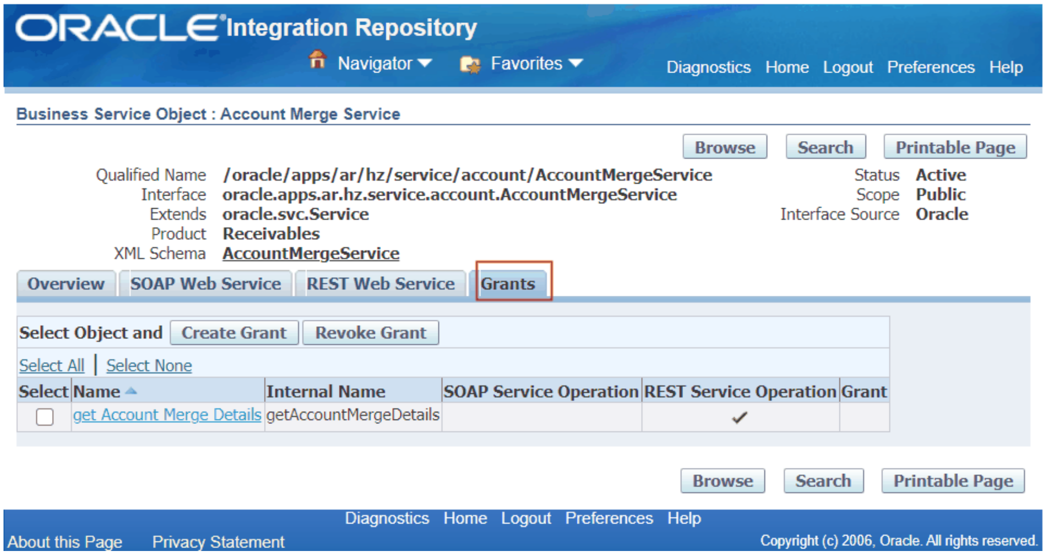
> **Note:** In this release, only PL/SQL APIs, Concurrent Programs, and Business Service Objects can be exposed as both SOAP and REST services. Java Bean Services, Application Module Services, Open Interface Tables, and Open Interface Views can be exposed as REST services only.

Managing Grants in the Grants Tab for PL/SQL APIs, Concurrent Programs, Business Service Objects, Java Bean Services, Application Module Services, Open Interface Tables, and Open Interface Views

With the exception of XML Gateway interfaces that the user security is managed in the XML Gateway user interface, security grants for all other interface types that can be exposed as web services are managed in the Grants tab of the interface details page. These interfaces are PL/SQL APIs, Concurrent Programs, Business Service Objects, Java Bean Services, Application Module Services, Open Interface Tables, and Open Interface Views.

For information on managing the user security for XML Gateway interfaces, see: Managing Security Grants for SOAP Web Services Only, page 3-15.

Interface Details Page with Grants Tab Highlighted



Business Service Object : Account Merge Service

Qualified Name	/oracle/apps/ar/hz/service/account/AccountMergeService	Status	Active
Interface	oracle.apps.ar.hz.service.account.AccountMergeService	Scope	Public
Extends	oracle.svc.Service	Interface Source	Oracle
Product	Receivables		
XML Schema	AccountMergeService		

Select Object and

Select	Name	Internal Name	SOAP Service Operation	REST Service Operation	Grant
□	get Account Merge Details	getAccountMergeDetails		✓	

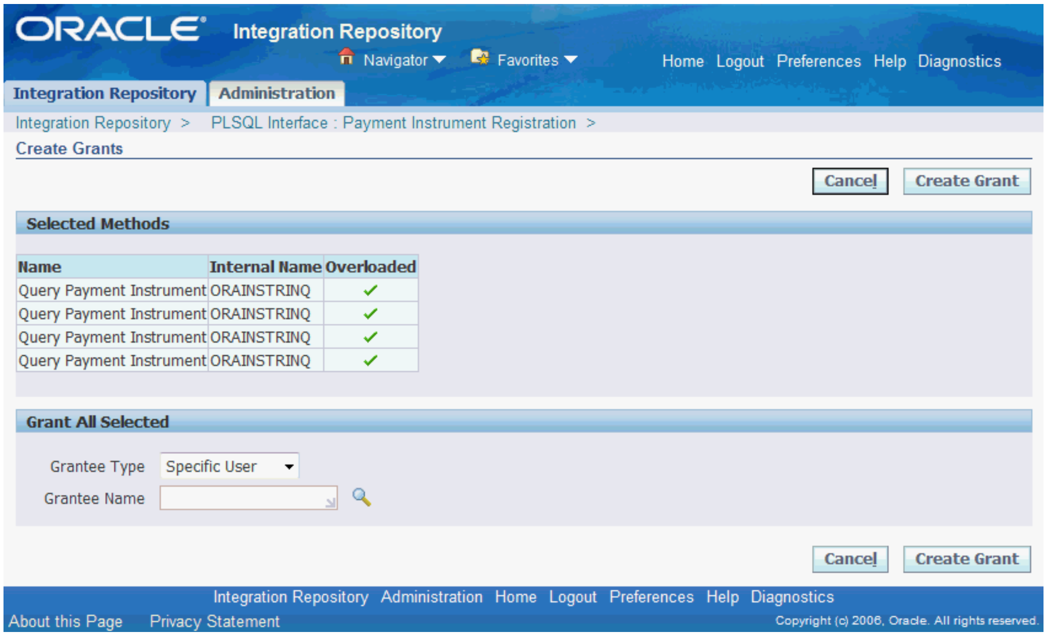
Copyright (c) 2006, Oracle. All rights reserved.

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.

Create Grants Page with Overloaded Functions



ORACLE Integration Repository
Navigator
Favorites
Home Logout Preferences Help Diagnostics
Integration Repository
Administration
Cancel
Create Grant
Cancel
Create Grant
Integration Repository Administration Home Logout Preferences Help Diagnostics
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

Integration Repository > PLSQL Interface : Payment Instrument Registration >

Create Grants

Selected Methods

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

Grant All Selected

Grantee Type: Specific User

Grantee Name:

- Only users who have 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, an Overloaded column appears in the Selected Methods table with the selected overloaded functions checked.

Revoking Security Grants

The administrator can revoke security grants in the following ways:

> **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.

- *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
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 called Grants appears with the grant details.

> **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`.

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

To create grants:

1. Log in to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility and 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 that can be exposed as a REST service or 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 interface that can be exposed as a REST service.

2. To view grant details:

   In the Grants tab, the REST Web Service tab, or the SOAP Web Service tab if it appears, 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 Service Life Cycle and Security Grants Through Backend Processing

In addition to managing service lifecycle activities and creating security grants through the Integration Repository user interface, administrators can perform these tasks through backend processing:

- Managing SOAP Service Lifecycle Activities Using Manual Steps, page 3-43
- Managing REST Service Lifecycle Activities Using Manual Steps, page 3-45
- Managing Security Grants Using an Ant Script, page 3-47
- Generating a Service Description List Using an Ant Script, page 3-49

Managing SOAP Service Lifecycle Activities Using Manual Steps

Oracle E-Business Suite Integrated SOA Gateway allows you to perform SOAP service design-time activities through the use of command line. This includes:

- Generating SOAP Services Using `soagenerate.sh`, page 3-44
- Redeploying SOAP Services in a Cloned Environment Using `Postclone.sh`, page 3-45

For information on setting up Oracle E-Business Suite Integrated SOA Gateway in a multinode environment, see *Configuring Oracle E-Business Suite Integrated SOA Gateway Release 12.1.2 and Release 12.1.3 in a Multinode Environment*, My Oracle Support Knowledge Document 1081100.1.

For more troubleshooting information, see the *Oracle E-Business Suite Integrated SOA Gateway Troubleshooting Guide, Release 12*, My Oracle Support Knowledge Document 726414.1.

Generating SOAP Services Using `soagenerate.sh`

When you try to generate a SOAP service from the Integration Repository user interface, if the system takes too long for the service generation to complete, the following HTTP 403 exception may appear on the interface:

```
oracle.apps.fnd.soa.util.SOAException:SystemError:Error while sending
message to server.

Server returned HTTP response code: 403 for URL: http://<hostname>:
<port>/webservices/SOAProvider/EbizAuth?
Generate=1656&soa_ticket=xxxxxxxxxxxxxxxxx_xxxx..' when attempting to
perform 'GENERATE'.
```

To resolve the issue, a standalone `soagenerate.sh` script is created allowing you to generate WSDL services for PL/SQL, concurrent program, and XML Gateway Map interfaces through backend processing.

Prerequisites to Run soagenerate.sh:

1. Environment variable (like `$IAS_ORACLE_HOME`) needs to be set by running `.env` file in APPL_TOP of your environment.

2. If you have the `appmgr` privileges and have the **read** permission on `$INST_TOP/ora/10.1.3/j2ee/oafm/config/oc4j.properties`, then you can run `soagenerate.sh` without any setup described in step 3.

3. If you do not have the **read** permission on `$INST_TOP/ora/10.1.3 /j2ee/oafm/config/oc4j.properties`, then you need to set the following properties present in `JAVA_TOP/oracle/apps/fnd/soa/provider/wsdl/data/soa.properties` :

   1. Set the following two database connection related properties in the `soa. properties` file:

      1. `SOA_CREATE_DB_CONN_CONTEXT = true`

      2. `JTFDBCFILE =`<Physical Location of dbc file, to which User has read access>

   2. Set the other required properties:

      1. `SOA_SERVER_TEMP_DIRECTORY_LOCATION=`<location of $INST_TOP/soa>

      2. `SOA_SERVER_URL=<protocol://host:port of Apps Self Service URL>`

      3. `SOA_ENABLE_STANDALONE_LOGGING = true`

4. You should have the **write** permission on

SOA_SERVER_TEMP_DIRECTORY_LOCATION mentioned in step 3.

5. You should have the **write** permission on the directory from where you are running this script.

Usage of soagenerate.sh:

```
$FND_TOP/bin/soagenerate [help] irepname=<irepname>
logfile=<logfile> printprops=<true|false>
```

Valid arguments for soagenerate are described as follows:

- **irepname**: (mandatory) irepname of the interface to be generated.

- **logfile**: (optional) logfilename, if a log file is to be created in any other directory.

  By default, a log file is created with name 'ServiceGenerator.log' in the directory from which soagenerate.sh is executed. If you want to create a log file in any other directory, give the location of the file in this argument.

- **printprops**: (optional) true|false, whether system properties should be printed in logfile.

  By default, system properties related with SOA are not printed in logfile. If you want to print system properties in logfile, specify printprops=true.

Redeploying SOAP services in a Cloned Environment Using Postclone.sh

You can run the following Postclone.sh script to have deployed SOAP services created in a cloned Oracle E-Business Suite environment:

```
$FND_TOP/bin/postclone.sh
Enter service type (SOAP/REST/BOTH) :
```

Enter "SOAP" as the Service Type value to clone SOAP services. To clone both SOAP and REST services, enter "BOTH" as the value.

This Postclone.sh script writes results to the $INST_TOP/soa/SOAPPostCloneResults.txt file. It includes postclone status and WSDL URL for each deployed interface. If the script fails to redeploy an interface, it is also mentioned in this file.

Managing REST Service Lifecycle Activities Using Manual Steps

In addition to performing REST design-time activities through the Integration Repository user interface (UI), you can use command line to perform the following activities:

- Deploying or Undeploying REST Services Using RestDeployer.sh, page 3-46

- Redeploying REST Services in a Cloned Environment Using Postclone.sh, page 3-47

For information on setting up ISG in a multinode environment, see *Configuring Oracle E-Business Suite Integrated SOA Gateway Release 12.1.2 and Release 12.1.3 in a Multinode Environment*, My Oracle Support Knowledge Document 1081100.1.

Deploying or Undeploying REST Services Using RestDeployer.sh

Once you deploy a REST service through manual steps by running the script `RestDeployer.sh`, a WADL link for the deployed REST service would be available in the Interface Details page through the Integration Repository user interface.

Perform the following steps to deploy or undeploy a REST service:

1. Find `irep_name` of the service to be deployed.

   For example, `irep_name` is the 'Internal Name' of a desired PL/SQL API shown in the Interface Details page in the Integration Repository user interface.

2. Change directory to `$FND_TOP/bin`.

3. Run `RestDeployer.sh irepname=<irep_name>`. For example,

   - Deploy a PL/SQL API 'FND_GLOBAL' as a REST service with POST HTTP verb using the following command:

     `RestDeployer.sh irepname=FND_GLOBAL`.

   - Deploy the following open interface tables contained in the 'OEOIMP' open interface with `Inbound` direction using one command:

     ```
     RestDeployer.sh irepname=OEOIMP[{OE_HEADERS_IFACE_ALL:
     GET+POST+DELETE+PUT}{OE_LINES_IFACE_ALL:GET+POST}]
     ```

     This deploys OE_HEADERS_IFACE_ALL table as a REST service with four supported HTTP verbs (GET, POST, DELETE, and PUT), and OE_LINES_IFACE_ALL table as a REST service with both the GET and POST HTTP verbs.

     Note that the supported verb for PL/SQL APIs is POST only; the supported verbs for Java Bean Services and Application Module Services are all the annotated verbs and POST. For open interface tables with Outbound direction and open interface views, the supported verb is GET only.

   When prompted, provide the following inputs:

   `Enter the target as deploy or undeploy:` deploy (or undeploy)

   `Enter the alias of the interface to be deployed:` <alias name>

4. Review log file `ServiceGenerator.log` for details.

   Follow the text after the "ServiceGenerator invoked at : <Date Time>" message in the log file. The irepName, ClassId, WADL URL and status information would be displayed. For example, the log file contains the following information for a deployed PL/SQL REST service:

```
irepName is : <irep_name>
ClassId = <classId>
Class Type = PLSQL
Generating service with classId = <classId>
WADL URL = https://<host>:<port>/webservices/rest/<alias name>?WADL
Service Generated and Deployed.
```

The message "Service Generated and Deployed" indicates that the REST service is successfully deployed.

Redeploying REST Services in a Cloned Environment Using Postclone.sh

You can execute the following `Postclone.sh` script to have deployed REST services created in a cloned Oracle E-Business Suite environment:

```
$FND_TOP/bin/postclone.sh
Enter service type (SOAP/REST/BOTH) :
```

Enter "REST" as the Service Type value to clone REST services. To clone both SOAP and REST services, enter "BOTH" as the value.

The `Postclone.sh` script writes results to the `$INST_TOP/soa/RESTPostCloneResults.txt` file. It includes postclone status and WADL URL for each deployed interface. If the script fails to redeploy an interface, it is also mentioned in this file.

Managing Security Grants Using an Ant Script

In addition to managing security grants through the Integration Repository user interfaces, an administrator can use the `isggrant.xml` script to perform this task.

```
ant -f $JAVA_TOP/oracle/apps/fnd/isg/isggrant.xml -
DirepNames=<interface_name[function1:function2:..]> -
Dactions=<CREATE | REVOKE> -DgranteeType=< USER | GROUP | GLOBAL>
-DgranteeKey= <Grantee Key>
```

See Using the Script with Arguments for the Grant, page 3-48.

Argument Description

The following arguments are specifically used in `isggrant.xml` for managing security grants:

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

  Use either one of the following syntax for the interface name:

  - `interface_name`
  - `interface_name[function1:function2: ... ]`

- **actions:** Comma separated list of actions to be performed. Supported operations are listed as follows:

  - create: It creates a security grant for a selected interface or service.

  - revoke: It removes a grant created earlier, including the privileges of a grantee of any type (such as user, group, or global) assigned to the grant.

- **granteeType:** Supported values are:

  - USER: It grants the access privilege of a selected interface or service to a specific user only.

  - GROUP: It grants the access privilege of a selected interface or service to a specific group only.

  - GLOBAL: It grants the access privilege of a selected interface or service to all Oracle E-Business Suite users.

- **granteeKey:** A required argument to provide a specific user or group value when granteeType is USER or GROUP. It is not required when the granteeType value is GLOBAL.

  - If granteeType is USER, provide user code (user name) as granteeKey.

  - If granteeType is GROUP, provide responsibility code as granteeKey.

Using the Script with Arguments for the Grant

The following examples explain how to use the script with arguments to create and revoke security grants:

- Create a grant to a specific user "OPERATIONS" with the access privileges of CHANGE_USER_NAME and TESTUSERNAME service operations within the FND_USER_PKG interface:

  > **Note:** `OPERATIONS` in the `DgranteeKey` argument is the user name (user code) value for an Oracle E-Business Suite user.

  ```
  ant -f $JAVA_TOP/oracle/apps/fnd/isg/isggrant.xml -
  DirepNames=FND_USER_PKG[CHANGE_USER_NAME:TESTUSERNAME] -
  Dactions=CREATE -DgranteeType=USER -DgranteeKey=OPERATIONS
  ```

- Revoke the privileges from the group SYSTEM_ADMINISTRATION that has given the access of all service operations in the FND_MESSAGE interface and the CHANGE_USER_NAME operations within the FND_USER_PKG interface:

  > **Note:** `FND_RESP|ICX|SYSTEM_ADMINISTRATION|STANDARD` in the `DgranteeKey` argument is the responsibility code value for the System Administration responsibility.

```
ant -f $JAVA_TOP/oracle/apps/fnd/isg/isggrant.xml -
Dactions=REVOKE -DirepNames=FND_MESSAGE,FND_USER_PKG
[CHANGE_USER_NAME] -DgranteeType=GROUP -
DgranteeKey=FND_RESP|ICX|SYSTEM_ADMINISTRATION|STANDARD
```

Generating a Service Description List Using an Ant Script

To understand SOAP and REST service generation and deployment status, an administrator can run an Ant script `DownloadServicesList.xml` using the following command:

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

This script generates a service descriptor file called `ISGServiceDescriptor_<timestamp>.xml` in `$INST_TOP/soa`. This file reports the list of APIs which are generated or deployed as SOAP services, or are deployed as REST services.

For example, the following sample service descriptor file lists the service description information:

- A Business Service Object service operation `findParties` contained in the interface `/oracle/apps/ar/hz/service/party/DqmSearchService` (with alias name `dqm`) is deployed as a REST service operation with the POST method.

- A PL/SQL API `HZ_CUST_ACCOUNT_V2PUB` is generated and deployed as a SOAP service with the Username Token security.

```
<INTEGRATION_REPOSITORY name="EBS_SID" xsd_version="2.0">
  <INTERFACE>
    <NAME>/oracle/apps/ar/hz/service/party/DqmSearchService</NAME>
    <TYPE>SERVICEBEAN</TYPE>
    <REST_ACTIONS>
      <UNDEPLOY/>
      <DEPLOY>
        <ALIAS>dqm</ALIAS>
        <FUNCTIONS_LIST verb="POST">
          <FUNCTION>findParties</FUNCTION>
        </FUNCTIONS_LIST>
      </DEPLOY>
    </REST_ACTIONS>
  </INTERFACE>
  <INTERFACE>
    <NAME>HZ_CUST_ACCOUNT_V2PUB</NAME>
    <TYPE>PLSQL</TYPE>
    <SOAP_ACTIONS>
      <UNDEPLOY/>
      <GENERATE>
        <ALL_FUNCTIONS/>
      </GENERATE>
      <DEPLOY>
        <POLICY>USERNAME_TOKEN</POLICY>
      </DEPLOY>
    </SOAP_ACTIONS>
  </INTERFACE>
</INTEGRATION_REPOSITORY>
```

4

Administering Composite Services

This chapter covers the following topics:

- Overview
- Understanding Composite Service Enablement Process
- Administering Composite Services

Overview

Composite services use the native service as building blocks to construct the sequence of business flows. Basically, this interface type orchestrates the invocation sequence of discrete Web services into a meaningful end-to-end business process through a Web service composition language BPEL (business process execution language). For example, use Oracle BPEL Process Manager (BPEL PM) to integrate the Order-to-Receipt business process that contains sales order entry, item availability check, pack and ship, and invoice to Accounts Receivable sub processes handled by various applications. This approach effectively tightens up the control of each individual process and makes the entire business flow more efficiently.

This chapter includes the following topics:

- Understanding Composite Services Enablement Process, page 4-1
- Administering Composite Services, page 4-3

Understanding Composite Service Enablement Process

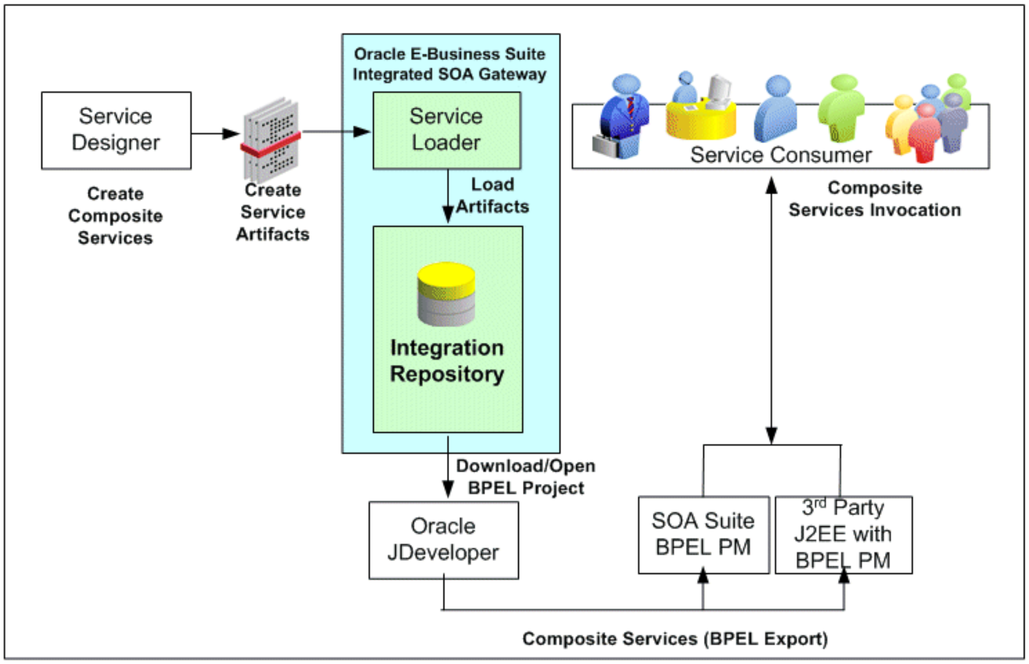
Composite services use native services as building blocks to orchestrate the business invocation sequence from discreate Web services into a meaningful end-to-end business flow through a Web service composition language BPEL. Strictly speaking, this type of interface is comparatively service enabled without additional service generation process as required by native interface types.

To design a composite service, integration repository developers use the BPEL language to specify the invocation sequence through Oracle JDeveloper. This composite service has its own WSDL definition and endpoint through the creation of a partner link which allows a business event, for example, to be published to the Oracle BPEL Process Manager or to interact with a partner service.

To make composite services available over the Internet for service consumers or Web service clients to use the services, Oracle E-Business Suite Integrated SOA Gateway uses various service components to host composite services. The relationship between each component is illustrated in the following composite service enablement diagram:

> **Note:** Integration repository developers use Service Designer (Oracle JDeveloper) to create composite services by orchestrating the invocation sequence of discrete Web services through Web service composition language BPEL. See the *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide* for details.

Composite Service Enablement Process


Service Designer
Create Composite Services
Create Service Artifacts
Oracle E-Business Suite Integrated SOA Gateway
Service Loader
Load Artifacts
Integration Repository
Service Consumer
Composite Services Invocation
Download/Open BPEL Project
Oracle JDeveloper
SOA Suite BPEL PM
3rd Party J2EE with BPEL PM
Composite Services (BPEL Export)

Integration repository developers create composite services using Oracle JDeveloper. Service loader then uploads these service artifacts to Oracle Integration Repository. Users granted with the Download Composite Service privilege (FND_REP_DOWNLOAD_CS) can further download the BPEL files to their local machines. The developers can open the downloaded BPEL files using Oracle JDeveloper, modify them if necessary, and deploy them. Oracle BPEL Process Manager (BPEL PM) or 3rd party J2EE BPEL PM will then pick up those deployed composite services.

> **Note:** Unlike native services that they are deployed through the Web Service region of an interface type detail page, composite services are typically not deployed within Oracle E-Business Suite like those of other service enabled interface types. You need a separate BPEL PM (SOA Suite or third party BPEL PM server) to deploy the BPEL composite services. For example, a composite service - BPEL type can be deployed through Oracle JDeveloper to a BPEL server in Oracle SOA Suite BPEL PM (Process Manager) or a third party BPEL PM in a J2EE environment. This deployed composite service - BPEL project can update Oracle E-Business Suite if necessary.

Administering Composite Services

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

- Viewing a Composite Service, page 4-3

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

- Downloading a Composite Service, page 4-4

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

Viewing Composite Services

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 a composite service details for a selected composite service. From the composite service interface details page, the administrators can find composite service name, description, BPEL file, and other annotated information.

To locate a composite service, navigate to the Composite Service interface type directly from the Oracle Integration Repository Browser window or perform a search by selecting Composite Service interface type in the Search page. Click a desired composite service name link from the browser tree or the search result to display the composite service - BPEL interface details page where the administrators can:

- View the composite service - BPEL details.

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

- Download a corresponding composite service - BPEL project file from Oracle Application Server to their local directories.

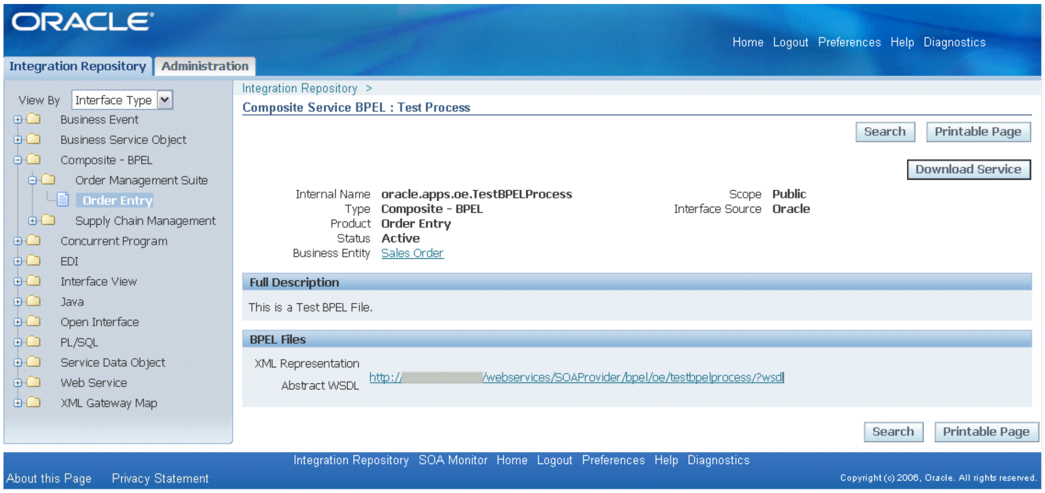
  For more information on how to download a composite service, see Downloading Composite Services, page 4-4.

Downloading Composite Services

In addition to viewing composite service details and a WSDL file, the administrators can download a BPEL .JAR file containing relevant composite service files to their local directories by clicking **Download Service** in the composite service - BPEL details page.

> **Important:** In general, only system integration developers and integration repository administrators can download the composite services. However, general users who are granted the Download Composite Service privilege, a permission set FND_REP_DOWNLOAD_PERM_SET, can also perform the download action. Otherwise, general users (or system integration analysts) will not find **Download Service** available in the BPEL details page.
>
> For more information about how to grant Download Composite Service privilege, see Role-Based Access Control (RBAC) Security, page 6-3.

Composite Details Page with Download Privilege



Composite Service BPEL : Test Process

Internal Name **oracle.apps.oe.TestBPELProcess**
Type **Composite - BPEL**
Product **Order Entry**
Status **Active**
Business Entity Sales Order

Scope **Public**
Interface Source **Oracle**

Full Description

This is a Test BPEL File.

BPEL Files

XML Representation
Abstract WSDL http://[redacted]/webservices/SOAProvider/bpel/oe/testbpelprocess/?wsdl

> **Note:** A system integration developer can further unzip the BPEL .JAR file and open the BPEL file in Oracle JDeveloper for further

modification on service endpoints if needed.

Additionally, the developer can deploy the BPEL process if necessary. Since composite services are typically not deployed within Oracle E-Business Suite, a separate BPEL PM (SOA Suite or a third party BPEL PM server) is needed to deploy the BPEL composite services. For example, deploy the BPEL process to Oracle BPEL server through Oracle BPEL Process Manager. See the *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

To download a composite service:

1. Log in to Oracle Integration Repository as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility and the Integration Repository link.

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

3. Expand the Composite Service interface type node to locate a desired composite service.

4. Click the composite service that you want to download. The Composite Service Interface Details page appears.

5. Click **Download Service** to download the selected composite 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, Java Bean Services, Application Module Services, 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. With appropriate validation, if no error occurred, the validated custom definition sources compiled in a generated iLDT file can be uploaded to Oracle Integration Repository through backend processing.

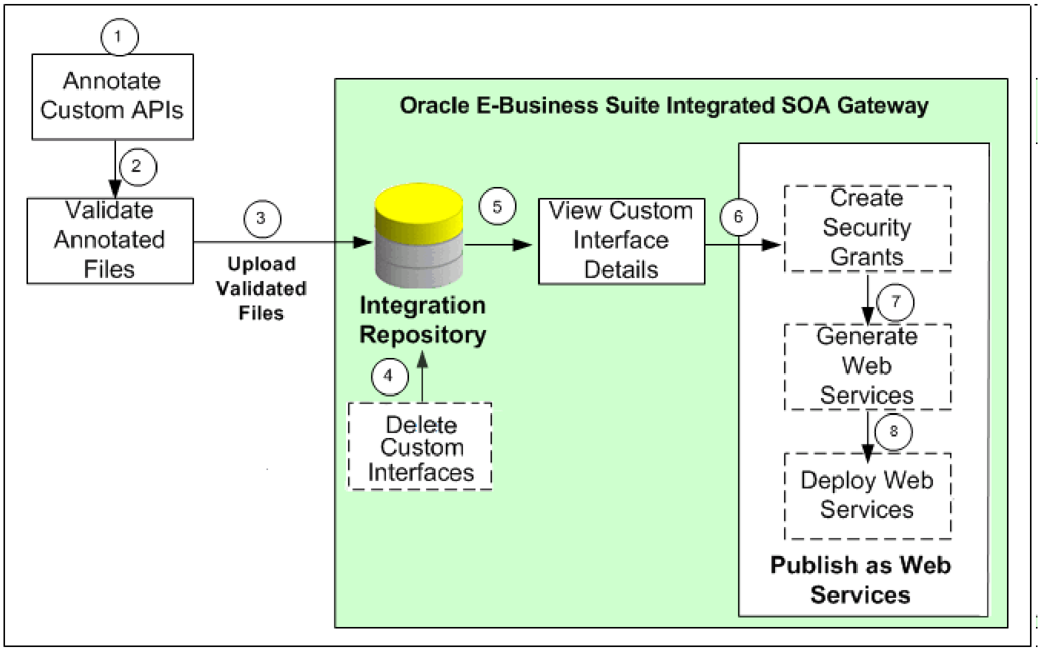
> **Note:** Custom interface types of EDI, Open Interface Tables, and Open Interface Views are not supported in this release.
>
> Oracle Integration Repository currently does not support the creation of custom Product Family and custom Business Entity.

After the upload, these custom integration interfaces are displayed together with Oracle seeded ones through the Integration Repository user interface based on the interface type 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 interfaces. Custom integration interfaces of service enabled interface types can be exposed as web services. 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.

Enabling Custom Integration Interface Process Flow

The entire process flow described here can be illustrated in the following diagram:

Custom Integration Interfaces Development Process Flow



1. Users who have 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 who have 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 5-4

   - Generating ILDT Files, page 5-8

3. Users who have the Integration Repository Administrator role upload the generated iLDT file to Oracle Integration Repository.

See: Uploading ILDT Files to Integration Repository, page 5-12.

4. (Optional) Users who have the Integration Repository Administrator role can delete the custom integration interfaces if needed.

   Before starting to use a custom integration interface from the Integration Repository, users who have the Integration Repository Administrator role can delete the custom interface if it is not yet deployed as a web service. The administrators can first locate the custom interface from the Integration Repository user interface, and then click **Delete Interface** in the Overview tab of the custom interface details page.

   If a custom interface has been deployed, it must be undeployed first before it can be deleted. That is, its web service status must be either 'Generated' for a custom SOAP service or 'Not Deployed' for a custom REST service. See: Deleting Custom Integration Interfaces, page 5-16.

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

6. (Optional) Users who have 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.

7. (Optional) Users who have the Integration Repository Administrator role can generate SOAP services if the custom interfaces can be service enabled.

   This is achieved by first locating the custom interface, and then clicking **Generate** (or **Generate WSDL**) in the selected custom interface details page. See: Generating Custom SOAP Web Services, page 5-17.

8. (Optional) Users who have the Integration Repository Administrator role deploy the services from Oracle Integration Repository to the application server.

   To deploy generated SOAP services, the administrators must first select one authentication type (Username Token or SAML Token) for each selected service and then click **Deploy** in the selected interface details page. This deploys the generated service to the application server. See: Deploying and Undeploying SOAP Custom Web Services, page 5-18.

   If the custom interfaces can be exposed as REST services, the administrators must enter a unique service alias for each selected custom interface before deploying the service. Additionally, the administrators need to specify HTTP methods for desired service operations contained in the selected interface if it is an interface type of Java Bean Services, Application Module Services, or Business Service Object.

> **Note:** Although Open Interface Tables and Open Interface Views can be exposed as REST services, custom Open Interfaces are not supported in this release.

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 5-18.

To better understand how to use Integration Repository Parser to validate and upload annotated custom interface definitions to Integration Repository and 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 5-4

- Administering Custom Integration Interfaces and Services, page 5-13

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. An integration repository administrator uses it to validate and generate the annotated custom interface definitions against the annotation standards. It can read almost all types of application source files. While executing the parser, the annotated source files are validated based on the interface type supported for customization. If no error occurs, an Integration Repository loader file (iLDT) will be created.

> **Note:** Please note that Integration Repository Parser does not support the integration interfaces registered under custom applications.
>
> It is currently tested and certified for Linux, Unix, Oracle Solaris on SPARC, HP-UX Itanium, HP-UX PA-RISC, IBM AIX on Power Systems and Windows.

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/x86-64: Intel C/C++ Compiler (icc) version 7.1.032
- Oracle Solaris on SPARC (64-bit): Oracle Studio 12
- Microsoft Windows Server (32-bit): Microsoft Visual Studio 2005 (VC 8.0)
- HP-UX Itanium: HP ANSI C B3910B A.0.06.05
- HP-UX PA-RISC (64-bit): HP92453-01 B.11.11.10 HP C Compiler
- IBM AIX on Power Systems (64-bit): XL C Enterprise 8.0

- ***Prerequisites for Installing Perl Modules on Windows***

  It is necessary to create a manifest file for `perl.exe` in the `10.1.3Home\perl\5.8.3\bin\MSWin32-x86-multi-thread` directory if your installation is on Windows.

  To create a manifest file for `perl.exe`:

  1. Log on to the Oracle E-Business Suite application tier server.
  2. Change directories to `c:\WINDOWS\WinSxS`.
  3. Verify if there is a file that starts with `x86_Microsoft.VC80.CRT`. For example, `x86_Microsoft.VC8.CRT_xxxxxxxxxxxxxxxx_8.0.50727.42_x-ww_0de06acd`.
  4. Record this filename.
  5. Change directories to where the `perl.exe` resides in the 10.1.3 Home.

     For example,

     ```
     cd e:\PROD\apps\tech_st\10.1.3\perl\5.8.3\bin\MSWin32-x86-
     multi-thread
     ```

  6. Open a file with text editor (such as Notepad) to create a manifest file.
  7. Enter the following statements, for example, with the 'version' and 'publicKeyToken' taken from the `x86_Microsoft.VC80.CRT` file name:

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
 <assembly xmlns='urn:schemas-microsoft-com:asm.v1'
manifestVersion='1.0'>
  <dependency>
   <dependentAssembly>
     <assemblyIdentity type='win32' name='Microsoft.VC80.CRT'
      version='8.0.50727.42' processorArchitecture='x86'
      publicKeyToken='xxxxxxxxxxxxxxxx'/>
   </dependentAssembly>
  </dependency>
 </assembly>
```

8. Save the file with the name `perl.exe.manifest`.

To install Perl modules:

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

   From the Oracle E-Business Suite application instance `APPL_TOP`, set the environment by running the `APPS<CONTEXT_NAME>APPS.env(.cmd)` script.

2. Set 10.1.3 `ORACLE_HOME`:

   Navigate to the `<INST_TOP>/ora/10.1.3` and source the `.env/.cmd` file to set your 10.1.3 `ORACLE_HOME`.

3. Add directory `$FND_TOP/perl` to environment variable `PERL5LIB`:

   1. Find physical path of `$FND_TOP/perl`.

   2. Add this physical path in `PERL5LIB` variable.

   3. Example: export `PERL5LIB=/slot/ems3404/appmgr/apps/apps_st/appl/fnd/12.0.0/perl:/slot/ems3404/appmgr/apps/tech_st/10.1.3/perl/lib/5.8.3:/slot/ems3404/appmgr/apps/tech_st/10.1.3/perl/lib/site_perl/5.8.3:/slot/ems3404/appmgr/apps/apps_st/appl/au/12.0.0/perl:/slot/ems3404/appmgr/apps/tech_st/10.1.3/Apache/Apache/mod_perl/lib/site_perl/5.8.3/i686-linux-thread-multi`.

4. Use the following steps for installation on different platforms:

   - **On Unix**

     - Find the value of `$IAS_ORACLE_HOME/perl` in your environment, for example `/slot/ems1340/appmgr/apps/tech_st/10.1.3/perl`.

     - Locate the `$IAS_ORACLE_HOME/perl/lib/5.8.3/i686-linux-thread-multi/Config.pm`.

     - Take backup of this file. Replace all occurrences of

`/ade/smayer_perl58_main_linux/perl58/bin/Linux/Opt` with value of `$IAS_ORACLE_HOME/perl`. For example, `/slot/ems1340/appmgr/apps/tech_st/10.1.3/perl`.

- **On Windows**
  - Search for all `Config.pm` files underneath `%IAS_ORACLE_HOME%\perl`, and record their location, such as:

    ```
    %IAS_ORACLE_HOME%\perl\5.8.3\bin\Config.pm%
    IAS_ORACLE_HOME%\perl\5.8.3\lib\MSWin32-x86-multi-
    thread\Config.pm
    ```

  - For each `Cofing.pm` file, modify all parameters that point to `perl` with the correct location of `%IAS_ORACLE_HOME%\perl`. For example, in the `%IAS_ORACLE_HOME%\perl\5.8.3\bin\Config.pm` file, modify `archlibexp` from `'%ORACLE_HOME%\perl\5.8.3\lib\MSWin32-x86-multi-thread` to `e:\PROD\apps\tech_st\10.1.3\perl\5.8.3\lib\MSWin32-x86-multi-thread`.

  - For each `Cofing.pm` file, modify all parameters that point to `Visual C++` with the correct location of `Visual C++`.

    The location of `Visual C++` is identified through the `msdevdir` parameter in the context file at `%INST_TOP%\apps\admin\<CONTEXT_NAME>.xml`.

    For example, in the `%IAS_ORACLE_HOME%\perl\5.8.3\lib\MSWin32-x86-multi-thread\Config.pm` file, modify `libpth` to the correct location of `Visual C++`:

    `libpth=d:\VC8\VC\lib` (`d:\VC8\VC` is an example).

5. 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:

   1. Change to the directory where you want the `Perl` modules to be installed using the following command:

      `cd <Perl module name>`

      For example: `cd Compress-Raw-Zlib-2.009`

2. Install the modules using the following commands:
   - **On Unix**

     ```
     perl Makefile.PL
     make
     make install
     ```

     Note: Ignore any warning in `make` command.

   - **On Windows**

     ```
     perl Makefile.PL
     nmake
     nmake install
     ```

Using the Integration Repository Parser

Once you have the Integration Repository Parser 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 already 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 that is already 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, the Header version of the target source file needs to be incremented so that the modifications to the object defined in the source file can take effect in the Integration Repository.

How to execute the parser to validate the files and upload them are further discussed in this section:

- Generating ILDT Files, page 5-8
- Uploading ILDT Files to Integration Repository, page 5-12

Generating ILDT Files

Prerequisites - Setting Up Environment Variables

Before executing the Integration Repository Parser to generate iLDT files, set the following environment variables which may affect parser operation:

- 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 the 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 other platforms, check your platform documentation on how to set the classpath variable.

- JAVA_HOME: It is used to find the Java runtime if it is set. Otherwise, /local/java/jdk1.5.0 is used instead (For example, the application session server setup). Typical location of Java in Oracle E-Business Suite Release 12 environment is $IAS_ORACLE_HOME/appsutil/jdk.

Executing the Integration Repository Parser

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>
```

For example:

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

> **Note:** If an error message "Java runtime not found" appears while executing the Integration Repository Parser, then set the environment variable JAVA_HOME to variable OA_JRE_TOP which is typically located at $IAS_ORACLE_HOME/appsutil/jdk/jre.

While executing the parser, pay attention to any error messages on the console. Typically 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 5-12.

Integration Repository Parser (irep_parser.pl) Usage Details

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 supported interface types, if files that do not match will be ignored.

Here is the current 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, and Composite Service for BPEL type.
>
> Custom interface types of EDI, Open Interface Tables, and Open Interface Views interfaces are not supported in this release.

- `.java`: All Java files are completely parsed.
- `.p(kh/ls)`: PL/SQL package specs 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 so generic.
- `.wfx`: It processes the Business Event file, looking for annotated events.

Files Specifications

Argument `filespec` tokens can 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 5-11 (This is backwards 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 varbatim 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 generate iLDT (Interface Repository Seed Data) files if possible. 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 5-11.

- `-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 primary-level annotation), then an empty file is created anyway. If a file of the same name existed from a previous run, it is clobbered to be 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. It is equivalent to using the both `-generate` and the `-verbose` flags. It also supplies a default `prod:path` (of `nul:relative/path/unknown`) to all plain-file filespecs that marks the resulting iLDT as a test file.

This allows 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`: It designates a FND username (other than the default SEED username).
- `-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, etc. This is in addition to any error messages generated.

  It is useful for querying the parser version, if 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
  ```

Uploading ILDT Files to Integration Repository

While executing the Integration Repository Parser to validate the annotated custom interface definitions against the annotation standards and generate iLDT file, if no error occurs during the iLDT generation, an integration repository administrator can upload the generated iLDT file to the Integration Repository where they can be 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. Use Telnet to have command access to the Oracle E-Business Suite Release 12 instance.

2. 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>
   ```

   For example, `FND_TOP/bin/FNDLOAD apps @instance_name 0 Y UPLOAD $FND_TOP/patch/115/import/wfirep.lct SOAIS_pls.ildt`

   `ORACLE Password:` *`password`*

3. Pay attention to any error messages in the generated log file. Typically the error messages 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.

4. For Business Service Object only: Submit a concurrent request for program `FNDIRLOAD`.

   Examine the request log file to see if any issues occur while executing the concurrent request.

Once these annotated source files are successfully uploaded, they will appear in the Integration Repository user interface based on the interface type together with Oracle seeded integration interfaces. The administrators can perform administrative tasks on these custom integration interfaces including generate, deploy, or redploy Web services.

Administering Custom Integration Interfaces and Services

After being uploaded to the Integration Repository, custom integration interfaces will be embedded into appropriate interface categories where the interfaces belong but with 'Custom' interface source in contrast to Oracle seeded ones with interface source 'Oracle'.

Since custom integration interfaces are annotated based on Integration Repository annotation standards for supported interface types, the behavior of these interfaces is really the same as Oracle seeded integration 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 integration interfaces and services.

Viewing Uploaded Custom Integration Interfaces From the Integration Repository

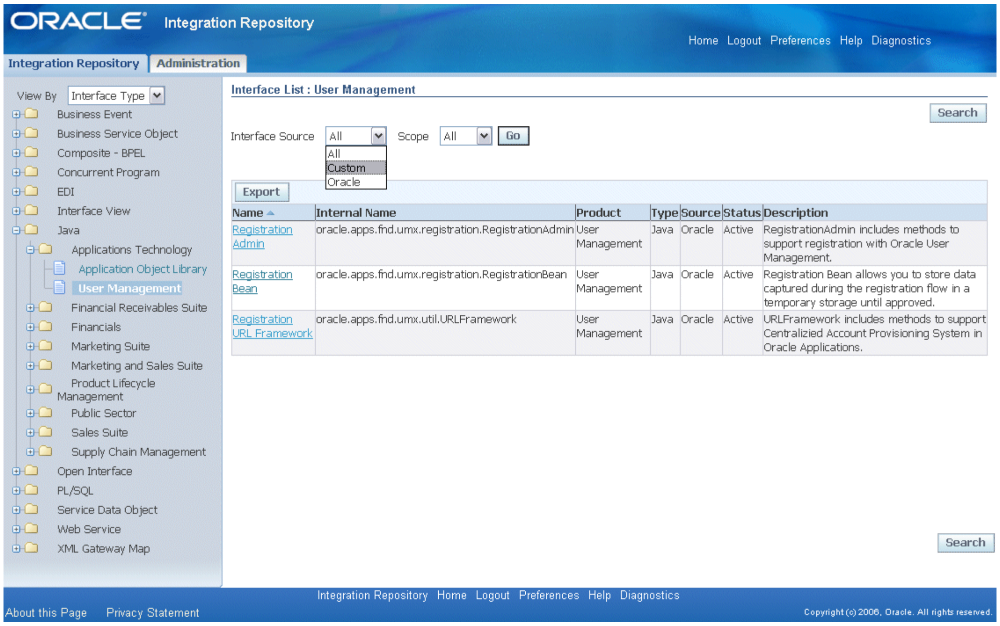
Before performing administrative tasks, you must first locate a custom integration interface from the Integration Repository and then access the interface details page.

> **Note:** The custom interface details page shows 'Custom' as the Interface Source value, while the source value of 'Oracle' is for native packaged integration interfaces.

You can find a custom interface in the following ways:

- 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.

Interface List Page with Interface Source "Custom" Selected



Interface List : User Management

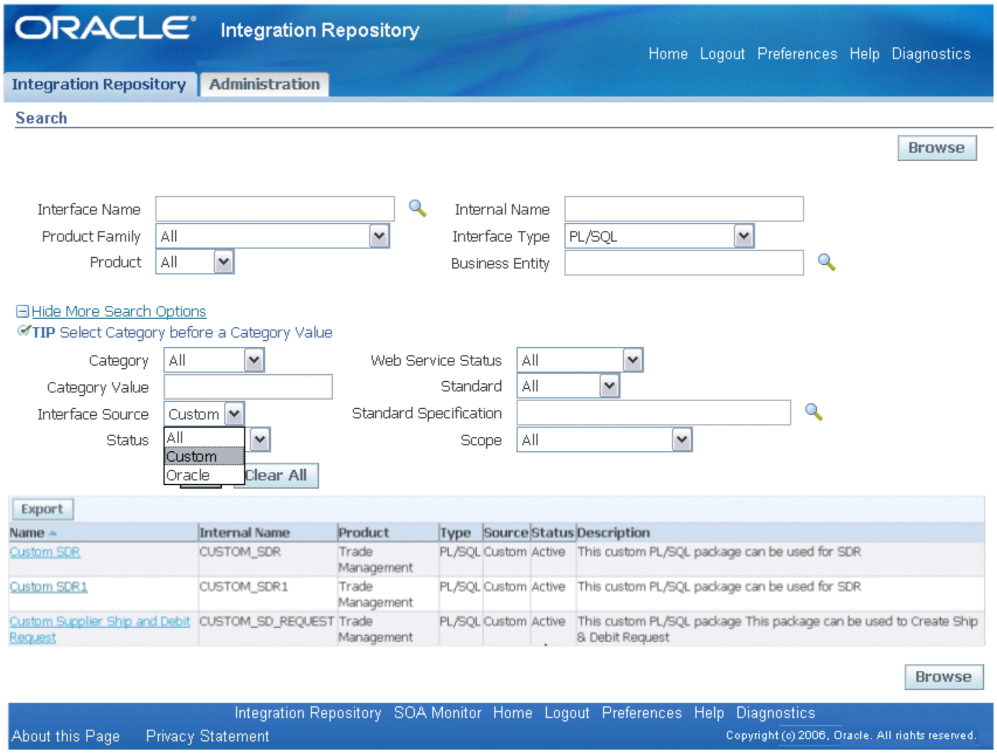
Interface Source: All Scope: All

Name	Internal Name	Product	Type	Source	Status	Description
Registration Admin	oracle.apps.fnd.umx.registration.RegistrationAdmin	User Management	Java	Oracle	Active	RegistrationAdmin includes methods to support registration with Oracle User Management.
Registration Bean	oracle.apps.fnd.umx.registration.RegistrationBean	User Management	Java	Oracle	Active	Registration Bean allows you to store data captured during the registration flow in a temporary storage until approved.
Registration URL Framework	oracle.apps.fnd.umx.util.URLFramework	User Management	Java	Oracle	Active	URLFramework includes methods to support Centralizied Account Provisioning System in Oracle Applications.

- From the Search page, click **Show More Search Options** to 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.

Search Page with Interface Source "Custom" Selected


ORACLE
Integration Repository
Home Logout Preferences Help Diagnostics
Integration Repository
Administration
Search
Browse
Hide More Search Options
TIP Select Category before a Category Value
All
Custom
Oracle
Clear All
Export
Browse
Integration Repository SOA Monitor Home Logout Preferences Help Diagnostics
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

Interface Name		Internal Name	
Product Family	All	Interface Type	PL/SQL
Product	All	Business Entity	

Category	All	Web Service Status	All
Category Value		Standard	All
Interface Source	Custom	Standard Specification	
Status	All	Scope	All

Name	Internal Name	Product	Type	Source	Status	Description
Custom SDR	CUSTOM_SDR	Trade Management	PL/SQL	Custom	Active	This custom PL/SQL package can be used for SDR
Custom SDR1	CUSTOM_SDR1	Trade Management	PL/SQL	Custom	Active	This custom PL/SQL package can be used for SDR
Custom Supplier Ship and Debit Request	CUSTOM_SD_REQUEST	Trade Management	PL/SQL	Custom	Active	This custom PL/SQL package This package can be used to Create Ship & Debit Request

For more information on how to search for custom integration interfaces, see the *Oracle E-Business Suite Integrated SOA Gateway User's Guide*.

After locating a desired custom interface, the administrator can perform the following administrative tasks:

- **Managing Custom Integration Interfaces**
  - Deleting Custom Integration Interfaces, page 5-16
- **Managing Custom Web Service Lifecycle Activities**

  For Custom SOAP Web Services

  - Managing Security Grants for SOAP Services Only, page 5-17
  - Generating Custom SOAP Web Services, page 5-17
  - Deploying, Undeploying, and Redeploying Custom SOAP Web Services, page 5-18
  - Subscribing to Custom Business Events, page 5-18

    This task allows the administrators to subscribe to selected custom business events and create subscriptions for the selected events.

  For Custom REST Web Services

  - Managing Security Grants for Custom REST Web Services, page 5-18
  - Deploying Custom REST Web Services, page 5-18
  - Undeploying Custom REST Web Services, page 5-19

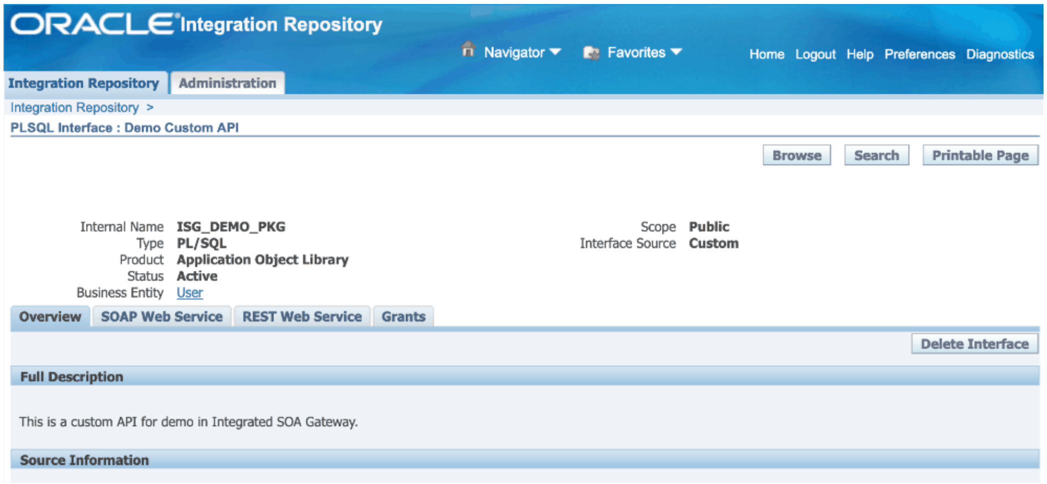
- **For Custom Composite Integration Interface**
  - Viewing and Downloading Custom Composite Services, page 5-19

Deleting Custom Integration Interfaces

Once a custom integration interface is validated and uploaded to the Integration Repository, integration repository administrators can delete the custom interface from the repository if the custom interface is not yet deployed and it is no longer used or needed.

To delete a custom interface, first locate the custom interface from the repository and then click **Delete Interface** in the Overview tab of the selected custom interface details page. This action removes the selected custom interface from the integration repository.

Overview Tab with the "Delete Interface" Button Shown for a Custom Interface



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

Scope **Public**
Interface Source **Custom**

This is a custom API for demo in Integrated SOA Gateway.

If a custom interface has been deployed, it must be undeployed before it can be deleted. That is, you can only delete a custom SOAP service with 'Generated' status or a custom REST service with 'Not Deployed' status from the Overview tab. Otherwise, a warning message appears indicating that you cannot delete a deployed service.

For information on undeploying a SOAP service, see Deploying and Undeploying

SOAP Web Services, page 3-10. For information on undeploying a REST service, see Undeploying REST Web Services, page 3-36.

Managing Security Grants for SOAP Services Only

To let appropriate users use these newly-uploaded custom integration interfaces, the administrators can create security grants, if needed, by authorizing the access permissions for selected interface methods to appropriate users.

In this release, XML Gateway (inbound) is the only interface type that can be exposed as SOAP services only. To manage user security for XML Gateway interfaces, you need to log in to Oracle XML Gateway user interface.

For more information, see Managing Security Grants for SOAP Services Only, page 3-15
.

Generating Custom SOAP Web Services

Once custom integration interfaces are uploaded to Oracle Integration Repository, an integration repository administrator can transform these interface definitions into WSDL descriptions for the interface types with the support for SOAP services.

To generate a WSDL URL, the administrator must first locate the desired custom interface and then generate the SOAP service by clicking **Generate WSDL** in the interface details page.

If the custom SOAP service is successfully generated, a WSDL link becomes available along with the 'Generated' SOAP Service Status in the SOAP Web Service tab (or the Web Service - SOA Provider region for the XML Gateway interface type).

Click the WSDL link to view the WSDL description. See: Reviewing Web Service WSDL Source, *Oracle E-Business Suite Integrated SOA Gateway User's Guide*.

Additionally, the following buttons are available for further actions on the custom SOAP service:

- **Regenerate WSDL**

  This lets you regenerate the WSDL if necessary.

- **Deploy**

  Before deploying a service that is service enabled through SOA Provider, you must select at least one authentication type in the Authentication Type field for the selected service. The selected authentication type(s) will be used by SOA Provider to secure web service content and authenticate web service operation. Clicking **Deploy** in the SOAP Web Service tab (or the Web Service - SOA Provider region for XML Gateway interface type) lets you deploy the generated SOAP service from Oracle Integration Repository to the application server.

  See: Deploying, Undeploying, and Redeploying SOAP Web Services, page 5-18.

For detailed information on how to generate a custom SOAP service, see Generating SOAP Web Services, page 3-5.

Deploying, Undeploying, and Redeploying SOAP Web Services

Once a SOAP service is successfully generated for a custom integration interface, the administrator can deploy the generated SOAP service to the application server, and redeploy or undeploy it again if necessary from the server.

For detailed information on how to deploy, undeploy, or redeploy SOAP services, see Deploying, Undeploying, and Redeploying SOAP Web Services, page 3-10.

Subscribing to Custom Business Events

Similar to the native business events, an integration repository 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 the event subscription has been successfully created, a confirmation message appears in the business event details 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 `WF_BPEL_Q`, then the event message will be enqueued in `WF_EVENT_T` structure to Advanced Queue `WF_BPEL_Q`.

The **Unsubscribe** button becomes available in the details page if the selected event has been subscribed. Clicking the **Unsubscribe** button removes the event subscription. A confirmation message also appears after the subscription has been successfully removed.

Managing Security Grants for Custom REST Services

Security grants for custom REST services are managed in the Grants tab of the selected custom interface details page. 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.

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 SOAP and REST services, see Managing Security Grants for SOAP and REST Web Services, page 3-38.

Deploying Custom REST Web Services

After custom interfaces that can be exposed as REST services are uploaded to the Integration Repository, the administrator can deploy the custom REST services.

Before deploying a custom interface as a REST service, the administrator must specify service alias for the selected interface. If the selected interface type is Java Bean Services, Application Module Services, or Business Service Objects, the administrator also needs to specify HTTP verbs for the desired methods contained in the selected interface before deployment.

If the custom REST 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 3-20.

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 3-36.

Viewing and Downloading Custom Composite Services

If a custom interface is needed for a composite service - BPEL type, the integration developer will first create a composite service by orchestrating discrete native services into a meaningful process flow using BPEL. Based on the annotation standards specifically for composite service, the developer will then annotate the composite service, and create and unzip the JAR file of the BPEL project.

After appropriate validation on the BPEL project JAR files to ensure the compliance with the composite service annotation standards, the administrators will then upload them to the Integration Repository.

Viewing Custom Composite Services

To view a custom composite service, from the Search page, select 'Composite' from the Interface Type field and then click **Show More Search Options** to select 'Custom' from the Interface Source drop-down list along with any product family or scope as the search criteria.

By clicking a custom composite service name link from the search results, you will find the composite service interface details page displaying composite service details for this selected custom interface.

Downloading Custom Composite Services

Similar to downloading native packaged 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.

Note: An integration repository developer can further unzip the BPEL .JAR file and open the BPEL file in Oracle JDeveloper for further modification on service endpoints if needed. Additionally, the integration repository developer can deploy the BPEL process if necessary. Since composite services are typically not deployed within Oracle E-Business Suite, a separate BPEL PM (SOA Suite or a third party BPEL PM server) is needed to deploy the BPEL composite services.

For example, you can deploy the BPEL process to Oracle BPEL server through Oracle BPEL Process Manager. See *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide* for details.

For more information on how to download a composite service, see Downloading Composite Services, page 4-4.

6

Securing Web Services

This chapter covers the following topics:

- Overview
- Managing Function Security and Data Security
- Managing Role-Based Access Control Security
- Managing MOAC Security
- Managing Web Service Security

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 and the execution of integration interfaces and services, Oracle E-Business Suite integrated SOA Gateway uses the following approaches to enforce the security:

- 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 6-9

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 only allows users with authorized privileges to access or execute 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 without security checks.

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 which application data in the row. Regardless of the interface types, APIs are stored procedures that enable you to insert and update data in Oracle E-Business Suite. When having the function security layer enforced on the access to an API, it actually implicitly restricts the data access to the application.

Building on function security, data security provides another layer of security control to model and enforce security authorizations of specific data records. In other words, data security further refines the security of accessing application records down to the data level.

To allow appropriate users with right privileges to execute certain methods within an API, the concept of security grant is used to reinforce the security with a flexible mechanism. This approach enables the data access privileges to be granted to an appropriate user, user group, or all users. To accomplish this, the interface methods of an inbound API are precreated as permissions and stored in AOL's function repository. An Integration Repository Administrator can select one or more methods contained in an API and then grant the selected method(s) to appropriate users.

An integration repository administrator can create security grants in the following ways:

- If an inbound service-enabled interface has only one method, then this single method will be the default selection in creating grants.

  Concurrent Program interface type contains only one method. User security for XML Gateway interface is managed in Oracle XML Gateway user interface.

- If there is more than one method contained in an interface, then the administrator can have a choice in either granting one method to appropriate users or granting multiple methods simultaneously to the users.

  Interface types containing multiple methods are PL/SQL, Business Service Object, Java interface, and Open Interface Table and View.

Creating Security Grants

Only integration repository administrators (or users who have the Integration Repository Administrator role) can create security grants by authorizing the access permission of a selected interface method or procedure and function to an appropriate user, user group, or all users.

Security grants are managed in the Grants tab for the interface types that can be exposed as REST services. These interfaces include PL/SQL APIs, Concurrent Programs, Business Service Objects, Java Bean Services, Application Module Services, and Open Interface Tables and Views. See: Managing Grants for Interfaces with Support for SOAP and REST Services, page 3-38.

> **Note:** XML Gateway interfaces can be exposed as SOAP services only, but the user security is managed in the Oracle XML Gateway user interface through the Trading Partner User Setup form. See: Managing XML Gateway User Security in the Trading Partner User Setup Form, page 3-15.

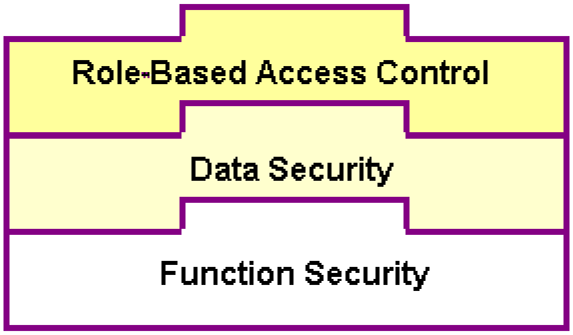
For more information on function security and data security, refer to the Oracle Application Object Library Security chapter, *Oracle E-Business Suite Security Guide*.

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 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 data security, users can be restricted by security rules to access or view only certain types of data on the screen once they have selected a menu while an administrator can have more data access to the same page.

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.

Role-Based Access Control (RBAC) Security for Oracle E-Business Suite Integrated SOA Gateway

By leveraging the concept of permission sets, each Integration Repository administrative function used in Oracle E-Business Suite Integrated SOA Gateway is first created as a permission and then relevant permissions are grouped into a permission set. Permission sets 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/Redeploy	FND_REP_DEPLOY	Deploy Web Service
Undeploy	FND_REP_UNDEPLOY	Undeploy Web Service

Privilege	Permission	Permission Display Name
Subscribe to Agent	FND_REP_SUBSCRIBE	Subscribe to Agent
Create Grants	FND_REP_METHOD_GRNT	Grant execute privileges to methods

Please note that the Deploy/Redeploy and Undeploy privileges are intentionally kept as separate permissions. This allows further security restriction on the service undeployment if needed.

Integration Repository Download Composite Service Permission Set

Because the download composite service feature can be performed by appropriate users not limited to the users with administrator or developer role, this feature has it own 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 appropriate 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

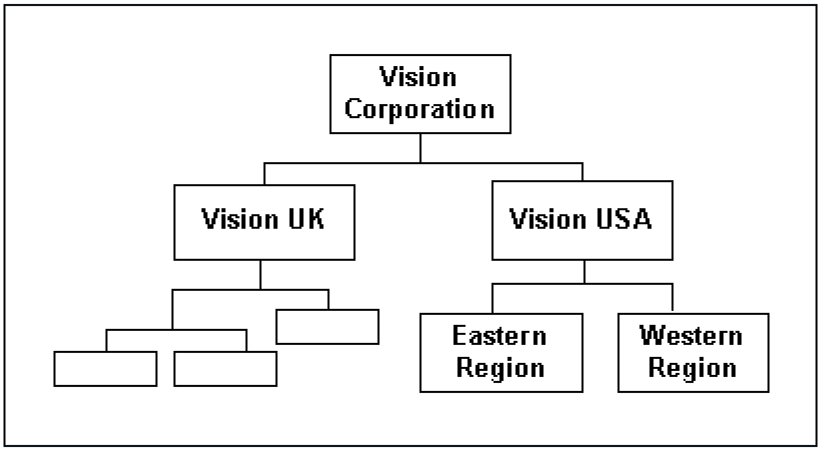
Multiple organizations can be sets of books, business groups, legal entities, operating units, or inventory organizations. You can define multiple organizations and the relationships between them in a single installation of Oracle E-Business Suite.

To have a secured way for users to only access data for the operating units they have access to, Oracle E-Business Suite Integrated SOA Gateway uses the MOAC security feature to determine the operating unit access and derive the Organization ID based on relevant profile values.

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 data in multiple operating units without changing responsibility.

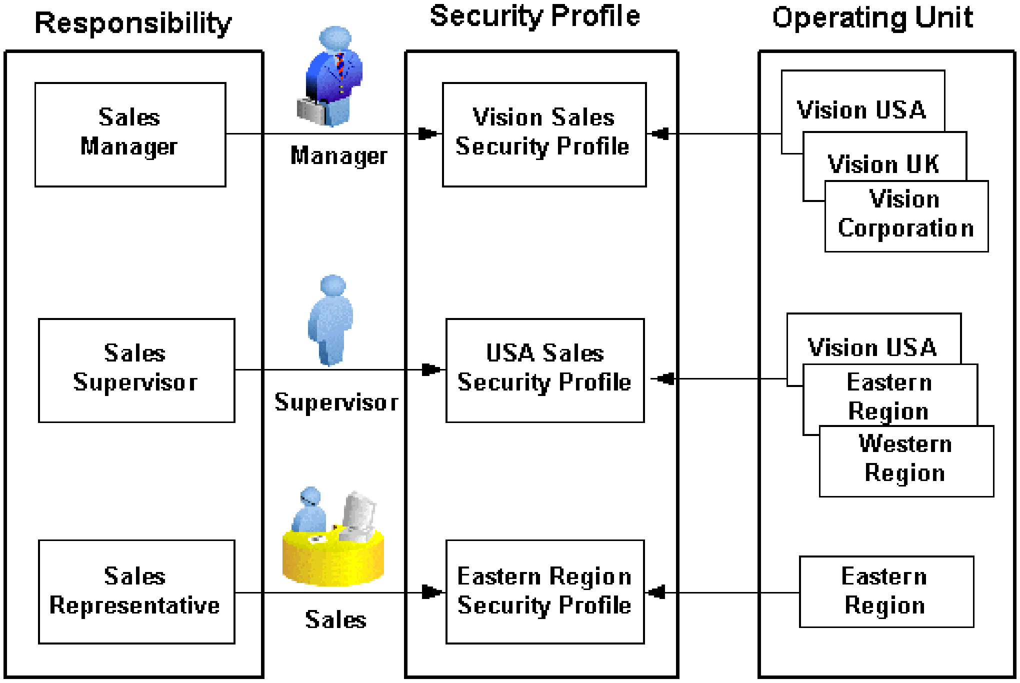
Security profiles are defined based on organization hierarchies. For example, a sales company consists of USA and UK operating units; the USA operating unit has Western Region Sales and East Region Sales. Sales managers are responsible for both USA and UK sales, supervisors are responsible for either USA or UK, and sales representatives are only responsible for their designated sales regions. The Sales organization hierarchy can be illustrated as follows:

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.

In addition to the *MO: Security Profile* profile option, MOAC security uses the following profile options to regulate the operating units access in a multi-organization environment:

- *MO: Operating Unit*

  Use this profile option if you want to access only one operating unit through a single responsibility. In this case, the responsibility determines the Organization ID.

  However, if you also define the *MO: Security Profile* profile option, then the *MO: Operating Unit* profile option will be ignored.

- *MO: Security Profile*

  Use this profile option if you want to access multiple operating units through a single responsibility.

  Because you can access multiple operating units without changing your responsibility, you need to set this profile value with multiple operating units. In addition, you must set the default operating unit in the *MO: Default Operating Unit* profile option. This allows the default Organization ID can be identified and entered to default organization for the context sensitive applications without requiring you to explicitly specify the Organization ID.

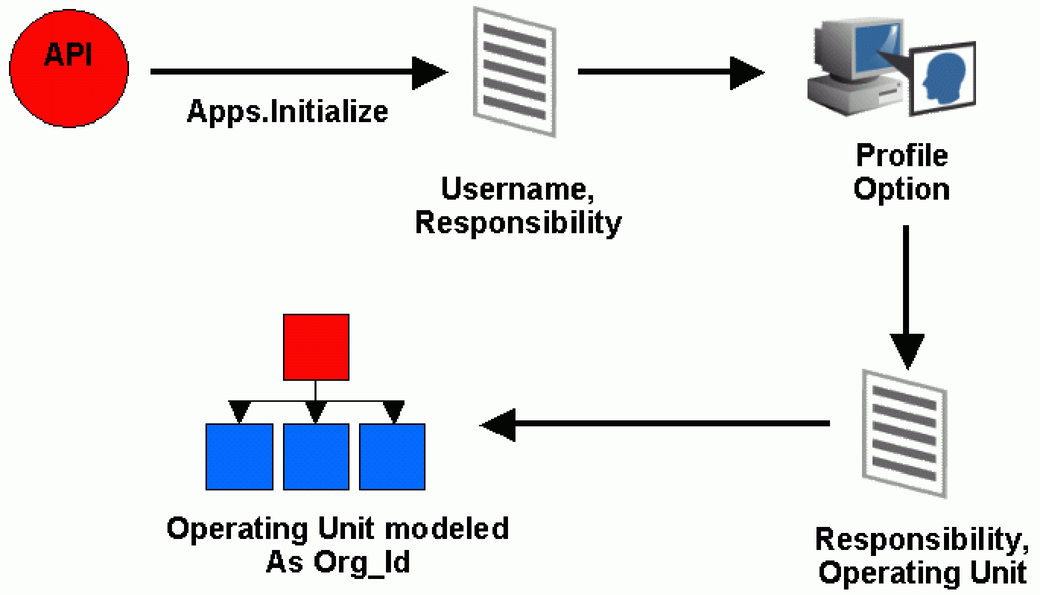
> **Additional Information:** The security profile allows you to assign multiple operating units for the same business group; global security profile allows you to assign multiple operating units across business groups. Based on your HR implementation and how you want to partition the data, you can decide which security profile would be good for you and meet your business needs.

- *MO: Default Operating Unit*

  Use this profile option in conjunction with the *MO: Security Profile* profile option to specify a default operating unit. This profile value determines the application entries once you log on to the system.

The following diagram illustrates how Oracle E-Business Suite uses the profile options in a multi-organization environment.

Applications Context for Multiple Organizations



1. When the system integrator runs, the process achieves the integration with Oracle E-Business Suite using PL/SQL APIs.

2. The Apps.Initialize process takes the parameters of Username and Responsibility.

3. With these parameters, a lookup on all System Profile Values assigned to that responsibility is done to determine the Operating Unit within a multi-organization

environment.

4. The Operating Unit is modeled as Organization ID derived from the security profile values, such as the values in *MO: Operating Unit* or *MO: Security Profile* profile options.

5. The data is read and written into the Oracle E-Business Suite with the parameters of Username, Responsibility and Organization ID.

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 multiple authentication types for inbound service requests through the following security mechanisms:

- For SOAP Services

  - UsernameToken Based Security, page 6-10

  - SAML Sender-Vouches Token Based Security, page 6-11

  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.

  SOA Provider supports the selected authentication type(s) during the service deployment cycle. When a SOAP request message is received through SOA Provider, the SOAP message is passed on to OC4J Web Service Framework for authentication based on the selected authentication type(s).

  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 6-14

  - Token Based Authentication, page 6-15

  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).

UsernameToken Based Security

In the UsernameToken based security mechanism, the username/password is sent in the SOAP header. The SOA Provider authenticates the user based on this information. The username/password sent with the SOAP header 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. Encryption is not supported in this release.

When a SOAP request message is received through SOA Provider, the SOAP message is passed on to OC4J Web Service Framework for authentication. The framework authenticates the SOAP message 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>sysadmin</wsse:Username>
     <wsse:Password>password</wsse:Password>
     </wsse:UsernameToken>
   </wsse:Security>
...
  </S11:Header>
...
</S11:Envelope>
```

A typical 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-wsswssecurity-secext-1.0.xsd>
  <wsse:UsernameToken>
     <wsse:Username>sysadmin</wsse:Username>
     <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-username-token-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 an 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 method called 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 as its authentication mechanism, this SOAP request message containing or referencing SAML assertions is received through SOA Provider and passed on to OC4J Web Service Framework for authentication. The framework authenticates the SOAP message based on the `wsse:security` Web Security headers. As part of the validation and processing of the assertions, the receiver or authentication framework 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.

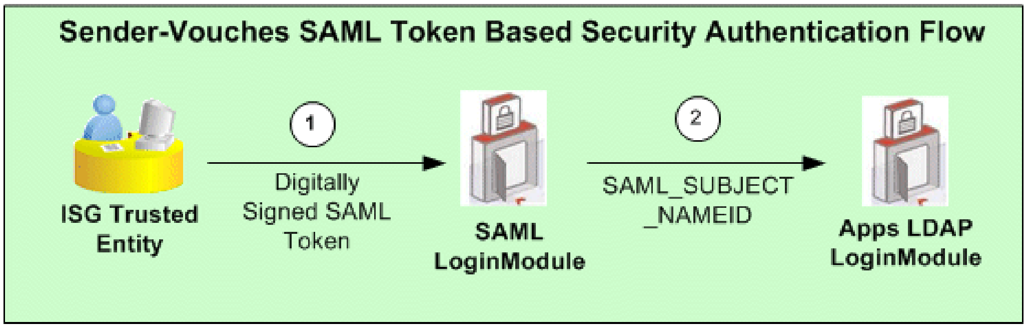
In other words, in order to validate and authenticate a non-Oracle E-Business Suite user, but logged on to the enterprise information system, a trusted sender-vouches SAML token security mechanism must be used to establish the correspondence between the SOAP message and the SAML assertions added to the SOAP message.

Note: Since anyone 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.

Important: To ensure SAML Token security works properly, necessary setup steps need to be performed. For setup instructions, see Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12, My Oracle Support Knowledge Document 556540.1. For more SAML Token security information, see Oracle E-Business Suite Integrated SOA Gateway Release Notes for Release 12.1.3, My Oracle Support Knowledge Document 1096553.1.

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. Once the SAML token is validated by the OC4J authentication framework (`SAMLLoginModule`), the SAML principal (username in `NameIdentifier`) will be obtained and verified against LDAP for Single Sign-On (SSO) users or Oracle E-Business Suite `FND_USER` for non-SSO users.

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
SAML LoginModule
2
SAML_SUBJECT _NAMEID
Apps LDAP LoginModule

1. A trusted application authenticates an 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 sign SAML Assertions in SOAP requests using that public key.

2. Oracle E-Business Suite Integrated SOA Gateway authentication layer verifies the digital signature and extracts the SAML Token after the verification.

3. OC4J `SAMLLoginModule` verifies the SAML conditions to ensure that it is a valid

SAML Token.

After the verification, OC4J SAMLLoginModule extract the SAML name identifier from the token. Application LDAPLoginModule connects to LDAP to verify if the name identifier specified in the SAML_SUBJECT_NAMEID exists in OID.

- If it does exist, the associated orclguid will be retrieved and mapped that to an Oracle E-Business Suite user with an equivalent orclguid for SSO users.

- If it does not exist, the associated orclguid will be mapped to an actual user in FND_USER table through the connection of OID to locate the user in FND_User table who has the equivalent orclguid. This authenticates the non-SSO users.

The format of the NameIdentifier indicates if the user has been authenticated against LDAP (SSO user) or Oracle E-Business Suite FND_USER (for non-SSO user). If the format is dn=xxxx, then this is a SSO user that has been authenticated against LDAP. Otherwise, this is a non-SSO user that 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="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" 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="2010-02-27T17:26:21.241
Z" 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</ConfirmationMethod>
    </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</ConfirmationMethod>
    </SubjectConfirmation>
  </Subject>
 </AuthenticationStatement>
</Assertion>
```

- `Issuer`: The value of this attribute should be defined in the `system-jazn-data.xml` as part of the `LoginModule`.

- `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 SSO user. Otherwise, the username is verified in `FND_USER` table for non-SSO user.

- `SubjectConfirmation`: It should be sender-vouches.

For information on how the SAML Token (sender vouches) type 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.

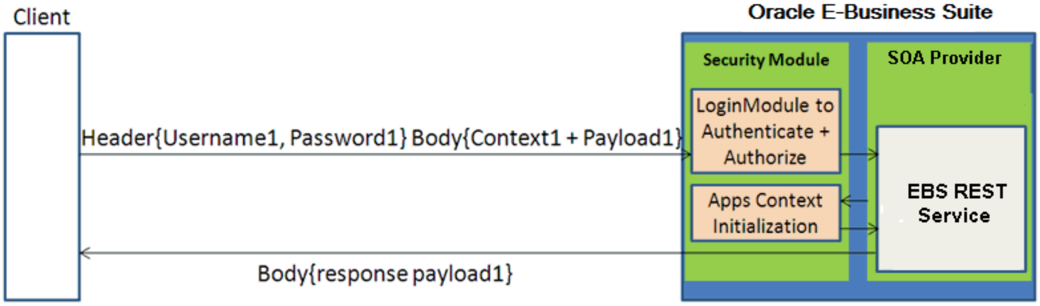
The LoginModule in turn extracts the credentials from HTTP header, authenticates user against Oracle E-Business Suite user table, and establishes identity for the authenticated user.

- For the authenticated and authorized user request, a security service is invoked to initialize the applications context, and then the REST service is executed.

- For the unauthenticated or unauthorized user request, a system fault is returned 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
SOA 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) may be sent as `Cookie` in HTTP header.

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>xxxxxxxxxxxxxxxxxxxxxxxxxx</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.

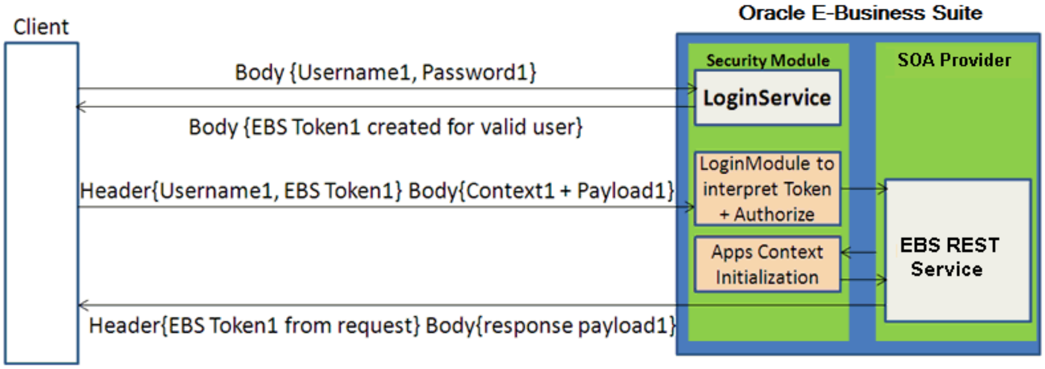
```
POST /webservices/rest/Invoice/create_invoice
Cookie: <accessTokenName>=<accessToken>
Content-Type: application/xml
```

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, a security service is invoked to initialize the applications context, and then the REST service is executed. Otherwise, system fault is returned.

The following diagram illustrates the authentication process flow of Token Based Authentication security:

Authentication Process for Token Based Authentication Security


Oracle E-Business Suite
Client
Body {Username1, Password1}
Body {EBS Token1 created for valid user}
Header{Username1, EBS Token1} Body{Context1 + Payload1}
Header{EBS Token1 from request} Body{response payload1}
Security Module
LoginService
LoginModule to interpret Token + Authorize
Apps Context Initialization
SOA Provider
EBS REST Service

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 leverages FND Logging Framework and provides an enhanced, flexible Web Service logging mechanism. It lets you easily monitor system activities, track log messages, and troubleshoot any issues encountered during service generation and deployment, as well as the invocation of Oracle E-Business Suite services by Web service clients. More importantly, logging can be enabled at the site level and user level either for all services or specific services or operations depending on your settings.

In other words, with proper logging setups and configuration, all design-time and run-time activities of all integration interfaces within Oracle E-Business Suite can be logged through this framework. You can configure and manage these log settings through a centralized user interface, and purge log messages through SOA Monitor if needed.

> **Note:** Logging feature is supported for SOAP services only. This feature is not supported for REST services.

Key Features

The enhanced Web service logging feature not only inherits the key features of FND Logging Framework, but also includes the following features:

- It provides centralized, user-friendly user interface for logging configuration for Oracle E-Business Suite SOAP services.
- It extends FND Logging Framework to allow logging to be set at the site level and at the user level.
- Flexible logging mechanism allows you to enable logging either for all services or specific services or operations.

- All design-time and run-time SOAP service activities within Oracle E-Business Suite can be logged and audited through this framework if the services have logging enabled properly.

- It provides integrated log view mechanism allowing you to view SOAP service generation and deployment logs through Integration Repository and service processing logs through SOA Monitor if the logging is configured properly.

- Log messages can be purged from the database tables through SOA Monitor.

- Log messages can be correlated across application and database servers.

To better understand the logging feature, the following topics are discussed in this chapter:

- Accessing the Log Configuration User Interface, page 7-2

- Viewing Existing Logging Configurations, page 7-4

- Adding a New Log Configuration, page 7-5

- Updating an Existing Configuration, page 7-14

- Deleting an Existing Configuration, page 7-14

- Viewing, Deleting and Exporting Log Messages, page 7-16

Accessing the Log Configuration User Interface

Integration repository administrators can configure log settings at different log granularity level, and manage log activities including adding, updating and deleting services and operations through a centralized Log Configuration user interface.

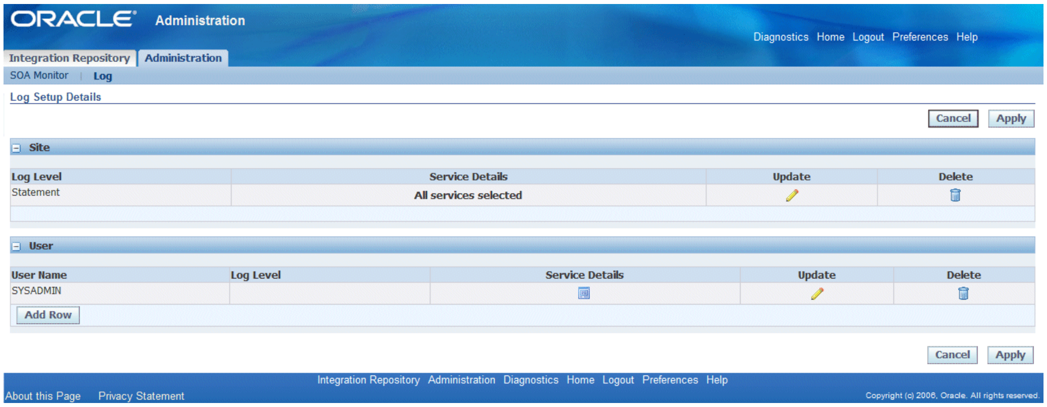
To access the log settings page, log on to Oracle E-Business Suite as a user who has the Integration Repository Administrator role.

Select the **Integrated SOA Gateway responsibility** from the navigation menu and then select **Log** link from the Administration section. The Administration tab appears with the Log subtab.

> **Note:** Only users who have the Integration Repository Administrator role can find the Administration section available 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 now grouped under the Administration section and displayed in the Administration tab. These tasks include managing log setups in the Log subtab and managing

SOAP requests in the SOA Monitor subtab.

Log Setup Details Page


ORACLE Administration
Diagnostics Home Logout Preferences Help
Integration Repository
Administration
SOA Monitor | Log
Log Setup Details
Cancel
Apply
Site

Log Level	Service Details	Update	Delete
Statement	All services selected	✎	🗑

User Name	Log Level	Service Details	Update	Delete
SYSADMIN		▦	✎	🗑

Add Row
Cancel
Apply
Integration Repository Administration Diagnostics Home Logout Preferences Help
About this Page
Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

The Log Setup Details page is the entry page to all logging setup and management activities. You can perform the following tasks through this page:

- Viewing Existing Logging Configurations, page 7-4

  All existing logging settings grouped by site level and user level are automatically displayed in the Log Setup Details page.

- Adding a New Logging Configuration, page 7-5

  By clicking **Add Row** for the desired log granularity, an empty row appears allowing you to add new log configuration and log severity.

  You can either enable all services for your log configuration or select specific services and operations and set proper log severity level for each selected service and operation. This option enables the logging feature further down to the service or operation level.

- Updating an Existing Configuration, page 7-14

  You can update an existing configuration by clicking the **Update** icon from the log search result table.

- Deleting an Existing Configuration, page 7-14

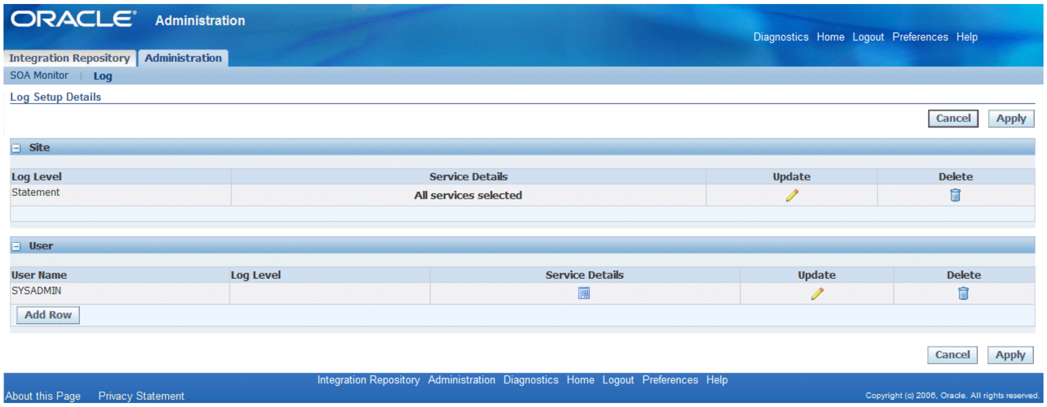
  Similar to the update activity, you can delete an existing configuration by clicking the **Delete** icon from the log search result table.

  This deletes the configuration for the selected services or operations.

Viewing Existing Logging Configurations

Logging can be enabled through the site and user levels. Once you are directed to the Log Setup Details page, all existing configurations will be automatically displayed either in the Site region or the User region based on the level of the log has been set. For examples, all log settings are listed in the User region if they are enabled at the user level.

Log Setup Details Page with Existing Configurations Displayed


ORACLE
Administration
Diagnostics Home Logout Preferences Help
Integration Repository
Administration
SOA Monitor | Log
Log Setup Details
Cancel Apply
Site

Log Level	Service Details	Update	Delete
Statement	All services selected	✎	🗑

User Name	Log Level	Service Details	Update	Delete
SYSADMIN		▤	✎	🗑

Add Row
Cancel Apply
Integration Repository Administration Diagnostics Home Logout Preferences Help
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

Each log entry listed in the table contains log severity, service details, or a specific username if it is enabled at the user level. All the log setups listed in the table format are registered for what will be logged.

From the configuration or log list table, you can perform the following tasks:

- Add a new log configuration by clicking **Add Row**. See: Adding a New Log Configuration, page 7-5.

- Update an existing configuration by clicking the **Update** icon for a desired log configuration.

  See: Updating an Existing Configuration, page 7-14.

- Delete an existing configuration by clicking the **Delete** icon for a desired log configuration.

  See: Deleting an Existing Configuration, page 7-15.

To view existing log configuration:

1. Log on to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility.

   From the navigation menu, click the Log link from the Administration section. The Log Setup Details page is displayed.

2. All existing log configurations are automatically displayed and grouped by the level of the log has been set either at the site or user level.

3. To update an existing configuration, click the **Update** icon for a desired setting.

   The Create Log Setup page is displayed where you can update the log severity for a selected log setting or add more services.

4. To delete an existing configuration, click the **Delete** icon for a desired log setting.

5. To add a new configuration, click **Add Row** to add a new log setting.

Adding a New Logging Configuration

After selecting the Log link from the navigation page, the Administration tab appears with the Log subtab selected.

Oracle E-Business Suite Integrated SOA Gateway allows logging feature to be enabled at the following granularity levels or log categories:

> **Note:** Logs for actions such as Generate, Deploy, Undeploy, and Redeploy services should be written only if the logging is enabled for the service at the Site level. Without log configuration set, the logs will not be written.

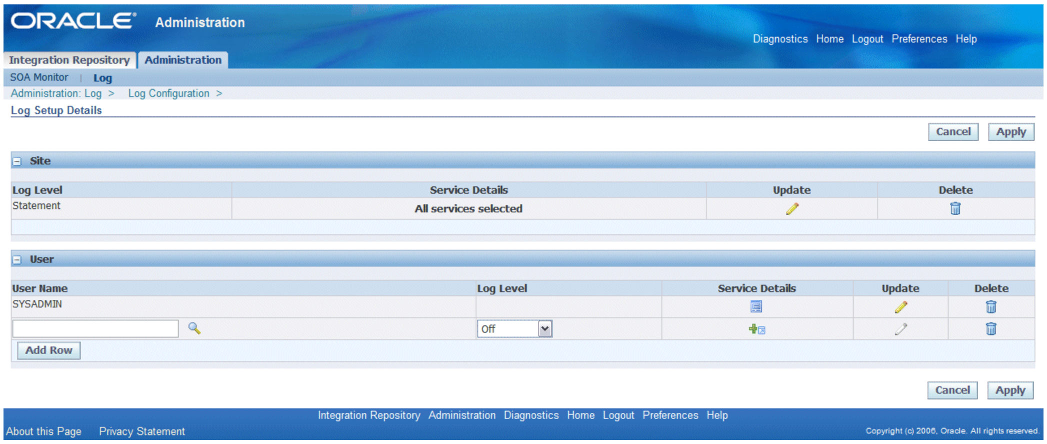
- *Site Level:* This indicates that the logging is implemented for all services or specific services or operations within Oracle E-Business Suite Integrated SOA Gateway.

  Since it is a site level configuration, only one site level log can be set on each instance. Once a site level log exists, **Add Row** in the Site region disappears indicating that you cannot add another site level log unless the existing one is deleted.

- *User Level:* This indicates that the logging is implemented at a specific user (such as `sysadmin`) or list of users.

> **Important:** If a user had set up a log configuration for enabling all services at the site or user level, and then the user decides to update the configuration only for specific services or operations later on, the 'all services' level configuration will no longer be valid and will be overridden by the newer update. For the same reason if the situation is reversed (selected services/operations first, and then 'all services' later), the logging with all services will be acknowledged.

Log Setup Details Page for Adding a New Configuration


ORACLE Administration
Diagnostics Home Logout Preferences Help
Integration Repository
Administration
SOA Monitor | Log
Administration: Log > Log Configuration >
Log Setup Details
Cancel
Apply
Site
User
Add Row
Cancel
Apply
Integration Repository Administration Diagnostics Home Logout Preferences Help
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

Log Level	Service Details	Update	Delete
Statement	All services selected	✎	🗑

User Name	Log Level	Service Details	Update	Delete
SYSADMIN		▤	✎	🗑
	Off		✎	🗑

When adding a new configuration, you need to specify the log level for your log setting. The following table describes the available log levels used for the logging configuration:

Log Level

Severity	Description	Audience	Examples
6-Unexpected	Fatal errors that prevent system execution. It can be raised as alerts.	Customer System Administrator, Support, Development	Required file not found. Database failure in placing an order.
5-Error	End user errors.	Customer System Administrator, Support, Development	Authentication failure. Invalid input value.
4-Exception	Internal software failure condition.	Customer System Administrator, Support, Development	Detailed exception stack trace. Handled Java exceptions.
3-Event	Key progress events, and configuration.	Customer System Administrator, Support, Development	User authenticated. Starting business transaction.

Severity	Description	Audience	Examples
2-Procedure	API level flow of application and important events.	Development	Calling an API. Returning from an API.
1-Statement	Low level detailed messages.	Development	Copying buffer x to y.

Adding a Site Level Log

In the Site region, click **Add Row**. This creates a row in editable mode. Select the log severity level. You can enable a site level log in the following ways:

- **Enabling all services:** After adding log severity level from the drop-down list, click **Apply** to save the record (along with any other record present at the user level). A confirmation message appears indicating your site level log setup is successfully saved.

  Without further adding services to your log, this enables all services for the site level. The Service Details field is displayed with 'All services selected' for this site level log setting.

- **Enabling selected services or operations:** After adding log severity level from the drop-down list, click the **Service Details** icon instead to add new services or operations to your site level log.

  To add and enable services in the log setup, see Adding Services to a Log Configuration, page 7-8

Once the newly created record exists in the system, **Add Row** in the Site region is no longer displayed. The new log is shown in read-only mode.

Adding a User Level Log

To add a user level log configuration, click **Add Row** in the User region. This adds an empty row allowing you to enter the following basic log information.

- **Log Level**: Select an appropriate value from the drop-down list.

- **Username**: Specify a valid username.

  This field appears only if the log is configured at the user level.

  If this field is not selected and you click the **Service Details** icon to add services or operations, an error message appears indicating that you must enter a valid username for the configuration.

Similar to the site level log configuration, a user can add site level logs in the following ways:

- **Enabling all services:** Without further adding services or operations to your log, click **Apply** to save and validate the selected username. If it is a valid username, a confirmation message appears indicating the user level log setup is successfully saved in the system.

  In this situation, the Service Details field is displayed with 'All services selected' for this log setting.

- **Enabling selected services or operations:** After adding a valid username and selecting log severity level from the drop-down list, click the **Service Details** icon instead to add new services or operations to your user level log. This enables the logging further down to the service or operation level.

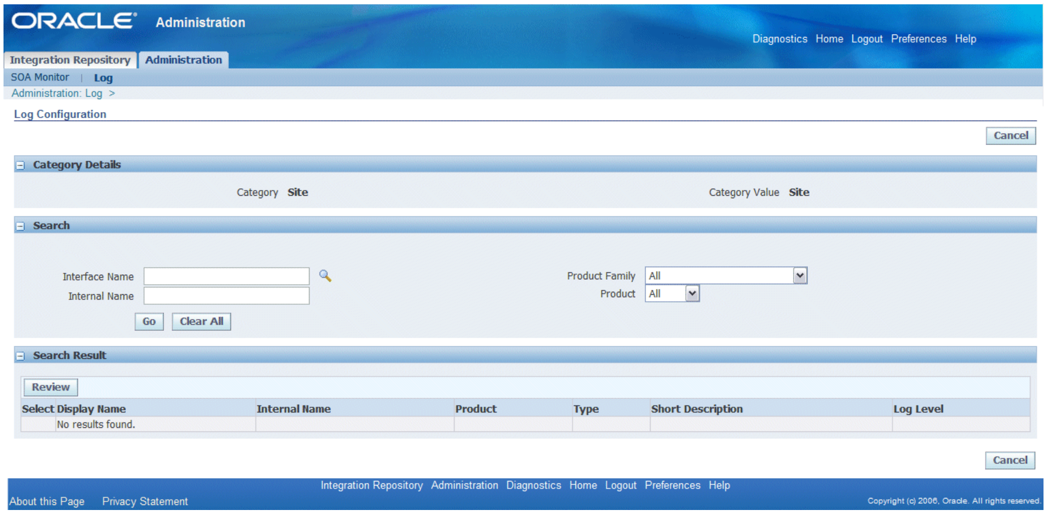
  To add and enable services in the log setup, see Adding Services to a Log Configuration, page 7-8

Adding Services to a Log Configuration

Log can be set at the Site and User levels. The administrator should be able to enable or disable logging for one or more services or operations contained in the selected services while configuring the log settings.

Click the **Service Details** icon in the log list table. The Log Configuration page is displayed with the selected log category indicating that the log is set at the Site level or at the User level with a specific username.

Log Configuration Page


ORACLE
Administration
Diagnostics Home Logout Preferences Help
Integration Repository
Administration
SOA Monitor | Log
Administration: Log >
Log Configuration
Cancel

Category Details

Category Site

Category Value Site

Search

Interface Name

Internal Name

Product Family All

Product All

Search Result

Select	Display Name	Internal Name	Product	Type	Short Description	Log Level
	No results found.					

Cancel
Integration Repository Administration Diagnostics Home Logout Preferences Help
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

Searching for Services in the Search Region

The Search region contains the following search fields allowing you to locate the desired services based on your criteria:

- Interface Name
- Internal Name
- Product Family
- Product

After executing the search by clicking **Go**, all the services that match the criteria are displayed in the Search Result region.

Logging can be enabled at the service package level (such as Order Capture - PL/SQL API package) with all methods contained in the service or only with specific operation methods (such as 'Create_Quote and Update_Quote' methods in the Order Capture API).

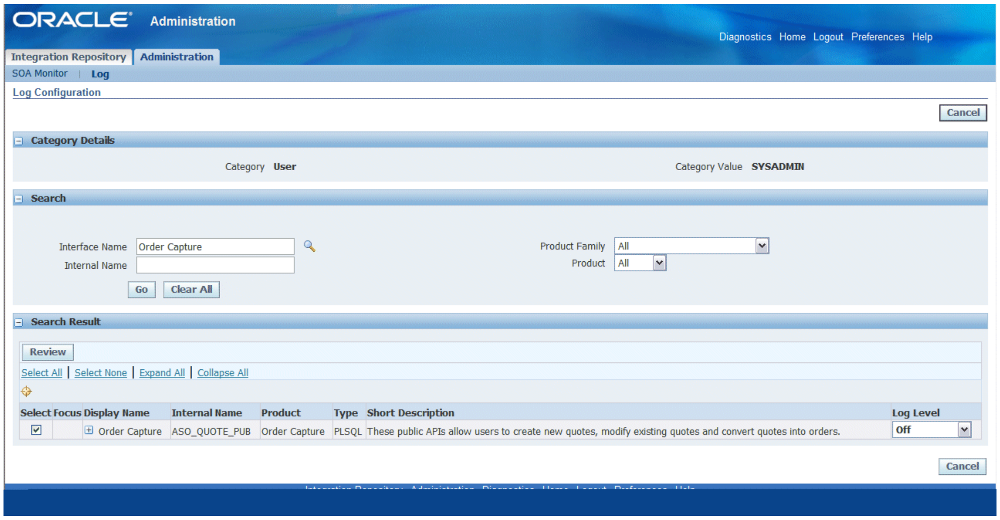
- **Enabling Logging at the Service Level**

  Regardless of the log category enabled at the site or user level, you can add the services for your log setups.

  From the Search Result region with a list of matched services, click the **Select** check boxes for the desired services that you want the logging to be enabled.

  Select the log level value for each selected service using the drop-down list. The log level information is automatically populated if it is specified earlier in the Log Setup Details page.

  Log Configuration Page for Enabling Logging at the Service Level



Category Details

Category: User
Category Value: SYSADMIN

Search

Interface Name: Order Capture
Internal Name:
Product Family: All
Product: All

Search Result

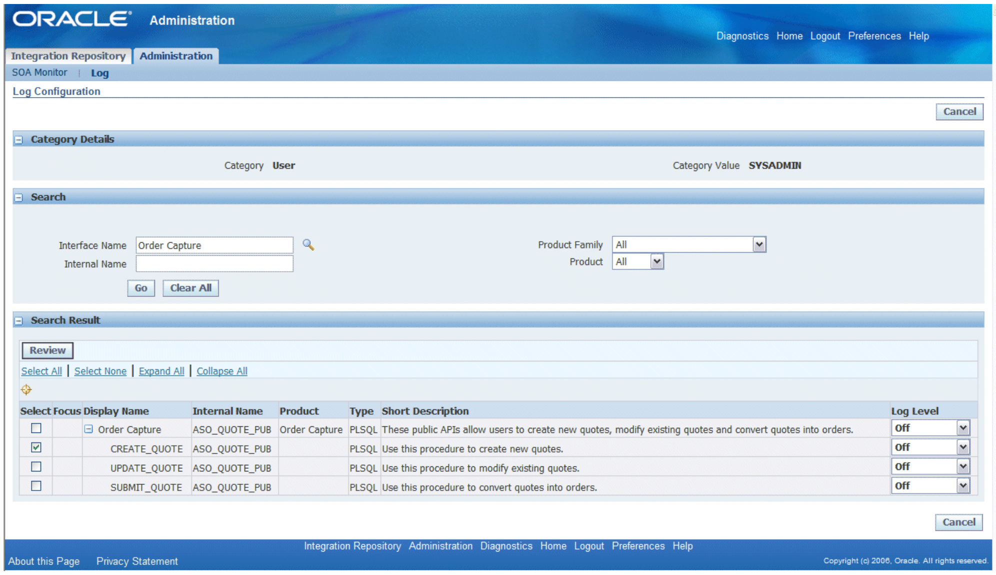
Select	Focus	Display Name	Internal Name	Product	Type	Short Description	Log Level
☑		⊞ Order Capture	ASO_QUOTE_PUB	Order Capture	PLSQL	These public APIs allow users to create new quotes, modify existing quotes and convert quotes into orders.	Off

  Without specifying a specific operation within a service, all operation methods contained in the selected service(s) are all included and have logging enabled.

- **Optionally Enabling Logging at the Operation Level**

This option provides the logging flexibility that instead of enabling all the operation methods contained in the selected service, you can identify one or more specific operation methods contained in the service package from the search result to further define the logging configuration to the operation level.

Log Configuration Page for Enabling Logging at the Operation Level


ORACLE Administration
Diagnostics Home Logout Preferences Help
Integration Repository
Administration
SOA Monitor
Log
Log Configuration
Cancel
Category Details
Search
Go
Clear All
Search Result
Review
Select All | Select None | Expand All | Collapse All
Cancel
Integration Repository Administration Diagnostics Home Logout Preferences Help
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

Category: User

Category Value: SYSADMIN

Interface Name: Order Capture

Internal Name:

Product Family: All

Product: All

Select	Focus	Display Name	Internal Name	Product	Type	Short Description	Log Level
☐		⊟ Order Capture	ASO_QUOTE_PUB	Order Capture	PLSQL	These public APIs allow users to create new quotes, modify existing quotes and convert quotes into orders.	Off
☑		CREATE_QUOTE	ASO_QUOTE_PUB		PLSQL	Use this procedure to create new quotes.	Off
☐		UPDATE_QUOTE	ASO_QUOTE_PUB		PLSQL	Use this procedure to modify existing quotes.	Off
☐		SUBMIT_QUOTE	ASO_QUOTE_PUB		PLSQL	Use this procedure to convert quotes into orders.	Off

To set the granularity at the operation level, select a desired service from the result table and click the plus sign (+) in the service name or click the **Expand All** link. This expands the service with all operation names in a tree structure. Select desired operations you want the logging to be implemented by clicking appropriate **Select** check boxes. Select the log severity level value from the drop-down list for each selected operation.

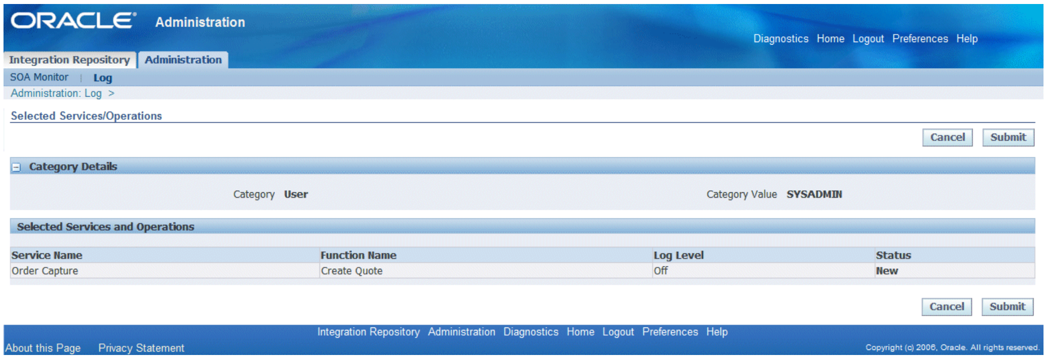
> **Note:** If both the service name and one or more corresponding operations are selected, then only the selected operations are considered. To enable a log only at the service level, none of its corresponding operations should be selected.

For example, if the 'Create_Quote' method in the Order Capture API is the only selected operation method name, then none of the order capture procedures will be logged except for new quote creation.

Click **Review** to review your selected services or operations for the log settings. The Selected Services/Operations page is displayed with the selected service(s) or operation name(s), log level, and status information indicating whether each selected service or operation is a New selection or not.

Note: If the selected service or operation is already present, 'Existing' is displayed in the Status field. In this situation, that existing configuration will be overwritten by the new changes.

Selected Services/Operations Page


ORACLE
Administration
Diagnostics Home Logout Preferences Help
Integration Repository
Administration
SOA Monitor | Log
Administration: Log >
Cancel
Submit
Integration Repository Administration Diagnostics Home Logout Preferences Help
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

Selected Services/Operations

Category Details

Category User

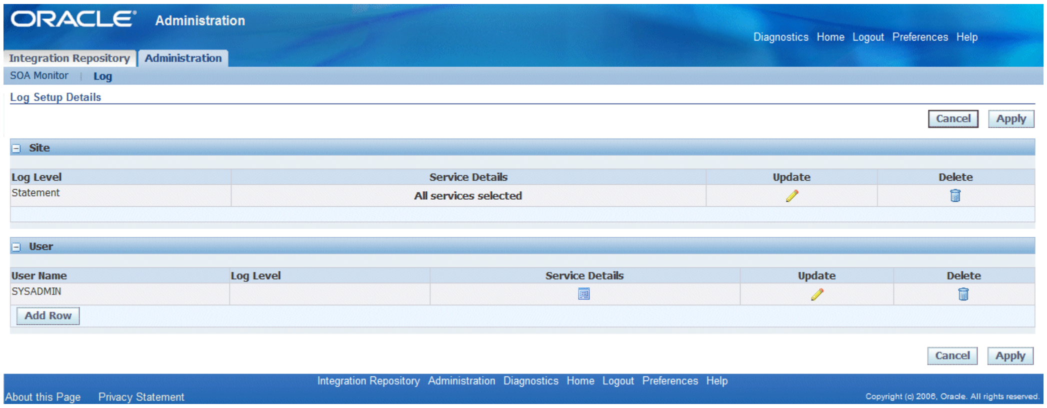
Category Value SYSADMIN

Selected Services and Operations

Service Name	Function Name	Log Level	Status
Order Capture	Create Quote	Off	New

Click **Submit** to confirm the action and add the selected services or operations to your log configuration. A confirmation message appears indicating that the log setup information is successfully saved in the system.

Log Setup Details Page


ORACLE
Administration
Diagnostics Home Logout Preferences Help
Integration Repository
Administration
SOA Monitor | Log
Cancel
Apply
Add Row
Integration Repository Administration Diagnostics Home Logout Preferences Help
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

Log Setup Details

Site

Log Level	Service Details	Update	Delete
Statement	All services selected	✎	🗑

User

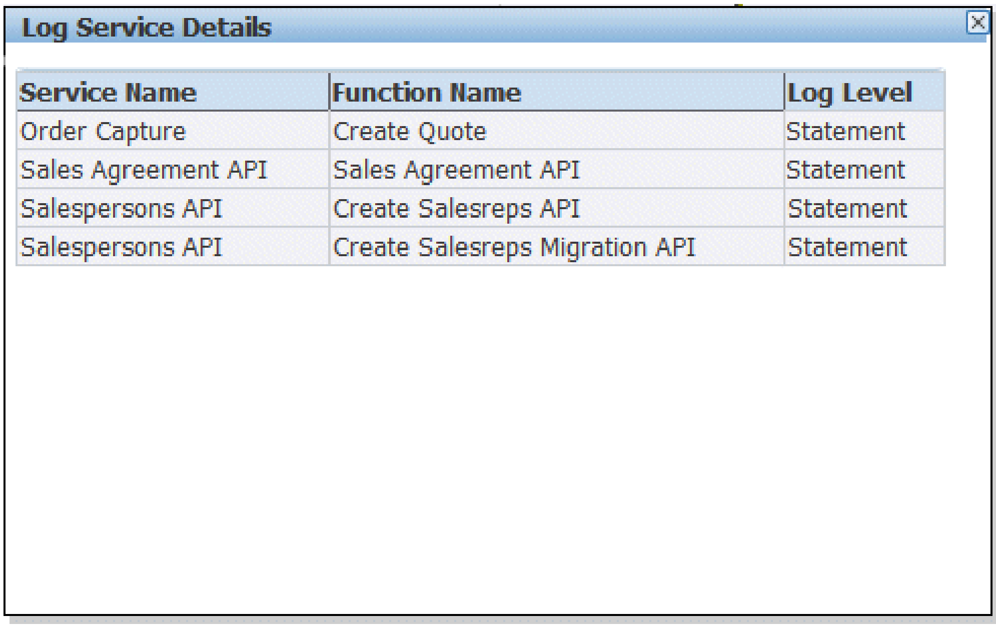
User Name	Log Level	Service Details	Update	Delete
SYSADMIN		▤	✎	🗑

The following icons in the Log Setup Details page become visible allowing you to perform further update on this log setup:

- **Service Details**: It provides you a quick view on the selected log details.

  Click the **Service Details** icon to open the Log Service Details pop-up window with a list of enabled services and details including service name, specific operation names or all the operations included in the selected service, and log severity.

Log Service Details Pop-up Window



Log Service Details

Service Name	Function Name	Log Level
Order Capture	Create Quote	Statement
Sales Agreement API	Sales Agreement API	Statement
Salespersons API	Create Salesreps API	Statement
Salespersons API	Create Salesreps Migration API	Statement

- **Update**: It allows you to update an existing log setup.

  See: Updating an Existing Configuration, page 7-14.

- **Delete**: It removes the selected log setup from the system.

  See: Deleting an Existing Configuration, page 7-15.

To add a new log configuration:

1. Log on to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility.

   From the navigation menu, click the Log link from the Administration section. The Log Setup Details page is displayed.

2. To add a new configuration, click **Add Row** for the desired log level or category that you want the log to be configured.

   An empty row appears allowing you to enter the following log information:

   - Username: Specify an appropriate username if the log is configured at the user level.

   - Log Level: Select an appropriate value from the drop-down list.

3. Click **Apply** to save and validate your log setting if you do not want to add service to your log.

This enables all services for your log configuration.

4. To add the service information to the new log configuration, click the **Service Details** icon to open the Log Configuration page.

5. In the Log Configuration page, enter the following search criteria in the Search region:

   - Interface Name
   - Internal Name
   - Product Family
   - Product

   Clicking **Go** retrieves all the services that match the criteria.

6. From the Search Result region with a list of matched services, configure your log in the following ways:

   - Add services with all the operations

     1. Click the **Select** check boxes for the desired services that you want the logging to be enabled.
     2. Select the log level value for each selected service using the drop-down list.

   - Add selected operations in a service

     1. Select a desired service from the result table and click the plus sign (+) in the service name to expand and list the service with all operation names.
     2. Select desired operations you want the logging to be implemented by clicking appropriate **Select** check boxes.
     3. Select the log severity level value from the drop-down list for each selected operation.

7. Click **Review** to review your selected services or operations for your log settings. The Selected Services/Operations page is displayed with selected services or operations, log level, and status.

8. Click **Submit** to confirm the action and add the selected services or operations to your log setting in the Log Setup Details page. A confirmation message appears indicating the log configuration is successfully saved in the system.

9. To verify your configuration details, click the **Service Details** icon in the Log Setup

Details page.

The Log Service Details pop-up window appears with a list of enabled services and details including service names, operation names (All), and log severity.

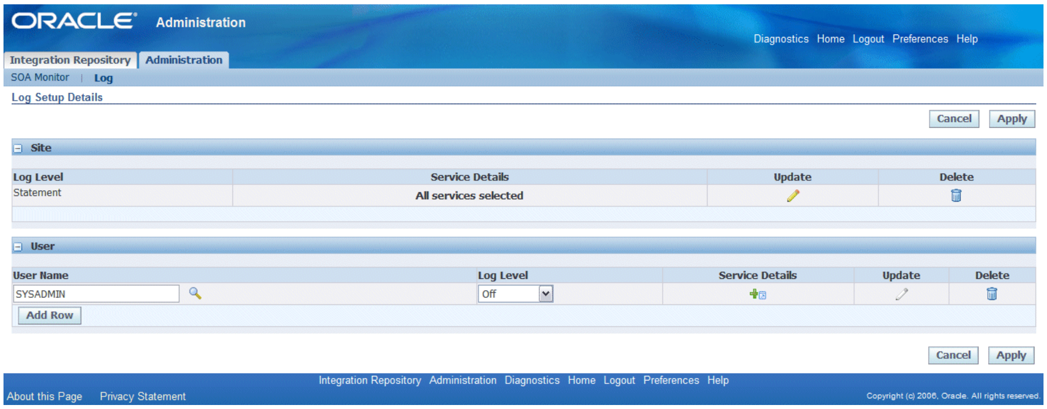
10. Click the **Update** icon to further edit your configuration.

Updating an Existing Configuration

From the Log Setup Details page, you can modify an existing log configuration including changing log severity and adding new services or operations.

To enable the update, click the **Update** icon for a given configuration. You should notice that the read-only text fields now become updatable and the **Service Details** icon is also enabled (with a '+' sign) allowing you to add new services or operations.

Log Setup Details Page with Service Details "+" Icon


ORACLE Administration
Diagnostics Home Logout Preferences Help
Integration Repository
Administration
SOA Monitor
Log
Log Setup Details
Cancel
Apply
Add Row
Integration Repository Administration Diagnostics Home Logout Preferences Help
About this Page
Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

Site

Log Level	Service Details	Update	Delete
Statement	All services selected	✎	🗑

User

User Name	Log Level	Service Details	Update	Delete
SYSADMIN	Off	+	✎	🗑

In the Log Setup Details page, you can perform simple update on the basic log setting such as log severity level or username if it is for a user level log. Click **Apply** to save and validate the change.

To add services or operations to the log setting, click the enabled **Service Details** icon to invoke the Log Configuration page where you can search and add desired services. After adding new services or operations to your log configuration, click **Review** to open the Selected Services/Operations page with selected services or operations, log level and status information indicating whether each selected service or operation is a new selection or not. Click **Submit** to confirm and add the selection to the log configuration and you are redirected back to the Log Setup Details page.

Click the **Service Details** icon to have a quick view on the selected service details.

> **Important:** If a user had set up a log configuration for enabling all services either at the site or user level, and then the user decides to

update the configuration only for specific services or operations later on, the 'all services' level configuration will no longer be valid and will be overridden by the newer update. For the same reason if the situation is reversed (selected services/operations first, and then 'all services' later), the logging with all services will be acknowledged.

For information on how to add new services or operations, see Adding Services to a Log Configuration, page 7-8.

To update an existing logging configuration:

1. Log on to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility.

   From the navigation menu, click the **Log** link from the Administration section to open the Log Setup Details page.

2. To update an existing configuration, click the **Update** icon for a desired configuration.

   The read-only text fields now become updatable allowing you to update basic log information such as log severity or username if it is a user level log. Click **Apply** to save and validate the change.

3. To add new services or operations to the selected setting, click the enabled **Service Details** icon (with a '+' sign) to open the Log Configuration page allowing you to add more services.

   Once desired services or operations are selected, click **Review** to open the Selected Services/Operations page with selected services or operations.

   Click **Submit** to confirm and add the selection to the log setting.

Deleting an Existing Configuration

If an existing log configuration is no longer needed, you can remove it directly from the Log Setup Details page.

To delete a configuration, click the **Delete** icon for a desired configuration that you want to delete. This removes the record from the table and system. A confirmation message appears indicating that the selected log setup is successfully deleted. This disables the logging feature for the selected services or operations contained in the removed log setting.

If the deletion is at the site level, the **Add Row** button appears in the Site region allowing you to configure site level log setup.

To delete an existing logging configuration:

1. Log on to Oracle E-Business Suite as a user who has the Integration Repository

Administrator role. Select the Integrated SOA Gateway responsibility.

From the navigation menu, click the Log link from the Administration section to open the Log Setup Details page.

2. To delete an existing configuration, click the **Delete** icon for a desired configuration. The selected configuration should be removed from the list and system.

Viewing, Deleting, and Exporting Log Messages

Integration repository administrators can view, delete, and export the log messages recorded for the associated services or operations that have logging enabled properly.

Note that sensitive information such as passwords, and security credentials in unencrypted plain text will not be logged.

Viewing Generate and Deploy Time Logs and Service Processing Logs

To effectively troubleshoot or debug any error occurs at each stage of service development life cycle, logs can be viewed and downloaded for further analysis. Based on your log configuration and setups, the logging framework will filter the logs specifically for the design time and run time for the services that have logging enabled properly.

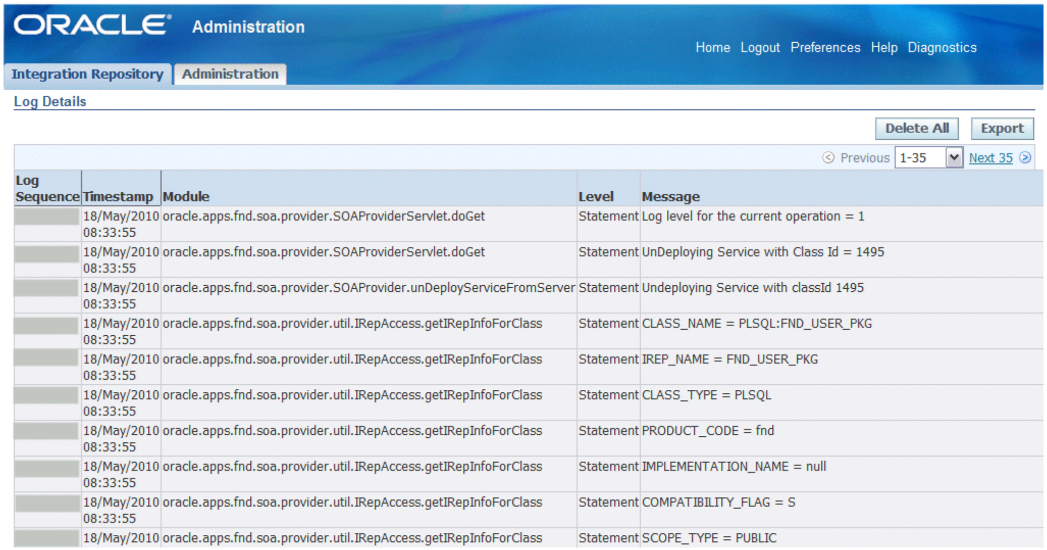
At design time during service generation and deployment, logs can be captured through the Integration Repository user interface if logging is enabled (at any log severity level) for specific services or all services at the Site level only. Administrators can find **View Log** in the Interface Details page for the services (or all services) that have logging enabled properly.

To view and download the log messages, click **View Log** to open the Log Details page where you can find a list of log messages compiled in a table.

Note that if logging is enabled for 'All Services' at the Site level, then **View Log** will be shown in the Interface Details page for all interfaces that can be service enabled. If the logging is enabled at the Site level for specific operations, then there will be no log generated and you will not find **View Log** in the Interface Details page. This type of log that is written during service generation and deployment is only available if the logging is enabled for specific services or all services at the Site level.

> **Note:** You will not find **View Log** available in the Interface Details page for a given service if the logging is enabled at the user level. Only site level logging configuration with specific services or all services will have Generate and Deploy time logs captured.

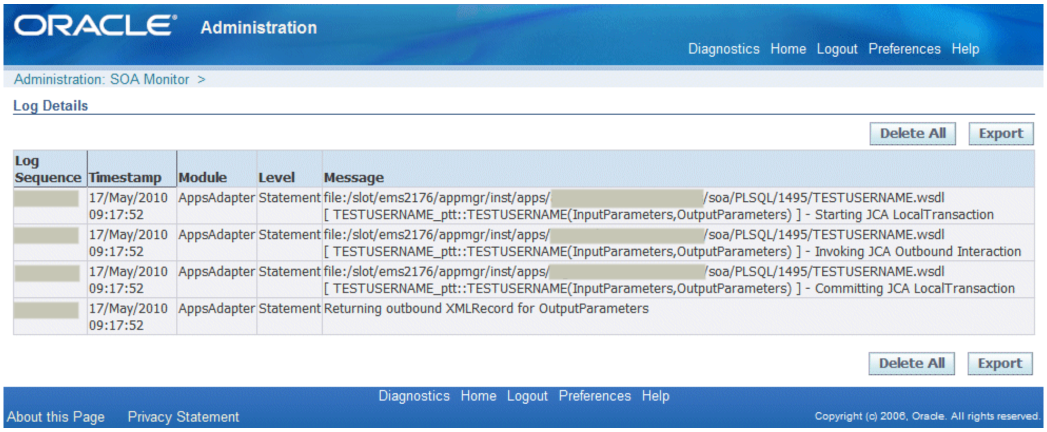
Log Details Page with Generate and Deploy Time Logs



Log Sequence	Timestamp	Module	Level	Message
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.SOAProviderServlet.doGet	Statement	Log level for the current operation = 1
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.SOAProviderServlet.doGet	Statement	UnDeploying Service with Class Id = 1495
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.SOAProvider.unDeployServiceFromServer	Statement	Undeploying Service with classId 1495
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	CLASS_NAME = PLSQL:FND_USER_PKG
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	IREP_NAME = FND_USER_PKG
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	CLASS_TYPE = PLSQL
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	PRODUCT_CODE = fnd
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	IMPLEMENTATION_NAME = null
[redacted]	18/May/2010 08:33:55	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	COMPATIBILITY_FLAG = S
[redacted]	18/May/2010	oracle.apps.fnd.soa.provider.util.IRepAccess.getIRepInfoForClass	Statement	SCOPE_TYPE = PUBLIC

At runtime during the invocation of Oracle E-Business Suite services by web service clients, if a service has logging enabled, the associated log messages are captured and can be viewed through SOA Monitor. If log messages are available for an instance, the **Log** icon appears in the search result table for that request in SOA Monitor. Click the **Log** icon to open the Log Details page where you can view logs recorded for the request against a specific instance.

Log Details Page for Service Processing Logs


ORACLE Administration
Diagnostics Home Logout Preferences Help
Administration: SOA Monitor >
Log Details
Delete All
Export
Delete All
Export
Diagnostics Home Logout Preferences Help
About this Page
Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

Log Sequence	Timestamp	Module	Level	Message
[redacted]	17/May/2010 09:17:52	AppsAdapter	Statement	file:/slot/ems2176/appmgr/inst/apps/[redacted]/soa/PLSQL/1495/TESTUSERNAME.wsdl [TESTUSERNAME_ptt::TESTUSERNAME(InputParameters,OutputParameters)] - Starting JCA LocalTransaction
[redacted]	17/May/2010 09:17:52	AppsAdapter	Statement	file:/slot/ems2176/appmgr/inst/apps/[redacted]/soa/PLSQL/1495/TESTUSERNAME.wsdl [TESTUSERNAME_ptt::TESTUSERNAME(InputParameters,OutputParameters)] - Invoking JCA Outbound Interaction
[redacted]	17/May/2010 09:17:52	AppsAdapter	Statement	file:/slot/ems2176/appmgr/inst/apps/[redacted]/soa/PLSQL/1495/TESTUSERNAME.wsdl [TESTUSERNAME_ptt::TESTUSERNAME(InputParameters,OutputParameters)] - Committing JCA LocalTransaction
[redacted]	17/May/2010 09:17:52	AppsAdapter	Statement	Returning outbound XMLRecord for OutputParameters

Deleting and Exporting Logs Listed in the Log Details Page

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

Note: Log records deleted here are instance specific, whereas the Purge program from the SOA Monitor requiring you enter specific date range in executing the concurrent request is not. The purge concurrent request through SOA Monitor 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 SOA Monitor, see Purging SOAP Messages, Audits, and Logs, page 8-10.

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 page to Microsoft Excel and save them to a designated directory and use it later.

For more information on viewing logs recorded during service generation and deployment through Integration Repository, see Viewing Generate and Deploy Time Logs, page 3-16.

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 SOA Monitor

This chapter covers the following topics:

- SOA Monitor Overview
- Searching SOAP Requests
- Viewing SOAP Request and Response Details
- Viewing Service Processing Logs
- Purging SOAP Messages, Audits, and Logs
- Enabling and Disabling Web Service Auditing

SOA Monitor Overview

SOA Monitor is a centralized, light-weight service execution monitoring and management tool. It not only monitors all the web service activities that SOA Provider and Web Service Provider process, but also provides auditing records for the service execution details if the auditing feature is enabled.

> **Note:** Only SOAP services are monitored and audited through SOA Monitor. Runtime REST service monitoring and auditing features are not supported in this release.

For the monitoring purpose, SOA Monitor stores basic information about a 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 unless the auditing feature is turned on.

When the auditing feature is enabled, SOA Monitor then saves SOAP request and response payloads, 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 occur.

> **Important:** To enable the SOA Monitor auditing feature, you must set the profile option "SOA: Web Service Audit".
>
> In this release, SOA Monitor is a permanent monitor tool and it is enabled at all times to monitor all web services.

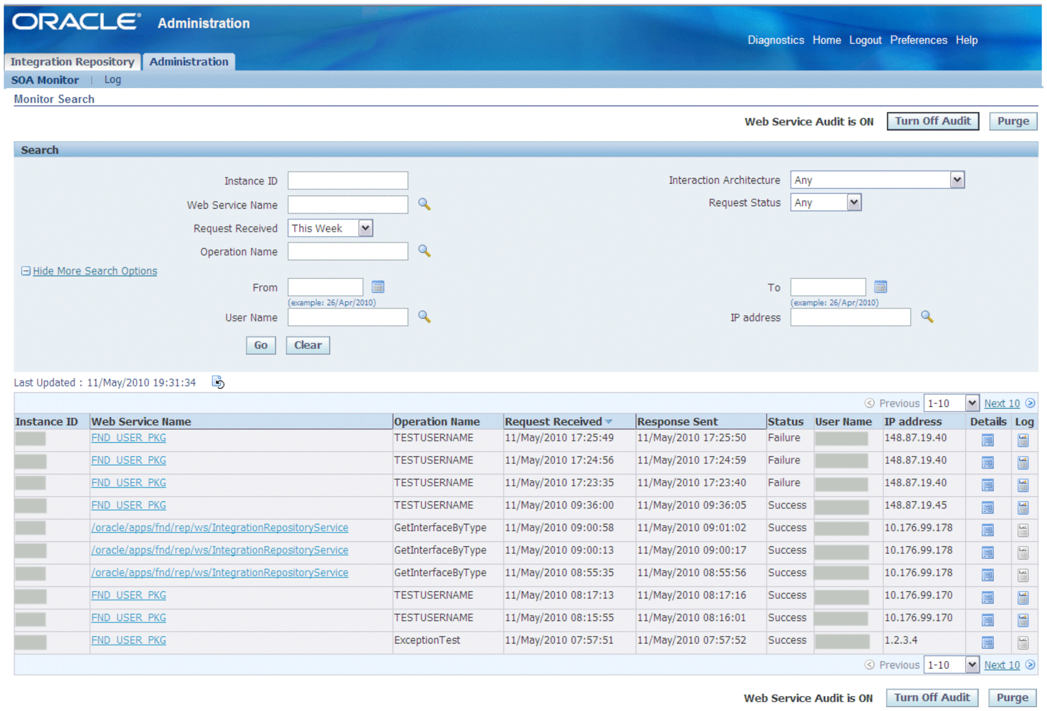
Accessing SOA Monitor

To access SOA Monitor, log on to Oracle E-Business Suite as a user who has the Integration Repository Administrator role.

Select the **Integrated SOA Gateway responsibility** from the navigation menu and then select **SOA Monitor** link from the Administration section. This opens the SOA Monitor Search page under the Administration tab.

> **Note:** Only users who have the Integration Repository Administrator role can find the Administration section available 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 now grouped under the Administration section and displayed in the Administration tab. These tasks include managing log setups in the Log subtab and managing SOAP requests in the SOA Monitor subtab.

Monitor Search Page


ORACLE Administration
Diagnostics Home Logout Preferences Help
Integration Repository Administration
SOA Monitor | Log
Monitor Search

Web Service Audit is ON [Turn Off Audit] [Purge]

Search

Instance ID:
Web Service Name:
Request Received: This Week
Operation Name:
Interaction Architecture: Any
Request Status: Any

⊟ Hide More Search Options

From: (example: 26/Apr/2010)
To: (example: 26/Apr/2010)
User Name:
IP address:

[Go] [Clear]

Last Updated : 11/May/2010 19:31:34

Previous 1-10 Next 10

Instance ID	Web Service Name	Operation Name	Request Received	Response Sent	Status	User Name	IP address	Details	Log
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 17:25:49	11/May/2010 17:25:50	Failure	[redacted]	148.87.19.40	▤	▤
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 17:24:56	11/May/2010 17:24:59	Failure	[redacted]	148.87.19.40	▤	▤
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 17:23:35	11/May/2010 17:23:40	Failure	[redacted]	148.87.19.40	▤	▤
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 09:36:00	11/May/2010 09:36:05	Success	[redacted]	148.87.19.45	▤	▤
[redacted]	/oracle/apps/fnd/rep/ws/IntegrationRepositoryService	GetInterfaceByType	11/May/2010 09:00:58	11/May/2010 09:01:02	Success	[redacted]	10.176.99.178	▤	▤
[redacted]	/oracle/apps/fnd/rep/ws/IntegrationRepositoryService	GetInterfaceByType	11/May/2010 09:00:13	11/May/2010 09:00:17	Success	[redacted]	10.176.99.178	▤	▤
[redacted]	/oracle/apps/fnd/rep/ws/IntegrationRepositoryService	GetInterfaceByType	11/May/2010 08:55:35	11/May/2010 08:55:56	Success	[redacted]	10.176.99.178	▤	▤
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 08:17:13	11/May/2010 08:17:16	Success	[redacted]	10.176.99.170	▤	▤
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 08:15:55	11/May/2010 08:16:01	Success	[redacted]	10.176.99.170	▤	▤
[redacted]	FND_USER_PKG	ExceptionTest	11/May/2010 07:57:51	11/May/2010 07:57:52	Success	[redacted]	1.2.3.4	▤	▤

Previous 1-10 Next 10

Web Service Audit is ON [Turn Off Audit] [Purge]

Integration repository administrators can perform the following activities through SOA 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 8-10
- Enabling and Disabling Web Service Auditing, page 8-14

Searching SOAP Requests

In the Search region, you can perform searches on SOAP requests processed through SOA Provider based on the criteria you specified.

SOA Monitor allows you to search SOAP requests by instance ID, interaction architecture, request status, web service name, operation name, and request received time.

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 including username, IP address, and a selected time frame for your search if clicking the **Show More Search Options** link in the Search region.

When the search is executed, all entries that match your search criteria will be retrieved and displayed in a tabular format. This information includes the instance ID, web service name, operation name, date and time the request was received and responded, username, IP address, and request and response statuses.

If 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 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 log message details by clicking the **Log** icon if they 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 8-10.

- Enable or Disable Web Service Auditing feature by clicking **Turn On Audit** or **Turn Off Audit**

  The Web Service Audit feature that is currently turned on or not can be found next to the **Turn On Audit** or **Turn Off Audit** in the Search Monitor page.

See: Enabling and Disabling Web Service Auditing, page 8-14.

To perform a search:

1. Log on to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility. From the navigation menu, select the SOA 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 architecture, 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.

   - IP Address: Select an appropriate value from the list of values.

   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 Request and Response details.

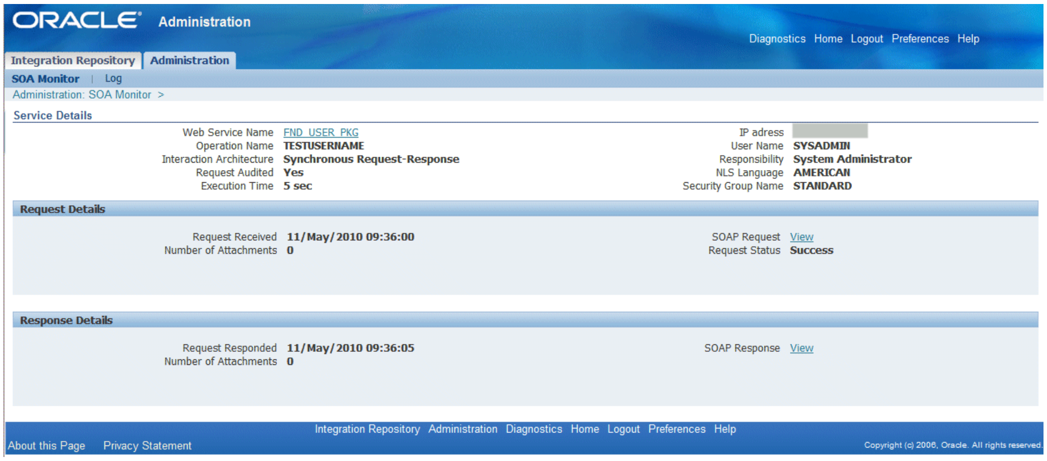
6. Click the **Log** icon if logs are available for a given instance ID to view log details.

7. Click **Purge** to purge SOAP requests and responses, audits, as well as log messages collected for certain period of time.

Viewing SOAP Request and Response Details

After executing a search, all SOAP messages that match your criteria will be retrieved. You can view the SOAP request and response details by clicking 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
Operation Name TESTUSERNAME
Interaction Architecture Synchronous Request-Response
Request Audited Yes
Execution Time 5 sec

IP adress [redacted]
User Name SYSADMIN
Responsibility System Administrator
NLS Language AMERICAN
Security Group Name STANDARD

Request Details

Request Received 11/May/2010 09:36:00
Number of Attachments 0
SOAP Request View
Request Status Success

Response Details

Request Responded 11/May/2010 09:36:05
Number of Attachments 0
SOAP Response View

Integration Repository Administration Diagnostics Home Logout Preferences Help
About this Page Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

General SOAP request heading is displayed at the top of the page. This header information includes Web service name, operation name, interaction architecture, host IP address, username, responsibility, NLS language, security group name, execution time, and whether the request is audited or not.

By clicking the 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 if you want.

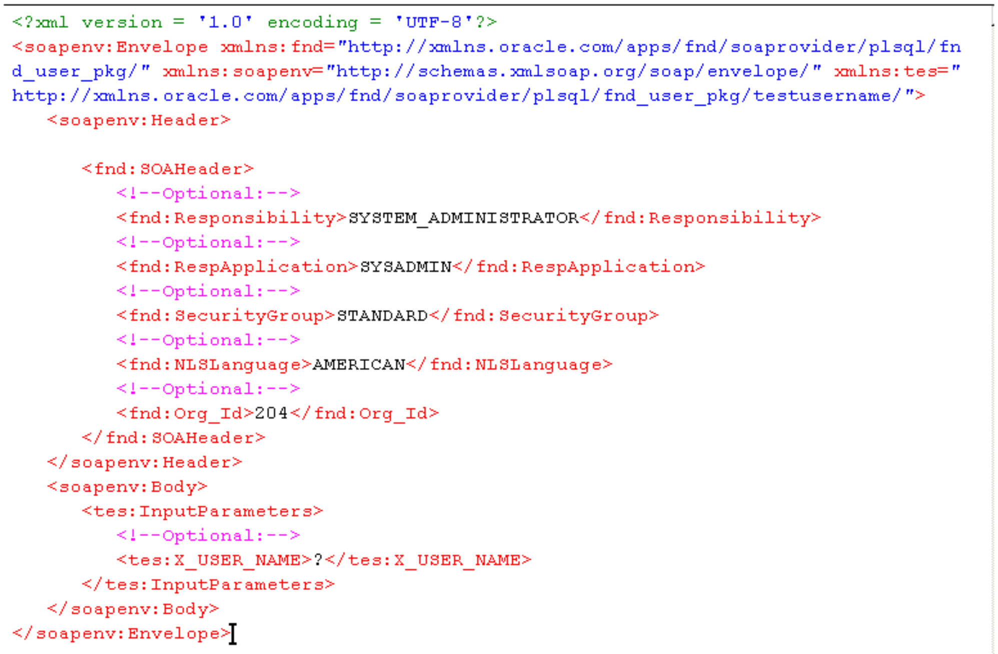
In addition to the general header, the following regions are displayed in the details page:

- **Request Details:** This region contains the SOAP request received date and time, number of attachments, request status, and the SOAP request payload view link.

  Clicking 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 'SOA: Web Service Audit' profile was turned on for enabling Web Service auditing feature. If the Web Service Auditing feature was turned off at the time of processing that request, **View** link will not appear. The same theory applies to process SOAP responses as well.

SOAP Request XML File



```
<?xml version = '1.0' encoding = 'UTF-8'?>
<soapenv:Envelope xmlns:fnd="http://xmlns.oracle.com/apps/fnd/soaprovider/plsql/fn
d_user_pkg/" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tes="
http://xmlns.oracle.com/apps/fnd/soaprovider/plsql/fnd_user_pkg/testusername/">
   <soapenv:Header>

      <fnd:SOAHeader>
         <!--Optional:-->
         <fnd:Responsibility>SYSTEM_ADMINISTRATOR</fnd:Responsibility>
         <!--Optional:-->
         <fnd:RespApplication>SYSADMIN</fnd:RespApplication>
         <!--Optional:-->
         <fnd:SecurityGroup>STANDARD</fnd:SecurityGroup>
         <!--Optional:-->
         <fnd:NLSLanguage>AMERICAN</fnd:NLSLanguage>
         <!--Optional:-->
         <fnd:Org_Id>204</fnd:Org_Id>
      </fnd:SOAHeader>
   </soapenv:Header>
   <soapenv:Body>
      <tes:InputParameters>
         <!--Optional:-->
         <tes:X_USER_NAME>?</tes:X_USER_NAME>
      </tes:InputParameters>
   </soapenv:Body>
</soapenv:Envelope>
```

Additionally, you can find the following information displayed 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 to show the error description and details.

- **Attachment:** If the SOAP request has attachments associated with it, the Attachment region will appear to list the attachment details including all attachment names and MIME Type information.

- **Response Details:** This region contains the SOAP request responded date and time, number of attachments, and the SOAP response view link.

  Clicking 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 'SOA: Web Service Audit' profile was turned on for enabling Web Service auditing feature. If the Web Service Auditing feature was turned off at the time of processing that response, the **View** link will not appear. The same theory applies to process SOAP requests as well.
  >
  > Additionally, if the Interaction Architecture 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 as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility.

   From the navigation menu, select the SOA Monitor link from the Administration section to open the Monitor Search page.

2. Perform a search to display search results. See: Searching SOAP Requests, page 8-3.

3. Click the **Details** icon for a given request to view SOAP request and response details. The Request and Response Details page appears allowing you to view the request and response details.

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, you can find attachment information listed in the Attachment region.

6. If this request status is 'Failed', then you will find the error details in the Error Information region.

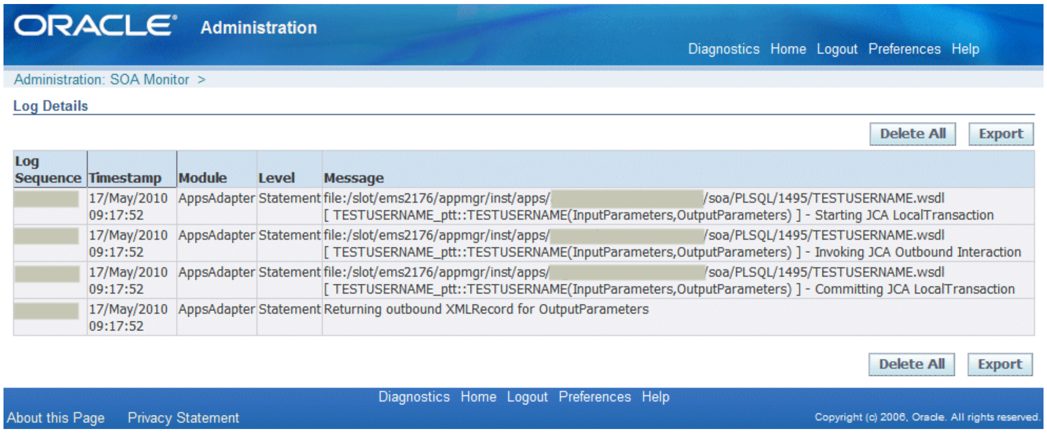
Viewing Service Processing Logs

To effectively monitor SOAP messages at run time during the invocation of Oracle E-Business Suite services by web service clients, log messages can be captured in SOA Monitor against that instance for the services or operations if logging is enabled regardless of the configuration set at the user or site level.

When a SOAP request is received, SOA Provider generates a unique numeric instance ID based on a database sequence and passes it to SOA Monitor. Therefore, each SOAP request in SOA Monitor appears with instance ID and the **Log** icon letting you retrieve the log details.

By clicking the **Log** icon in the search result table if a log is available for a given instance, you can view log messages in the Log Details page.

Log Details Page



Log Sequence	Timestamp	Module	Level	Message
[redacted]	17/May/2010 09:17:52	AppsAdapter	Statement	file:/slot/ems2176/appmgr/inst/apps/[redacted]/soa/PLSQL/1495/TESTUSERNAME.wsdl [TESTUSERNAME_ptt::TESTUSERNAME(InputParameters,OutputParameters)] - Starting JCA LocalTransaction
[redacted]	17/May/2010 09:17:52	AppsAdapter	Statement	file:/slot/ems2176/appmgr/inst/apps/[redacted]/soa/PLSQL/1495/TESTUSERNAME.wsdl [TESTUSERNAME_ptt::TESTUSERNAME(InputParameters,OutputParameters)] - Invoking JCA Outbound Interaction
[redacted]	17/May/2010 09:17:52	AppsAdapter	Statement	file:/slot/ems2176/appmgr/inst/apps/[redacted]/soa/PLSQL/1495/TESTUSERNAME.wsdl [TESTUSERNAME_ptt::TESTUSERNAME(InputParameters,OutputParameters)] - Committing JCA LocalTransaction
[redacted]	17/May/2010 09:17:52	AppsAdapter	Statement	Returning outbound XMLRecord for OutputParameters

The Log Details page contains all the log messages recorded for the selected service in a given instance. These log messages are compiled and listed in the table format including log sequence, log timestamp, module, severity level, and actual message.

Deleting and Exporting Logs Listed in the Log Details Page for a Specific Instance

After viewing log messages retrieved for a request in the Log Details page for a given instance, you can delete them if needed by clicking **Delete All**. A warning message appears alerting you that this will permanently delete all the logs retrieved in the page. Click **Yes** to confirm the action. An empty log table appears in the page after logs are 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 page to Microsoft Excel and save them to a designated directory and use it later.

For information on log severity level and how to configure logs, see Adding a New Logging Configuration, page 7-5.

For information on how to view logs recorded during service generation and deployment at design time, see Viewing Generate and Deploy Time Logs, page 3-16.

To view log messages in SOA Monitor:

1. Log on to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility.

   From the navigation menu, select the SOA Monitor link from the Administration section to open the Monitor Search page.

2. Perform a search to display search results. See: Searching SOAP messages, page 8-3.

3. In the search result table, click the **Log** icon for a desired instance. The Log Details page is displayed allowing you to view the log details.

4. Click **Delete All** 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 back to the Log Details page.

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

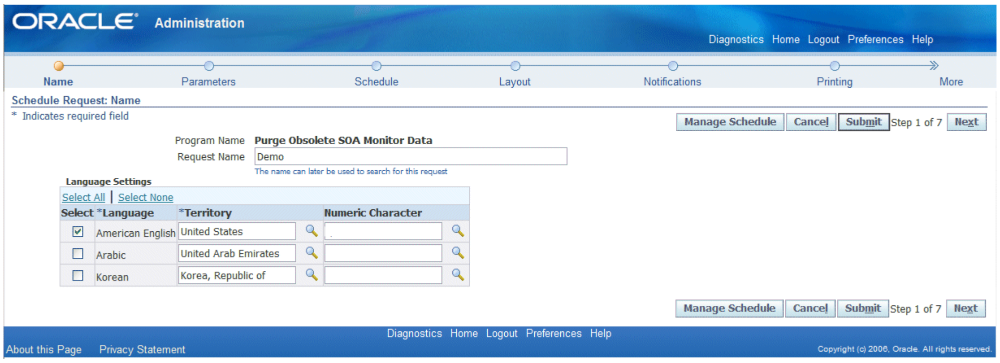
Purging SOAP Messages, Audits, and Logs

Oracle E-Business Suite Integrated SOA Gateway allows you to purge SOAP messages, logs, and audit records that have been collected through SOA Monitor for a period of time. Click **Purge** in the SOA Monitor Search page to launch a concurrent program Purge Obsolete SOA Monitor Data (FNDSOA_PURGE).

You will need to enter relevant purge parameters in the following Schedule Request windows including start and end dates before submitting the purge.

- **Name**: The concurrent program name 'SOA Purge Audit Data' is displayed automatically. Specify the Request Name for your request. You can also change the default language setting if necessary.

Purge Schedule Request: Name Window



Schedule Request: Name

* Indicates required field

Program Name **Purge Obsolete SOA Monitor Data**

Request Name Demo

The name can later be used to search for this request

Language Settings

Select	*Language	*Territory	Numeric Character
☑	American English	United States	
□	Arabic	United Arab Emirates	
□	Korean	Korea, Republic of	

Before clicking **Next** to proceed to the next step, you can click **Manage Schedule** to open the Schedule page where you can search existing schedules or create a new schedule.

- Search Existing Schedules: In the Simple Search region, enter appropriate search criteria including schedule name, application name, and created by. Click **Go** to execute your search.

  The schedules that match your criteria will be displayed in the results table. The schedule owner or creator can update and delete the schedule that she or he created before.

  Click **Create** to create a new schedule or click **Create Like** to create a schedule similar to a selected a schedule.

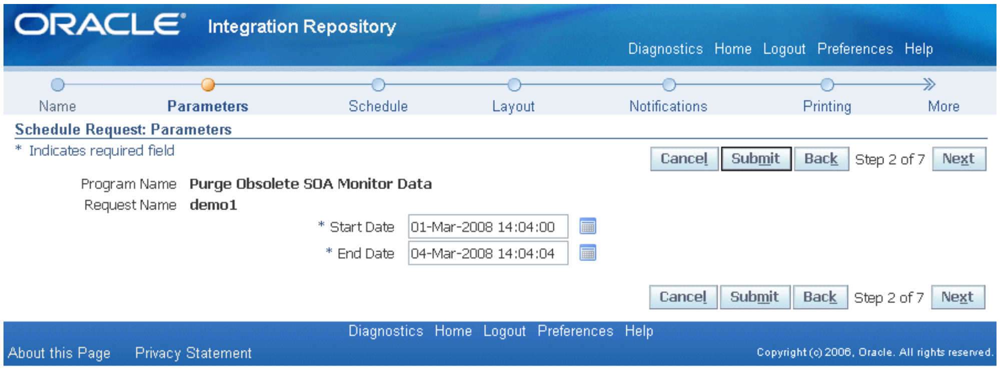
- Create a New Schedule: Click **Create** to open the Schedule page allowing you to define a new schedule with schedule name and description in either one of the following options:
  - A simple schedule with a specific date and time, or recurring intervals if it is a recurring schedule.
  - An advanced schedule by clicking **Advanced Schedule** to specify more scheduling information, such as specific days of the week or specific dates in a month.

    Note: If you do not select an end date for the more advanced schedules, the request will continue to run until it is cancelled.

  Click **Apply** to save your record.

- **Parameters**: You must enter the Start Date and End Date fields to identify the date range for your purge.

Purge Schedule Request: Parameters Window



Schedule Request: Parameters

* Indicates required field

Program Name Purge Obsolete SOA Monitor Data

Request Name demo1

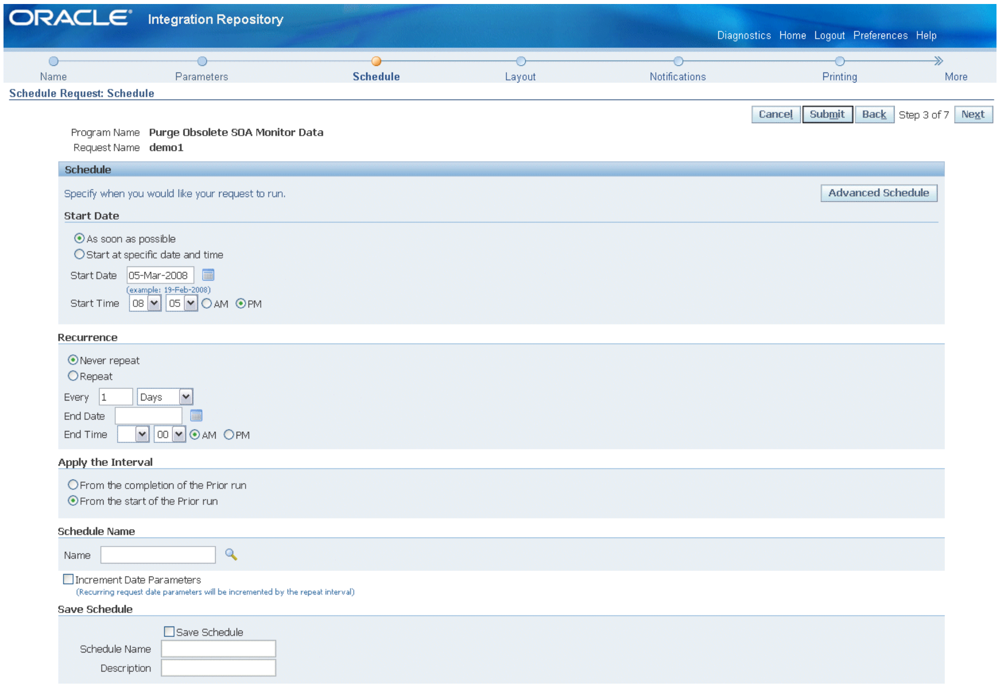
* Start Date 01-Mar-2008 14:04:00

* End Date 04-Mar-2008 14:04:04

Copyright (c) 2006, Oracle. All rights reserved.

- **Schedule**: Specify when you would like your request to be run.

Purge Schedule Request: Schedule Window



ORACLE Integration Repository
Diagnostics Home Logout Preferences Help
Name
Parameters
Schedule
Layout
Notifications
Printing
More

Schedule Request: Schedule

Program Name Purge Obsolete SOA Monitor Data
Request Name demo1

Schedule

Specify when you would like your request to run.

Start Date

● As soon as possible
○ Start at specific date and time

Start Date 05-Mar-2008
(example: 19-Feb-2008)
Start Time 08 05 ○ AM ● PM

Recurrence

● Never repeat
○ Repeat

Every 1 Days
End Date
End Time 00 ● AM ○ PM

Apply the Interval

○ From the completion of the Prior run
● From the start of the Prior run

Schedule Name

Name

□ Increment Date Parameters
(Recurring request date parameters will be incremented by the repeat interval)

Save Schedule

□ Save Schedule
Schedule Name
Description

You can choose the following options:

- A simple schedule such as, as soon as possible, a specific date and time, or recurring intervals if it is a recurring schedule.

- An advanced schedule by clicking **Advanced Schedule** to specify more scheduling information, such as specific days of the week or specific dates in a month. You can also choose a previously saved schedule.

  Note: If you do not select an end date for the more advanced schedules, the request will continue to run until it is cancelled.

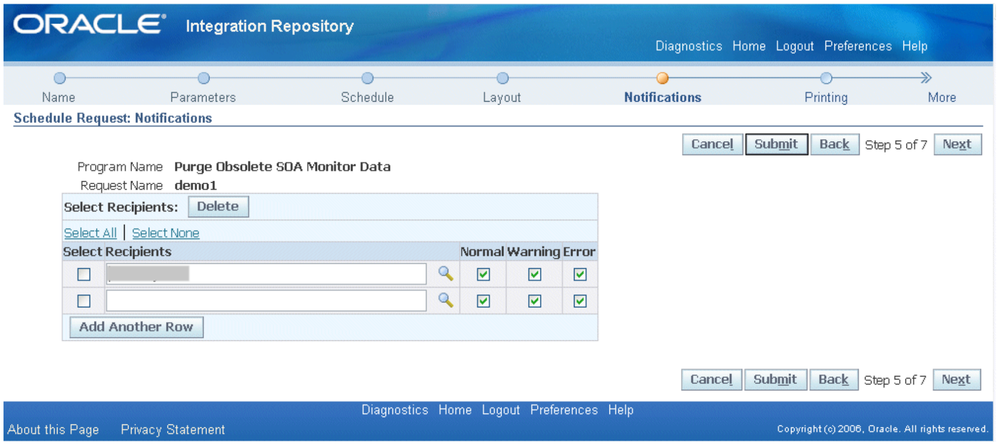
Click the **Increment Date Parameters** check box to make the selected schedule become repeatable with the recurrence intervals specified in the Recurrence region.

Instead of creating a new schedule by default, you can use a previously saved schedule by clicking the **Saved Schedule** radio button on the top of the Schedule page. A new page is displayed letting you specify a desired schedule name in the Name field. The selected schedule details will be populated automatically.

- **Layout**: This allows you to select layout based on a template. You can also specify the output format for your request.

- **Notifications**: Select the employee name from the list of available employees, and then choose the circumstance of when to notify this employee. This option sends an e-mail notification with a link to the request, based on if the request ran normally or resulted in a warning or error.

Purge Schedule Request: Notifications Window


ORACLE Integration Repository
Diagnostics Home Logout Preferences Help
Name
Parameters
Schedule
Layout
Notifications
Printing
More
Schedule Request: Notifications
Cancel
Submit
Back
Step 5 of 7
Next
Select Recipients: Delete
Select All | Select None
Add Another Row
About this Page
Privacy Statement
Copyright (c) 2006, Oracle. All rights reserved.

Program Name **Purge Obsolete SOA Monitor Data**
Request Name **demo1**

Select Recipients	Normal	Warning	Error
□ [redacted]	☑	☑	☑
□	☑	☑	☑

- **Printing**: For printed output, select the printer, copies, and print style.

- **Preview**: This allows you to preview all your parameter selection for the purge request.

Once you submit the purge request, a concurrent request number is automatically assigned. Your request will be executed based on your selected schedule to purge all SOAP requests during your specified date range.

The monitored SOAP requests and responses stored in the following tables will be purged in the following order of sequence:

1. Purging FND_SOA_REQUEST

   This deletes all SOAP requests for the specified date range.

2. Purging FND_SOA_BODY_PIECE

   This deletes the SOAP body pieces including payload corresponding to those SOAP requests that have been purged (for the specified date range).

3. Purging FND_SOA_ATTACHMENT

   This deletes all attachments associated with the SOAP requests and responses for the specified date range.

4. Purging FND_LOG_MESSAGES

   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.

To purge SOAP requests and responses:

1. Log on to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility.

   From the navigation menu, select the SOA Monitor link from the Administration section to open the Monitor Search page.

2. Click **Purge** to launch a concurrent program.

3. Enter the following information in the Schedule Request window:

   1. The concurrent program name SOA Purge Audit Data is displayed in the Program Name field. You can specify the request name for your purge request.

      Click **Next**.

   2. Enter the Start Date and End Date fields to specify the time range for your purge. Click **Next**.

   3. Enter appropriate information for the Schedule window. Click **Next**.

   4. Specify notification information by selecting employee names and the circumstances when the notifications will be sent. Click **Next**.

   5. Leave the default printing information unchanged if you do not want it to be printed. Click **Next**

   6. Preview your purge request selection.

4. Click **Submit** to submit your purge request.

   A request number will be automatically assigned to you for your purge request indicating your request has been submitted for processing.

Enabling and Disabling Web Service Auditing

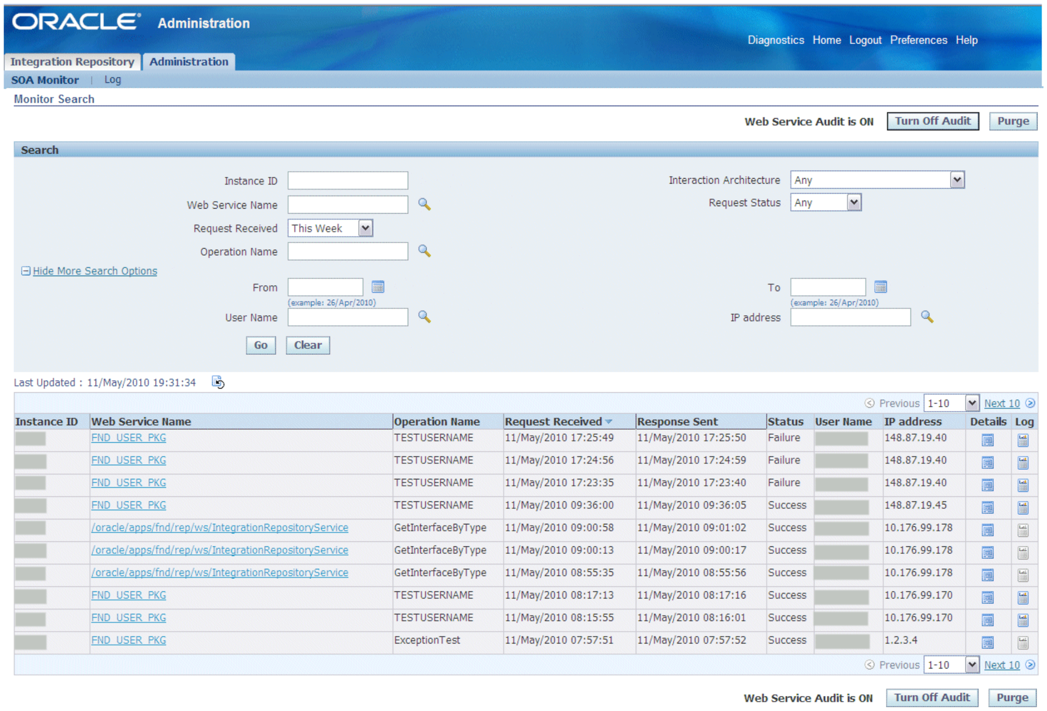
In addition to monitoring and providing runtime status of all service execution activities, SOA Monitor provides auditing feature allowing you to track SOAP message details such as requests, responses, faults, and so on.

If this auditing feature is enabled, then all incoming SOAP requests and corresponding responses that SOA Provider receives as well as the associated payloads and fault messages can be saved in SOA Monitor for auditing needs.

In addition to setting the profile option, integration repository administrators can enable or disable the feature directly in the SOA Monitor UI. By clicking **Turn On Audit** or **Turn Off Audit** in the Monitor Search page will override the 'SOA: Web Service Audit' profile value to enable or disable the feature.

The Web Service Audit is ON or OFF information appears indicating the feature is enabled or disabled.

Monitor Search Page with Web Service Auditing Information



Monitor Search

Web Service Audit is ON Turn Off Audit Purge

Search

Instance ID
Web Service Name
Request Received: This Week
Operation Name
Interaction Architecture: Any
Request Status: Any

⊟ Hide More Search Options

From (example: 26/Apr/2010)
To (example: 26/Apr/2010)
User Name
IP address

Go Clear

Last Updated : 11/May/2010 19:31:34

Previous 1-10 Next 10

Instance ID	Web Service Name	Operation Name	Request Received	Response Sent	Status	User Name	IP address	Details	Log
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 17:25:49	11/May/2010 17:25:50	Failure	[redacted]	148.87.19.40	▤	▤
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 17:24:56	11/May/2010 17:24:59	Failure	[redacted]	148.87.19.40	▤	▤
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 17:23:35	11/May/2010 17:23:40	Failure	[redacted]	148.87.19.40	▤	▤
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 09:36:00	11/May/2010 09:36:05	Success	[redacted]	148.87.19.45	▤	▤
[redacted]	/oracle/apps/fnd/rep/ws/IntegrationRepositoryService	GetInterfaceByType	11/May/2010 09:00:58	11/May/2010 09:01:02	Success	[redacted]	10.176.99.178	▤	▤
[redacted]	/oracle/apps/fnd/rep/ws/IntegrationRepositoryService	GetInterfaceByType	11/May/2010 09:00:13	11/May/2010 09:00:17	Success	[redacted]	10.176.99.178	▤	▤
[redacted]	/oracle/apps/fnd/rep/ws/IntegrationRepositoryService	GetInterfaceByType	11/May/2010 08:55:35	11/May/2010 08:55:56	Success	[redacted]	10.176.99.178	▤	▤
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 08:17:13	11/May/2010 08:17:16	Success	[redacted]	10.176.99.170	▤	▤
[redacted]	FND_USER_PKG	TESTUSERNAME	11/May/2010 08:15:55	11/May/2010 08:16:01	Success	[redacted]	10.176.99.170	▤	▤
[redacted]	FND_USER_PKG	ExceptionTest	11/May/2010 07:57:51	11/May/2010 07:57:52	Success	[redacted]	1.2.3.4	▤	▤

Previous 1-10 Next 10

Web Service Audit is ON Turn Off Audit Purge

To enable or disable the Web Service Auditing feature:

1. Log on to Oracle E-Business Suite as a user who has the Integration Repository Administrator role. Select the Integrated SOA Gateway responsibility.

   From the navigation menu, select the SOA Monitor link from the Administration section to open the Monitor Search page.

2. Click **Turn On Audit** or **Turn Off Audit** in the Monitor Search page to enable or disable the feature.

3. The Web Service Audit is ON or OFF information appears indicating the feature is enabled or disabled.

9

Implementing Service Invocation Framework

This chapter covers the following topics:

- Overview
- Implementing Service Invocation Framework

Overview

To invoke all integration 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 any SOAP service, described in WSDL, to be invoked.

By using this service invocation framework, developers or implementors can interact with SOAP services through WSDL descriptions. This approach lets developers or implementors use WSDL as a normalized description of disparate software, and it allows them to access this software in a manner that is independent of protocol or location.

Because this invocation framework allows updated implementations of a binding to be plugged into WSIF at runtime, 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 runtime. More importantly, this enhances the seamless business integration between loosely-coupled applications and accelerates service execution and consumption.

> **Note:** WSIF is a simple Java API for invoking web services. It is supported by Oracle Application Server 10g Release 3 (10.1.3) which is shipped together with Oracle E-Business Suite Release 12. To upgrade your instance from Release 12, ensure that your system is upgraded to appropriate versions of Oracle Application Server 10g. See *Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12*, My Oracle Support Knowledge Document 556540.1 for details.

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 typically 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 9-2
- Understanding Service Invocation Framework Major Features, page 9-5
- Implementing Service Invocation Framework, page 9-6

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.

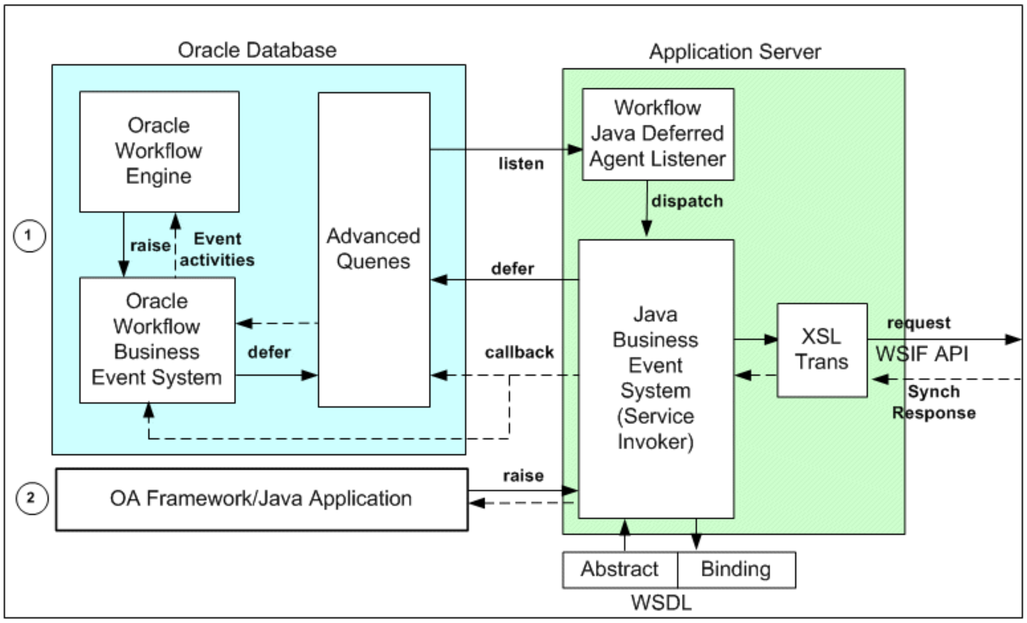
> **Note:** By leveraging Oracle Workflow Java Business Event System (JBES), service invocation framework allows almost any forms of Web services representing in WSDL URLs to be invoked out from Oracle E-Business Suite.

At run time, when an invoker event is raised, the event and subscription parameters are used to invoke Web services.

> **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.

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 within Oracle Workflow process:

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 PL/SQL or Java.

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 Manger within the call to the `WF_EVENT.Raise` API, or the Event Manger 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 *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 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 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 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 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 (that uses WSIF to invoke the Web service) within `OACORE OC4J container`.

After an event is raised either using 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 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 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 be then 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 successful the response is available immediately by using method `getResponseData()` on the `BusinessEvent` object.

Service Invocation Framework Major Features

Service Invocation Framework includes the following features:

- It supports various service invocation sources or points from 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 application context into SOAP envelopes.
- It provides errors and exception handling, and the invocation retry feature.
- It provides a ability to test business event for service invocation.

Implementing Service Invocation Framework

This section discusses the following topics:

- Setup Tasks, page 9-6
- Setup Tasks for Invoking SSL-based Web Services Over HTTPS, page 9-8
- Implementing Service Invocation Framework, page 9-13

Setup Tasks

Web services can be invoked from any one of following tiers:

- **OACORE OC4J**: Web service invocations from OA Framework page using a synchronous event subscription (phase < 100) is executed from within the OACORE OC4J container.
- **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 executes from within that JVM.

Proxy Host and Port Setups

In most cases, the Web service resides outside the firewall and the executing host does not have direct access to the WSDL or the Web service endpoint to send the SOAP request. Therefore, administrators must set up and configure proxy host and port appropriately for the tiers that Web service invocations occur in order to perform following activities:

- Parse and consume WSDL during subscription definition
- Invoke Web service from subscription definition

Setting Up Proxy Host and Port at OC4J Tier

For Web services invoked from OA Framework, the JBES seeded Java rule function would run within OACORE's OC4J container.

The oc4j.properties (`$INST_TOP/ora/10.1.3/j2ee/oacore/oc4j.properties`) should have the following properties or proxy values in order for it to work:

```
http.proxyHost=myproxyhost
```

```
http.proxyPort=80
```

To update the `oc4j.properties` file, you need to update AutoConfig context file with following entries and run AutoConfig:

```
<!-- proxy -->
  <proxyhost oa_var="s_proxyhost">myproxyhost</proxyhost>
   <proxyport oa_var="s_proxyport">80</proxyport>
   <porxybypassdomain oa_var="s_proxybypassdomain">any domain that needs
to be by-passed (such as *.example.com)</porxybypassdomain>
```

Setting Up Proxy Host and Port at Concurrent Manger (CM) Tier JVM

For Web services invoked from PL/SQL and Java using asynchronous subscriptions, the event is raised by the application code wherever it executes and then it is enqueued to 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 a concurrent manager tier Java service's JVM. If the Web service resides outside the firewall, it requires updating the following Service Parameters for Workflow Agent Listener Service from Oracle Workflow Manager available through Oracle Applications Manager:

- `SVC_PROXY_SET=true`
- `SVC_PROXY_HOST=<proxy_host>`
- `SVC_PROXY_PORT=<proxy_port>`

For detailed information, see Editing Service Parameters for a Container, *Oracle Workflow Administrator's Guide*.

Setting Up Proxy Host and Port When Using Standalone Java Class

You must set the following entries:

```
java -Dhttp.proxyHost=myproxyhost -Dhttp.proxyPort=80 class name
```

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 securely 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), some manual setup tasks need to be performed properly to read SSL-based WSDLs and invoke SSL service endpoints.

A client may receive one of following two types of server certificates to verify:

- Public certificate and it is issued by a Certification Authority (CA).

- Self-signed certificate or certificate is not in trusted certificate list.

Following two setups are required for the service invocation framework to invoke a SSL-based Web service:

- Import Server SSL Certificate into SIF's JVM Certificate Store, page 9-8

- Setup SSL Proxy Host and Port, page 9-11

- Performing Additional Setup Tasks, page 9-12

Importing Server SSL Certificate into SIF's JVM Certificate Store

Public Certificate Issued by a Certification Authority (CA)

If server certificate is a public certificate and issued by a public CA such as VeriSign, then it is most likely available in a SIF's JVM's certificate store or in a trusted certificate list.

Self-signed Certificate or Certificate is not in Trusted Certificate List

Complete the following tasks to import the server's SSL certificate into a SIF's 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.example.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.example.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 is successfully loaded in a browser, double click on the

Lock icon on the bottom right hand 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** and then select the Personal (or Other People) tab to select your certificate and click **Export**.

3. You can export or save 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 9-6.

There are many utilities available to import a certificate. 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 can 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, for example my_cert.cer.

Setting Up SSL Proxy Host and Port

If the 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 OC4J Tier

For Web services invoked from OA Framework, the JBES seeded Java rule function would run within OACORE's OC4J container.

The `oc4j.properties` (`$INST_TOP/ora/10.1.3/j2ee/oacore/config/oc4j.properties`) should have the following properties in order for it to work:

- `https.proxyHost=<proxyhost>`
- `https.proxyPort=<sslproxyport>`

> **Note:** The default `https` port 443.

AutoConfig does not support properties `https.proxyHost` and `https.proxyPort` currently. If the above properties are added to `oc4j.properties` manually, subsequent AutoConfig run will remove these two properties. In order to make sure the above properties are retained during AutoConfig run, the context file could be customized to add these two properties.

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 (CM) Tier JVM

For Web services invoked from PL/SQL and Java using asynchronous subscriptions, the event is raised by the application code wherever it executes and then it is enqueued to 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 a concurrent manager 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]
<class name>
```

Performing Additional Setup Tasks

Additionally, performing the following tasks to invoke services with TLS 1.2 only and TLS 1.2 with backward compatibility:

1. Apply Patch 22612527 with prerequisite Patch 13866584 to the FMW home (`FMW_HOME`).

2. Update the 32-bit JDK 7 under `$OA_JRE_TOP` with the Java Cryptography Extension (JCE) updates from the following page: https://www.oracle.com/java/technologies/javase-jce7-downloads.html

3. Update the 64-bit JDK 7 under the directory referenced by the `s_fmw_jdktop` context variable with the Java Cryptography Extension (JCE) updates.

4. Update the Oracle E-Business Suite context variables using Oracle Applications Manager.

    1. Log in to Oracle E-Business Suite as a user who has the **Workflow Administrator Web Applications** responsibility.

    2. Select the **Oracle Applications Manager** link from the Navigator, and then select **AutoConfig**.

    3. Select the application tier context file, and choose **Edit Parameters**.

    4. Update the following context variables:

        - `s_afjsmarg = -Dhttps.protocols=TLSv1,TLSv1.1,TLSv1.2 or -Dhttps.protocols=TLSv1.2`

            - To enable TLS 1.2 with backward compatibility, add the following:

              `s_afjsmarg = -Dhttps.protocols=TLSv1,TLSv1.1,TLSv1.2`

            - To enable TLS 1.2 only, add the following:

              `s_afjsmarg = -Dhttps.protocols=TLSv1.2`

        - `s_proxyhost = fully qualified host.domain name`

- s_proxyport = port value
- s_proxybypassdomain = domain name (For example, example.com)
- s_nonproxyhosts = wildcard domain name (For example, *.example.com)

5. Run AutoConfig using the adautocfg.sh script in the application tier $ADMIN_SCRIPTS_HOME directory.

6. Run the adstpall.sh script and the adstrtal.sh script in the same $ADMIN_SCRIPTS_HOME directory to stop and restart all services.

For more information about enabling TLS in Oracle E-Business Suite Release 12.1, see My Oracle Support Knowledge Document 376700.1.

Implementing Service Invocation Framework

As mentioned earlier, service invocation framework, leveraging Oracle Workflow Business Event System and a seeded Java rule function, oracle.apps.fnd.wf.bes.WebServiceInvokerSubscription, enables the invocation of Web services from Oracle E-Business Suite. Therefore, the invocation of Web services using 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

By using Oracle Workflow Business Event System to create events and event subscriptions, Web service invocation metadata can be defined. When a triggering event occurs, a Web service can be invoked through an appropriate event subscription.

Before invoking Web services, the following Web service invocation metadata must be defined first through the Business Event System:

- **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 access the Invoke Web Service wizard where you can specify a WSDL file 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.

  While defining a local subscription to the Invoker event, you can also specify the following subscription parameters:

  - Security parameters to support UsernameToken based WS-Security, page 9-14
    - WFBES_SOAP_USERNAME
    - WFBES_SOAP_PASSWORD_MOD
    - WFBES_SOAP_PASSWORD_KEY
  - Message transformation parameters to support XSL transformation, page 9-16
    - WFBES_OUT_XSL_FILENAME
    - WFBES_IN_XSL_FILENAME

Security Parameters to Support 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 enforce this UsernameToken based WS-security during Web service invocation, this WS-security mechanism provides a basic authentication for Web service invocation by passing a *username* and an optional *password* in the SOAP Header of a SOAP request sent to the Web service provider.

During the Web service requests or service invocation, the SOAP username and optional password locator information will be passed to the seeded Java rule function as the following subscription parameters when the Java rule function is defined through the Invoke Web Service wizard:

- The username for the operation is stored in invoker subscription parameter WFBES_SOAP_USERNAME.

  For example, it can be `WFBES_SOAP_USERNAME=SYSADMIN`.

- The password corresponding to the SOAP username is stored in FND vault using a PL/SQL script `$FND_TOP/sql/fndvltput.sql`. The module name and key to retrieve the password corresponding to the SOAP user is stored in the following subscription parameters:

  - WFBES_SOAP_PASSWORD_MOD

    For example, it can be `WFBES_SOAP_PASSWORD_MOD=PO`.

  - WFBES_SOAP_PASSWORD_KEY

    For example, it can be `WFBES_SOAP_PASSWORD_KEY=OrderConfirmService`.

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. For example, the event parameters are passed as follows:

- `BusinessEvent.setStringProperty("WFBES_SOAP_USERNAME", "SYSADMIN");`

- `BusinessEvent.setStringProperty("WFBES_SOAP_PASSWORD_MOD", "PO" );`

- `BusinessEvent.setStringProperty("WFBES_SOAP_PASSWORD_KEY", "OrderConfirmService");`

The seeded Java rule function then retrieves the password from FND vault and generates WS-Security header for the request to authenticate the Web services.

Parameters to Set Values for Input Parts

Two topics are discussed in this section:

- Event Payload as SOAP Body, page 9-15

- 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, then 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 available at the time of invoking the Web service requires 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 PL/SQL layer using a PL/SQL API `WF_EVENT.Raise` or from 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 *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 also 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/xmlgateway/ont__poi/">
<type>
   <schema elementFormDefault="qualified" targetNamespace="http:
//xmlns.oracle.com/apps/ont/soaprovider/xmlgateway/ont__poi/">
    <include schemaLocation="http://<hostname>:
<port>/webservices/SOAProvider/xmlgateway/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://host:
port/webservices/SOAProvider/xmlgateway/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 same as the 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 = "<ns1:SOAHeader
xmlns:ns1=\"http://xmlns.oracle.com/xdb/SYSTEM\" " +
            "env:mustUnderstand=\"0\"
xmlns:env=\"http://schemas.xmlsoap.org/soap/envelope/\"> \n" +
         " <ns1:MESSAGE_TYPE>XML</ns1:MESSAGE_TYPE>\n" +
         " <ns1:MESSAGE_STANDARD>OAG</ns1:MESSAGE_STANDARD>\n" +
         " <ns1:TRANSACTION_TYPE>PO</ns1:TRANSACTION_TYPE>\n" +
         " <ns1:TRANSACTION_SUBTYPE>PROCESS</ns1:
TRANSACTION_SUBTYPE>\n" +
         " <ns1:DOCUMENT_NUMBER>123</ns1:DOCUMENT_NUMBER>\n" +
         " <ns1:PARTY_SITE_ID>4444</ns1:PARTY_SITE_ID>\n" +
         "</ns1: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 an error condition 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 is >= 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, you need to create external subscription to the receive event to pass the Web service response.

> **Note:** If it is raised from Java with subscription has 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 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 *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 *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 you have agent listener set up properly.

Detailed information about these callback subscription parameters, see *Oracle E-Busines Suite Integrated SOA Gateway Developer's Guide*.

Managing Errors

If there is a run-time 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 run-time exception when the Workflow Java Deferred Agent Listener executes event subscription to invoke the Web service, the event is enqueued to 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 `SYSADMIN` with Web service definition, error details and event details. `SYSADMIN` 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 *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 OC4J, 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 OC4J can be tested. If there is a run-time 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
- WS-Security: Information required to add UsernameToken header to a SOAP request
    - WFBES_SOAP_USERNAME
    - WFBES_SOAP_PASSWORD_MOD
    - WFBES_SOAP_PASSWORD_KEY
- Input Message part value: Pass values for any part that may be required to embed application context into SOAP envelopes
    - WFBES_INPUT_`<partname>`
- 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 *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
- invokeService
- addWSSecurityHeader
- setInputParts

For more information on these methods, see *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 application 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.

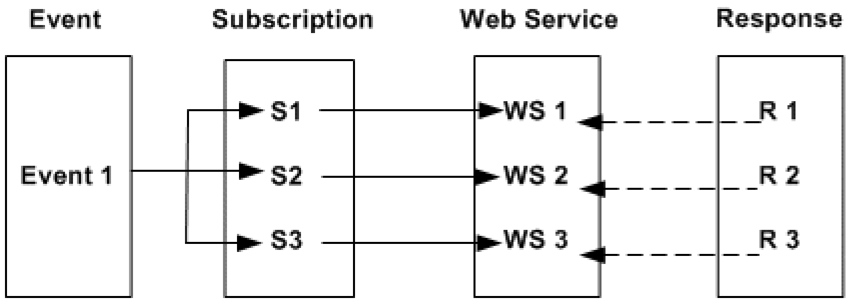
- Support Document Style Web Services Only

  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

Oracle E-Business Suite Integrated SOA Gateway utilizes Oracle Diagnostics, a tool to help standardize data gathering and troubleshooting. Through the Oracle Diagnostics Framework (ODF), integration repository administrators with appropriate diagnostic roles can run related tests to check the overall health of Oracle E-Business Suite Integrated SOA Gateway and gather interface and deployment information. The diagnostics will help the administrators with ease of maintenance of integration setup and transaction.

You can access Oracle Diagnostics through different user interfaces, including Oracle Applications Manager and other administrative consoles. For more information, see the *Oracle Diagnostics Framework User's Guide*.

The Oracle E-Business Suite Integrated SOA Gateway diagnostic tests are available in Oracle Diagnostics under the Application Object Library application.

The following topics are included in this chapter:

- How to Run Diagnostic Tests, page A-1

- Oracle E-Business Suite Integrated SOA Gateway Diagnostic Tests, page A-3

How to Run Diagnostic Tests

Use the following steps to run Oracle E-Business Suite Integrated SOA Gateway diagnostic tests and view the reports:

- Granting Roles to the User, page A-2

- Executing the Diagnostic Tests, page A-2

Granting Roles to the User

Use the following steps to grant the 'Application Super User Role' (UMX|ODF_APPLICATION_SUPER_USER_ROLE) and 'Diagnostics Super User' roles to a desired user:

1. Log on to the Oracle E-Business Suite as a user who has the User Management responsibility.

2. Click the Users link from the navigation menu to open the User Maintenance window.

3. Enter information in the search area to locate the appropriate user who you need to assign the roles.

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

5. In the Update User window, click **Assign Roles**.

6. In the search window, search for the 'Application Super User Role' (UMX|ODF_APPLICATION_SUPER_USER_ROLE). Choose this 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.

8. Repeat step 5 to step 7 to assign the 'Diagnostics Super User' role to the user.

Executing the Diagnostics Tests

Use the following steps to execute the diagnostic tests for Integrated SOA Gateway and view the reports:

1. Log on to Oracle E-Business Suite as a user who has been assigned to the above roles.

2. Select the Application Diagnostics responsibility from the Navigator and click the Diagnose link.

3. In the Diagnostic Tests window, click **Select Application** to open the Search and Select: Application pop-up window.

4. Search for Application Short Name as FND and then click **Select** to return to the Diagnostic Tests window.

5. In the Tests column, expand the '**Integrated SOA Gateway**' group by clicking on

the '+' icon to view all the available diagnostic tests.

6. Select the diagnostic test you want to execute and then click **Execute**.

7. Specify the following information in the Request Details window if needed:

   - Request Name: This field is populated automatically with the selected test name.

   - Generate Downloadable Report check box: Select this check box.

   - Download Format: Select an appropriate value from the drop-down list.

8. Click **Submit** to submit your request.

9. In the View Execution Results window, click **Refresh** to view the request status. Once the report is generated with 'Completed' status, click **View Report** to view your report details. Click **Download Report** to download the report with your selected format.

Oracle E-Business Suite Integrated SOA Gateway Diagnostic Tests

Oracle E-Business Suite Integrated SOA Gateway provides the following diagnostic tests through Oracle Diagnostics that you can use to check the interface type and deployment information as well as validate the service invocation.

IREP1: Public Class Count by Interface Type

This diagnostic test checks and reports the count of the Oracle Integration Repository public packages by Interface Types (PL/SQL, XML Gateway, and Concurrent Program). It also reports the number of packages that have either been generated or deployed.

This test does not require any input parameters.

IREP2: List Class Method and Check for Wrapper Package Status

For an inputted interface type, this diagnostics test checks and reports the packages and their methods for all the packages which are either Generated or Deployed. Additionally, if the interface type is PL/SQL, it detects if the wrapper packages are valid or not.

Input Required:

Interface Type: This test requires you to enter a valid interface type. Valid inputs are PLSQL, XML Gateway and Concurrent Program.

SOA Gateway Deployment Test

This diagnostic test checks if the Oracle E-Business Suite Integrated SOA Gateway deployment code works properly.

Input Required:

IREP Class Name: This test requires you to enter the internal name of a package as the IREP Class Name, such as `FND_USER_PKG`.

It checks and reports the test result if a valid WSDL file is generated upon deployment.

Service Invocation Framework Test

This diagnostic test uses the Service Invocation Framework to call a native Oracle E-Business Suite Web service (using `oracle.apps.wf.sif.test.fnd_user_pkg` event name and the subscription for `fnd_user_pkg.TestUserName` operation for the test).

Prerequisites:

Before running this diagnostic test, ensure the following information is in place:

- The `FND_USER_PKG` interface within the same Oracle E-Business Suite instance needs to be generated and deployed before this diagnostic test can be run.

- Once `FND_USER_PKG` is deployed, the method `TESTUSERNAME` should be granted to the user using which this invocation is being performed.

  This diagnostic test takes four input parameters (User_Name_To_Check, SOAP_User_Name, SOAP_Password_Mod, and SOAP_Password_Key) required for running this test. For description of each input parameter, see Input Required, page A-4.

- The username and password to be used by the service invocation framework for invoking services should be stored in FND vault using a PL/SQL script `$FND_TOP/sql/afvltput.sql`. For example,

  ```
  sqlplus apps/password@db @$FND_TOP/sql/afvltput.sql <Module>
  <Key> <Value>.
  ```

  The module name and key to retrieve the password value corresponding to the SOAP user are required to be passed to the SIF diagnostic test as inputs.

  For more information about UsernameToken based WS-Security header during Web service invocation, see Supporting WS-Security, *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

For more information on how to invoke Web services using the service invocation framework, see Implementing Service Invocation Framework, page 9-6.

Input Required:

- User_Name_To_Check: This parameter is passed to TESTUSERNAME operation of the FND_USER_PKG Web service to check whether that user exists in the database or not.

  Username is needed to be checked by fnd_user_pkg.TestUserName operation.

- SOAP_User_Name: This parameter is used in SOAP security header.

- SOAP_Password_Mod: This parameter is used in conjunction with Password Key to get the password of above Username from FND vault and that password is used in SOAP security header.

- SOAP_Password_Key: This parameter is used in conjunction with Password Module to get the password of above Username from FND vault and that password is used in SOAP security header.

It checks and reports the test result if it gets a correct string response.

SOA Gateway Health Check

This diagnostics test checks the overall health of Oracle E-Business Suite Integrated SOA Gateway.

Prerequisites:

In order to have the report generated successfully, you must create security grants to authorize the following interfaces access privileges to appropriate users who will execute the report. These interfaces are the SOA Gateway Health Check report running against; therefore, the authorized users must be created first.

- SOA Health Check Test Package (SOA_DIAG_TST)

  This is the predefined PL/SQL package executed by the SOA Gateway Health Check Report.

- Concurrent Program for SOA Health Check Report (SOA_DIAG_TEST)

  This is the predefined Concurrent Program executed by the SOA Gateway Health Check Report.

- ISG Sample Map (WF:ISGSAMP)

  This is the predefined XML Gateway Map executed by the SOA Gateway Health Check Report.

To create a security grant, search for a predefined integration interface first. Next, select the interface method (or procedure and function) from the interface details page (such as select TESTFUNCTION from the SOA_DIAG_TST interface details page) and click **Create Grant**. Select an appropriate user as a grantee. For more information, see Creating Grants, page 3-15.

Input Required:

To successfully run this report, you should provide the following SOAHeader elements as input parameters. These elements are used to pass values that may be required to set applications context during service execution.

- Application User Id
- Application User Password
- Responsibility Short Name
- Responsibility Application
- Security Group
- NLS Language
- Org_Id
- Security Group

This test reports the health check details for Integrated SOA Gateway including the following areas:

- Integrated SOA Gateway Install Steps Check

  It checks and reports the Results of Install Patch Test including the related patch information, patch description, and install status of each patch.

  Additionally, it checks if Oracle Application Server Adapter for Oracle Applications is properly installed, as well as if ASADMIN user is properly configured.

- Integrated SOA Gateway check for PL/SQL Interface

  It checks and reports the statuses of generate/regenerate, deploy, and run-time functions for the predefined PL/SQL package. You could look for more information in the log files.

- Integrated SOA Gateway check for XML Gateway Interface

  It checks and reports the statuses of generate/regenerate, deploy, and run-time functions for the predefined XML Gateway map. You could look for more information in the log files.

- Integrated SOA Gateway check for Concurrent Program Interface

  It checks and reports the statuses of generate/regenerate, deploy, and run-time functions for the predefined Concurrent Program. You could look for more information in the log files.

Troubleshooting SOA Gateway Health Check Report

In the case of the report is not run or errors occur while it is running, use the following methods to resolve or troubleshoot the issues:

- Enabling the OAM logging to troubleshoot the issue if the report is not run.

  Reason: This SOA Gateway Health Check Report uses the logger `oracle.apps.fnd.oam.sdk.util.logger.Logger` and it only prints the log statements at DEV level. To resolve the issue, enable the OAM logging.

  Resolution:

  1. Go to the Oracle Applications Manager Site Map.

  2. In the Others section, click on the Applications Manager Log link. This opens the Oracle Applications Manager Log pop-up window.

  3. Set the Current Log Level to DEV.

  4. Click **Go** to run the report and look for more information in the log.

- Use `opatch` command to troubleshoot if errors occur while the report is running.

  In the `opatch` section, it indicates if the patch is applied or not. For the interfaces, it gives the fault code. If the above information is not enough to have the issue resolved, enable the OAM logging to DEV level and look at the log information.

  Additionally, use the `opatch` command to get the patches installed information. The `opatch` command used by this report is as follows:

  ```
  IAS_ORACLE_HOME+"/OPatch/opatch lsinventory -oh
  "+IAS_ORACLE_HOME+" -invPtrLoc "+IAS_ORACLE_HOME+"/oraInst.loc
  ```

  This command can be used from the terminal.

  For the services, enable the SOA log and run the report and look 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.

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.

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.

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.

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.

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.

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.

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 a machine-processable format (specifically 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.

XML

XML (Extensible Markup Language) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

Index