

Oracle® E-Business Suite

Integrated SOA Gateway Implementation Guide

Release 12.1

Part No. E12169-04

February 2010

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

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

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

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

Send Us Your Comments

Preface

1 Oracle E-Business Suite Integrated SOA Gateway Overview

Oracle E-Business Suite Integrated SOA Gateway Overview.....	1-1
Major Components Features and Definitions.....	1-3
Native Service Enablement Architecture Overview.....	1-5

2 Setting Up Oracle E-Business Suite Integrated SOA Gateway

Setup Overview.....	2-1
Enabling ASADMIN User.....	2-1
Creating a New Oracle E-Business Suite User Account.....	2-2
Creating an Oracle E-Business Suite User Account.....	2-3
Granting Apps Schema Connect Role to the User.....	2-4
Configuring User in Technology Stack.....	2-5
Setting Profile Options.....	2-6

3 Administering Native Services

Overview.....	3-1
Administering Native Services.....	3-2
Generating Web Services.....	3-4
Deploying and Undeploying Web Services.....	3-10
Subscribing to Business Events.....	3-14
Managing Security Grants.....	3-16

4 Administering Composite Services

Overview.....	4-1
Understanding Composite Service Enablement Process.....	4-1
Administering Composite Services.....	4-3
Viewing Composite Services.....	4-3
Downloading Composite Services.....	4-4

5 Administering Custom Integration Interfaces and Services

Overview.....	5-1
Setting Up and Using the Integration Repository Parser.....	5-3
Generating ILDT Files.....	5-8
Uploading ILDT Files to Integration Repository.....	5-11
Administering Custom Integration Interfaces and Services.....	5-12
Creating Security Grants.....	5-15
Generating Web Services.....	5-15
Deploying, Undeploying, and Redeploying Web Services.....	5-16
Subscribing to Custom Business Events.....	5-16
Viewing and Downloading Custom Composite Services.....	5-17

6 Securing Web Services

Overview.....	6-1
Managing Function Security and Data Security.....	6-1
Managing Role-Based Access Control Security.....	6-5
Managing MOAC Security.....	6-8
Managing Web Service Security.....	6-12

7 Monitoring and Managing SOAP Messages Using SOA Monitor

SOA Monitor Overview.....	7-1
Searching SOAP Requests and Responses.....	7-3
Viewing SOAP Request Details.....	7-5
Viewing SOAP Response Details.....	7-6
Viewing Method Details.....	7-8
Purging SOAP Requests and Responses.....	7-9
Enabling and Disabling Web Service Monitoring.....	7-13

8 Implementing Service Invocation Framework

Overview.....	8-1
Service Invocation Framework Architecture Overview.....	8-2

Service Invocation Framework Major Features.....	8-5
Implementing Service Invocation Framework.....	8-6

A Oracle E-Business Suite Integrated SOA Gateway Diagnostic Tests

Overview.....	A-1
How to Run Diagnostic Tests.....	A-1
Oracle E-Business Suite Integrated SOA Gateway Diagnostic Tests.....	A-3

Glossary

Index

Send Us Your Comments

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

Part No. E12169-04

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.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

- 1 Oracle E-Business Suite Integrated SOA Gateway Overview**
- 2 Setting Up Oracle E-Business Suite Integrated SOA Gateway**
- 3 Administering Native Services**
- 4 Administering Composite Services**
- 5 Administering Custom Integration Interfaces and Services**
- 6 Securing Web Services**
- 7 Monitoring and Managing SOAP Messages Using SOA Monitor**
- 8 Implementing Service Invocation Framework**
- A Oracle E-Business Suite Integrated SOA Gateway Diagnostic Tests**
- Glossary**

Related Information Sources

This book is included on the Oracle E-Business Suite Documentation Library, which is supplied in the Release 12.1 Media Pack. You can download soft-copy documentation as PDF files from the Oracle Technology Network at <http://www.oracle.com/technology/documentation/>. The Oracle E-Business Suite Release 12.1 Documentation Library contains the latest information, including any

documents that have changed significantly between releases. If substantial changes to this book are necessary, a revised version will be made available on the "virtual" documentation library on My Oracle Support (formerly *OracleMetaLink*).

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.
- **PDF Documentation** - See the Oracle E-Business Suite Documentation Library for current PDF documentation for your product with each release. The Oracle E-Business Suite Documentation Library is also available on My Oracle Support and is updated frequently.
- **Release Notes** - For information about changes in this release, including new features, known issues, and other details, see the release notes for the relevant product, available on My Oracle Support.
- **Oracle Electronic Technical Reference Manual** - The Oracle Electronic Technical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for each Oracle E-Business Suite product. This information helps you convert data from your existing applications and integrate Oracle E-Business Suite data with non-Oracle applications, and write custom reports for Oracle E-Business Suite products. The Oracle eTRM is available on My Oracle Support.

Related Guides

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

Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle E-Business Suite data.

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 E-Business Suite Flexfields Guide

This guide provides flexfields planning, setup, and reference information for the Oracle E-Business Suite implementation team, as well as for users responsible for the ongoing maintenance of Oracle E-Business Suite product data. This guide also provides information on creating custom reports on flexfields data.

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 JDeveloper 10g with Oracle Application Extension.

Oracle Application Framework Personalization Guide

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

Oracle E-Business Suite Installation Guide: Using Rapid Install

This book is intended for use by anyone who is responsible for installing or upgrading Oracle E-Business Suite. It provides instructions for running Rapid Install either to carry out a fresh installation of Oracle E-Business Suite Release 12, 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 Application Server Adapter for Oracle Applications User's Guide

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

Please note that this guide is in the Oracle Application Server 10g 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 E-Business Suite User Interface Standards for Forms-Based Products

This guide contains the user interface (UI) standards followed by the Oracle E-Business Suite development staff. It describes the UI for the Oracle E-Business Suite products and how to apply this UI to the design of an application built by using Oracle Forms.

Oracle Diagnostics Framework User's Guide

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

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

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

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

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

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

Oracle e-Commerce Gateway 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 Report Manager User's Guide

Oracle Report Manager is an online report distribution system that provides a secure and centralized location to produce and manage point-in-time reports. Oracle Report Manager users can be either report producers or report consumers. Use this guide for information on setting up and using Oracle Report Manager.

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.

Oracle XML Publisher Report Designer's Guide

Oracle XML Publisher is a template-based reporting solution that merges XML data with templates in RTF or PDF format to produce a variety of outputs to meet a variety of business needs. Using Microsoft Word or Adobe Acrobat as the design tool, you can create pixel-perfect reports from the Oracle E-Business Suite. Use this guide to design your report layouts.

This guide is available through the Oracle E-Business Suite online help.

Oracle XML Publisher Administration and Developer's Guide

Oracle XML Publisher is a template-based reporting solution that merges XML data with templates in RTF or PDF format to produce a variety of outputs to meet a variety of business needs. Outputs include: PDF, HTML, Excel, RTF, and eText (for EDI and EFT transactions). Oracle XML Publisher can be used to generate reports based on existing Oracle E-Business Suite report data, or you can use Oracle XML Publisher's data extraction engine to build your own queries. Oracle XML Publisher also provides a robust set of APIs to manage delivery of your reports via e-mail, fax, secure FTP, printer, WebDav, and more. This guide describes how to set up and administer Oracle XML Publisher as well as how to use the Application Programming Interface to build custom solutions.

This guide is available through the Oracle E-Business Suite online help.

Integration Repository

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

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

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.

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) provides a customer-focused robust communication and integration infrastructure between independently managed components and loosely coupled applications. This infrastructure not only allows greater and effective business integration between heterogeneous applications, but also facilitates the development and execution of complex business processes into highly flexible and reusable Web services. With this standardized and interoperable Web service platform, Oracle E-Business Suite Integrated SOA Gateway provides a powerful framework that accelerates dynamic business processes and service integration between applications over the Web.

Oracle E-Business Suite Integrated SOA Gateway is a complete set of service infrastructure. It supports almost all integration interface types and services invoked within Oracle E-Business Suites no matter if they are Oracle seeded integration interfaces or custom ones, or if they are native packaged interfaces or the services that are orchestrated using native services. With this pre-built, reusable business services and service-oriented components, Oracle E-Business Suite Integrated SOA Gateway provides a capability of allowing various users to perform different tasks and to monitor and manage service integration throughout the entire service deployment life cycle.

For example, system integration developers can perform end-to-end service integration activities including creating and annotating custom integration interfaces, orchestrating discrete Web services into meaningful end-to-end business processes, defining Web

service invocation metadata, and testing the Web service invocation.

Application users or system integration analysts can then browse through and search on available integration interfaces and services regardless of custom or Oracle packaged ones, as well as view each interface details through the centralized repository.

Integration repository administrators can take further actions on transforming native interfaces into Web services, and then deploying the services for public use and access. The administrators are also responsible for enforcing service related securities, monitoring and managing the entire integrated service deployment life cycle to ensure smooth service integration between applications.

With pre-built, reusable business services and an essential service-oriented framework allowing service generation, deployment, invocation, and management, Oracle E-Business Suite Integrated SOA Gateway is the intrinsic part of Oracle E-Business Suite for service enablement. It not only enables services within and beyond Oracle E-Business Suite, but also facilitates dynamic business execution through a seamless service integration and consumption over the internet.

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

Major Features

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

- Provide robust, consistent integration framework with extensive infrastructure based on SOA principles
- Integrate loosely coupled and heterogeneous applications
- Contain pre-built and reusable business services
- Provide native service enablement capability within the Oracle E-Business Suite
- Use native services as building blocks to create composite services
- Support annotated custom integration interfaces from Oracle Integration Repository
- Enforce function security and role-based access control security to allow only authorized users to execute administrative functions
- Enable Web service invocation from Oracle E-Business Suite
- Audit and monitor Oracle E-Business Suite service operations from native SOA Monitor

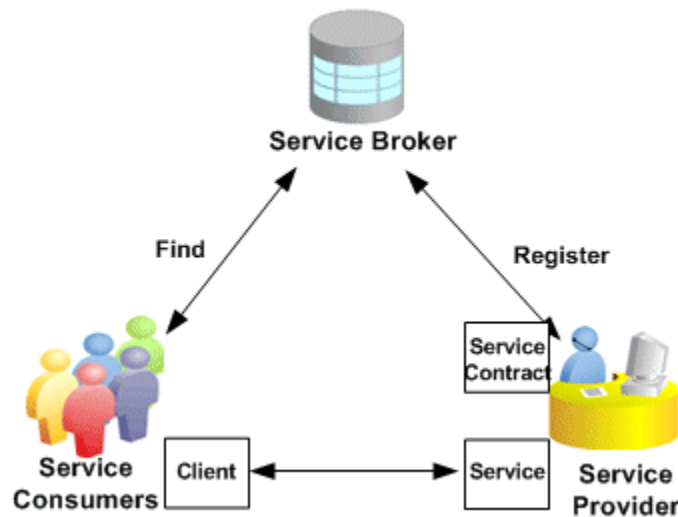
Major Components Features and Definitions

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 of enabling services:



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/Views
- EDI
- Business Service Object (Service Beans)
- Composite Services

Manage Security

Security is the most critical feature that is designed to guard service content from unauthorized access.

To ensure secure access and the execution of integration interfaces and Web services, Oracle E-Business Suite integrated SOA Gateway enforces the security rules through security grants to authorize interface methods access or feature access (such as the downloading composite services feature) to appropriate users. Multiple organization

access control security rule is also implemented for authorizing interface execution related to multiple organizations.

Additionally, Web service security rule is enforced for Web service authentication, requiring an username and password to be passed as part of the security header in the SOAP request sent to the Web service.

SOA Monitor

SOA Monitor is an audit and managing tool that allows all SOAP request and response messages received by SOA Provider and Web Service Provider to be logged and displayed (if the SOA Monitor 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.

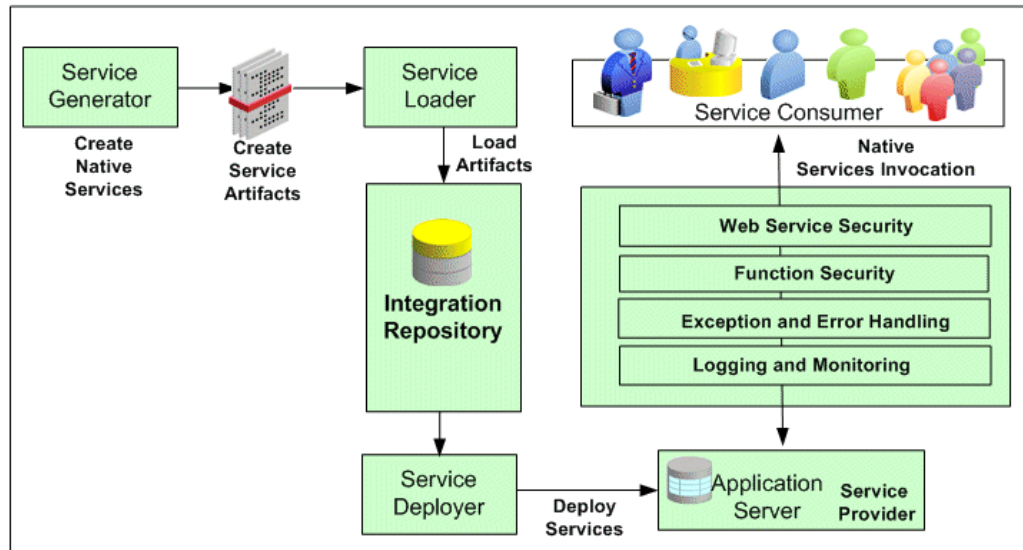
See Implementing Service Invocation Framework, page 8-1.

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 run time, and ease the service management throughout the entire service integration and deployment life cycle.

The seamless integration between each component forms the Oracle E-Business Suite Integrated SOA Gateway architecture.

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



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

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.

Service Generator

All the native packaged public integration interfaces are published in Oracle Integration Repository by default and these interfaces are generated into Web services by Service Generator.

Service Provider

Service Provider is the primary engine underlying the Web service capability. To support all published integration interface types and services in Oracle E-Business Suite Integrated SOA Gateway, an enhanced Web Service Provider called SOA Provider is particularly used to achieve the necessary functionality of Service Provider plus additional features in supporting various interface types.

At run time, SOA Provider references integration services and data from Oracle Integration Gateway in processing inbound SOAP request messages that invoke Web services and sends the SOAP response out. It also manages the execution of the service, and gets back the response to the requests.

Native Service Enablement 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: In Release 12.0, Oracle E-Business Suite is service partially enabled using **Web Service Provider** for the following interface types:

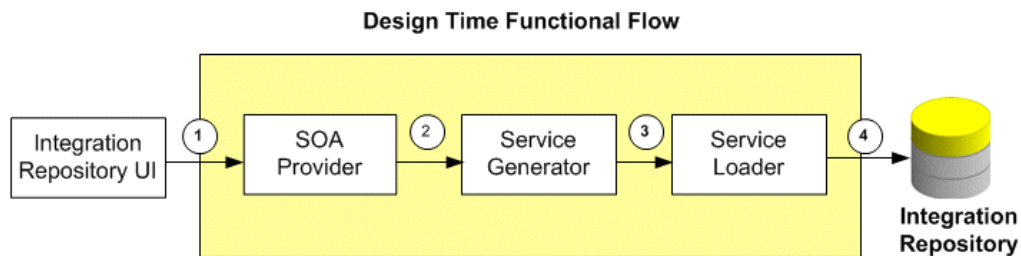
- XML Gateway Map
- Business Service Object that was formerly known as Service Bean

In this release, Oracle E-Business Suite Integrated SOA Gateway will continue to support the Release 12.0 based Web Service Provider service enablement, plus the following additional interface types using **SOA Provider**:

- Inbound Native Interfaces
 - PL/SQL
 - XML Gateway Map
 - Concurrent Program
- Business Event
- Composite Service - BPEL

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 interface type. 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:



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.

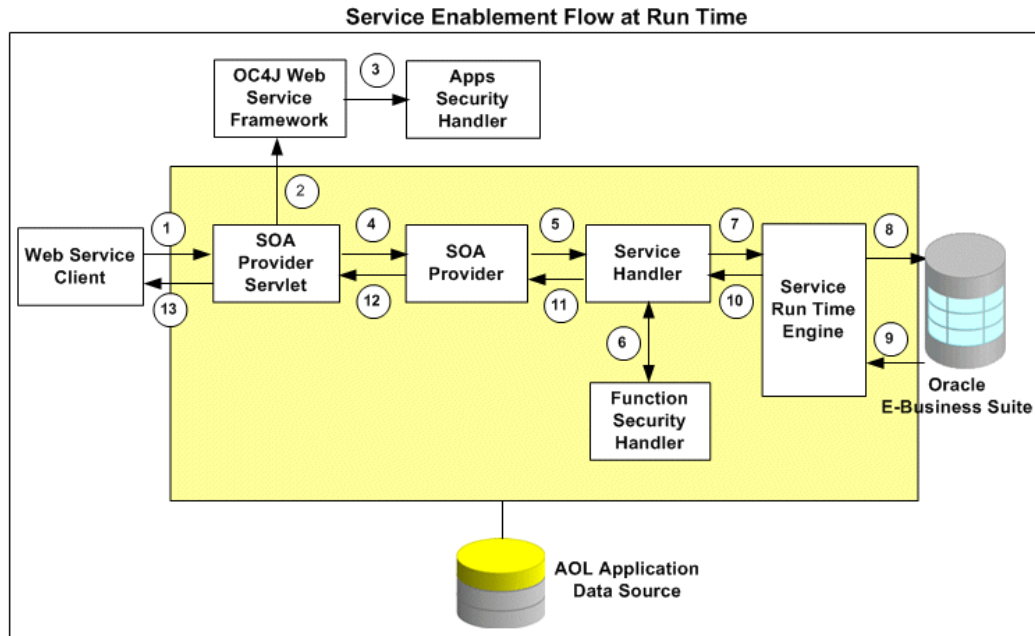
Integration repository administrators as defined by the Integration Repository Administrator role can further deploy the services from Oracle Integration Repository to the application server where services can be exposed to customers through service provider.

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 6-6.

Native Service Enablement Run Time

At run time, 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 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.

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 2-2
- Setting Profile Options, page 2-6

For information on how to install Oracle E-Business Suite Integrated SOA Gateway, see *Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12*, My Oracle Support Knowledge Document 556540.1 for details.

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

Enabling ASADMIN User

Use the following steps to enable ASADMIN user:

1. Log in to Oracle E-Business Suite with an 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 SOAPProvider for DB connection</description>
<credentials>!<NEW PASSWORD></credentials>
</user>
```

Note: The password should be preceded by a '!' (Exclamation) so that when OAFM is started, it gets encrypted. For example, if the password is 'welcome', then you have to change it to '!welcome'. In addition, the password of ASADMIN should be synchronized between system-jazn-data.xml file and the database through the application user interfaces.

Caution: Do not use the same password given in this example for "ASADMIN" user. Use different password instead to avoid some potential security risk in your system.

- Bounce the middle-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 2-4
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 ADADMIN 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 with an 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 with an 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.
6. 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 with an administrator role and choose the User Management responsibility.
2. Select the Users link from the navigation menu.
3. Enter appropriate information in the search area to locate 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 field 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 run time.

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`:
Please 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 a '!' (exclamation) so that when OAFM oc4j is started, it gets encrypted. For

example, if the password is 'welcome', then you have to change it to '!weclome'.

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 SOAPProvider for DB
connection</description>
  <credentials>!ASADMIN</credentials>
```

New Values:

```
<user>
  <name>New_User</name>
  <display-name>Default Apps SOA
User</display-name>
  <description>Used by SOAPProvider for DB
connection</description>
  <credentials>!New_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 monitoring 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 feature.
- The maximum number of SOAP request and response messages for SOA Monitor to buffer before saving them to the database.
- The maximum save interval in seconds for the SOA Monitor to save SOAP messages to the database.

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	<p>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.</p> <p>You can select one of the following three profile values:</p> <ul style="list-style-type: none"> WSP - Web Service Provider <p>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.</p> SOAP - SOA Provider <p>If this profile value is selected, then the Web Service - SOA Provider region will be displayed in the XML Gateway Map interface details page.</p> BOTH - Both 	Yes	<p>SOAP (SOA Provider)</p> <p>Important: If you do not start from this release and your system is upgraded from the Release 12.0 instead, then you should set the profile value 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 Map interface type in Oracle E-Business Suite 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. That way, the Web Service - Web Service Provider region and the Web Service - SOA Provider region can both be displayed simultaneously in the interface</p>

Profile Option	Description	Required	Default Value
	<p>Web Service Provider and SOA Provider</p> <p>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.</p>		<p>details page if Web services are available.</p> <p>Otherwise, a fault message appears if it is still set to the default profile value 'SOAP' (SOA Provider).</p> <p>If you start with Rapid Install of Oracle E-Business Suite for this release, then the default service provider is SOA Provider ('SOAP' profile value). In this situation, Web Service Provider will be disabled and any invocations of generic XML Gateway Web services will return a fault message.</p>

Profile Option	Description	Required	Default Value
SOA: Service Monitor	<p>Use this profile option to enable/disable the SOA monitoring feature.</p> <p>If it is enabled, SOAP request and response messages are saved to SOA Monitor.</p>	Yes	<p>ON</p> <p>Please note that this profile value can be overridden by clicking on the Turn Off Web Service Monitoring or Turn On Web Service Monitoring button in the SOA Monitor main page. For example, clicking Turn Off Web Service Monitoring will override the default value and disable the SOA monitoring feature.</p>
SOA: Maximum Number of Requests - Responses to Buffer	Use this profile option to set the maximum number of SOAP request and response messages for SOA Monitor to log before saving them to database.	Yes	5
SOA: Maximum Monitor Save Interval (Seconds)	Use this profile option to set the maximum save interval in seconds between two consecutive saves for SOA Monitor to pass logged SOAP messages and save them to the database.	Yes	120

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

Administering Native Services

Overview

Native services are native packaged public integration interfaces that are directly published in the Oracle Integration Repository by default. However, to make these native interfaces available over the Internet for service requesters or Web service clients to use or consume, the integration repository administrator must first transform these native interface definitions into a machine-processable format that complies with Web standards using WSDL description.

Once the Web services representing in WSDL URLs have been successfully generated, the administrator can further deploy them from the Integration Repository to Oracle Application Server. Additionally, those deployed services can also be undeployed from the server or redeployed again if necessary.

In addition to the service generation and deployment, the administrator can subscribe to a business event to create an event subscription with Out Agent.

To enforce the security access on certain integration interface methods and allow authorized users to have the access privileges, the administrator can create or revoke security grants to restrict user access to a specific interface method. This security mechanism will guard the integration interface access only to the right people at the right time.

The following native interfaces can be service enabled in Oracle E-Business Suite Integrated SOA Gateway:

- PL/SQL
- XML Gateway Map
- Concurrent Program

Important: Service enablement for concurrent programs linked to

Open Interfaces are currently not supported. This type of concurrent program can be viewed and displayed under the Open Interface category which Oracle Integration Repository does not support for service enablement.

- Business Event
- Business Service Object

Note: The Business Service Object, formerly known as Service Bean, is service enabled by Web Service Provider, not SOA Provider.

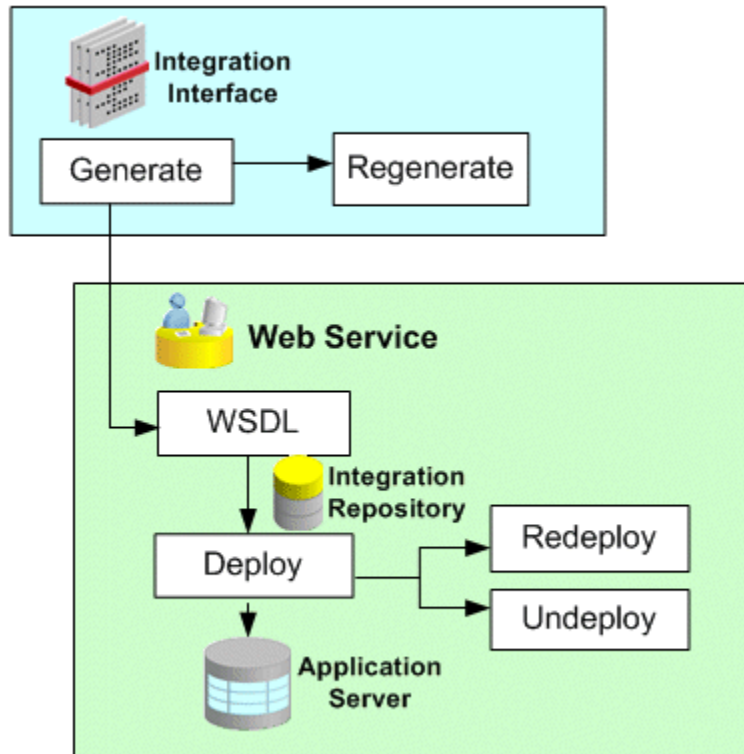
The service for XML Gateway Map can be enabled by both Web Service Provider in release 12.0 and SOA Provider in this release. For backward compatibility, a profile option 'FND: XML Gateway Map Service Provider' is used to let you select an appropriate service provider in enabling services for XML Gateway 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*.

Administering Native Services

Only integration repository administrators (defined by the Integration Repository Administrator role) can transform an integration interface definition into a Web service and then deploy (undeploy, or redeploy) the service.

The Web service generation and deployment process flow can be illustrated in the following diagram:

Service Generation and Deployment Process Flow



In addition to generating and deploying the Web services, the administrators can perform additional administrative tasks including creating subscription for a business event and creating security grants for appropriate interface methods access.

Note: The integration repository administrators (defined by the Integration Repository Administrator role) can find an additional SOA Monitor tab displayed next to the Integration Repository tab. This SOA Monitor tab allows the administrators to further audit or monitor 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 7-1.

The following administrative tasks are discussed in this section:

- **Generating Web Services**, page 3-4

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, a Web Service region becomes visible in the interface details page.

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

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

If the 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 3-16

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.

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 this permission set, see Role-Based Access Control (RBAC) Security, page 6-5.

Generating Web Services

To make integration interfaces available to customers over a network where customers can dynamically interact between applications, Oracle Integration Repository provides the feature allowing these interface definitions that are used internally within an organization to become Web available services.

To accomplish this goal, Oracle Integration Repository transforms these interface definitions 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.

To generate a Web service, you must be granted with the Integration Repository Administrator role allowing you to perform certain administrative functions or tasks, such as generating, deploying, or undeploying Web services.

First, locate the native interface that you want to generate and then click **Generate WSDL** in the native interface details page. This invokes Service Generator to generate service artifacts.

Note: A composite service consists of multiple native services which

have WSDL files generated already; therefore, there is no **Generate WSDL** or **Regenerate WSDL** shown in the composite service details page.

Generating Web Services



Important: Multiple requests to generate Web 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** button is disabled.

Once a Web service for a selected interface has been successfully generated, the Web Service - SOA Provider region will be displayed in the interface details page.

Note: If you are trying to generate a Web service for Business Service Object interface type, then you will find Web Service - Web Service Provider region available, instead of Web Service - SOA Provider region. If it is for XML Gateway Map interface type, you might find more than one Web Service Region available. See Generating Web Service Region(s) Based on a Selected Interface Type, page 3-6.

A WSDL link is available in the Web Service region. The Web Service Status field marked as 'Generated' also appears which indicates that this selected interface has WSDL description available. Additionally, you will notice that **Regenerate WSDL** appears allowing you to regenerate the WSDL if necessary.

You can perform the following tasks from the Web Service - SOA Provider region:

- Review the WSDL code by clicking the WSDL link.

See: Reviewing Web Service WSDL Source, *Oracle E-Business Suite Integrated SOA Gateway User's Guide*.

- Regenerate the WSDL code if necessary.

When an integration repository administrator clicks **Regenerate WSDL**, the system validates whether the interface definition has been changed since the last time the service was generated. If it has been changed after the last time the service was generated, a warning message appears to alert the administrator whether the service needs to be regenerated or not.

If the WSDL file that you want to regenerate is not used in any transactions, Oracle Integration Repository will regenerate the WSDL file and have **Deploy** available in the region allowing you to deploy the regenerated WSDL file. If the file is used in a transaction, an error message will appear.

- Deploy, redeploy, or undeploy the native service, page 3-10.

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.

See *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide* for details.

Generating Web Service Region(s) Based on a Selected Interface Type

In supporting Web Service Provider based and SOA Provider based service enablement for different interface types, an appropriate Web Service region will be displayed in the interface details page if a Web service is successfully generated.

- For XML Gateway Map interface type

The administrators could find either one of the following options available in the interface details page depending on the profile value set in the *FND: XML Gateway Map Service Provider* profile option:

- Web Service - Web Service Provider region

In Release 12.0, XML Gateway Map interface type is deployed by default through Web Service Provider. You can find both the standard WSDL URL (<http://host.com:port/webservices/AppsWSProvider/oracle/apps/fnd/XMLGateway?wsdl>) and deployed ones available in this region.

- Web Service - SOA Provider region (default)

You 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 run time.

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

- Business Service Object (BSO) interface type

This interface type has been service enabled by Web Service Provider since Oracle E-Business Suite Release 12.0. Web Service Provider will continue to support the BSO service enablement in this release. Therefore, you will only find the Web Service - Web Service Provider region displayed for BSO interface type.

- Other interface types supported by SOA Provider

The Web Service - SOA Provider region will be displayed.

Important: As mentioned earlier, in Release 12.0, Oracle E-Business Suite is service partially enabled by using Web Service Provider to enable XML Gateway Map and Business Service Object (formerly known as Service Bean) interface types.

In this release, Oracle E-Business Suite Integrated SOA Gateway will continue to support the Release 12.0 based Web Service Provider service enablement, plus the following additional interface types using SOA Provider:

- Inbound Native Interfaces

- PL/SQL
- XML Gateway Map
- Concurrent Program
- Business Event
- Composite Service - BPEL

To generate a Web service:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility. Select the Integration Repository link.
2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.
3. Expand an interface type node to locate your desired interface definition.
4. Click the interface definition name link to open the interface details page.
5. If this selected interface definition has never been generated into Web service, click **Generate WSDL** to generate the WSDL description.

Once the Web service is successfully generated, the Web Service - SOA Provider region becomes available. The Web Service Status field marked as 'Generated' also appears which indicates that this selected interface has WSDL description available.

Note: If you are trying to generate a Web service for Business Service Object interface type, then you will find Web Service - Web Service Provider region available, instead of Web Service - SOA Provider region. You might find more than one Web Service region available if it is for XML Gateway Map interface type.

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

Generating Web Services from Backend Processing

When trying to generate a Web 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://myhost.us.oracle.com/webservices/SOAProvider/EbizAuth?Generate=1656&soa_ticket=u0GKj8W6SbCL2BMDY_Mv9w..' 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 `$IAS_ORACLE_HOME`) needs to be set by running `.env` file in `APPL_TOP` of your environment.
2. If the user has `appmgr` privileges and has read permission on `$INST_TOP/ora/10.1.3/j2ee/oafm/config/oc4j.properties`, then he or she can run `soagenerate.sh` without any setup described in step 3.
3. If user does not have read permission on `$INST_TOP/ora/10.1.3/j2ee/oafm/config/oc4j.properties`, then he or she needs to set the following properties present in `JAVA_TOP/oracle/apps/fnd/soa/provider/wsdldata/soa.properties`:
 1. Set the following two database connection related properties in `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. User should have write permission on `SOA_SERVER_TEMP_DIRECTORY_LOCATION` mentioned in step 3.
5. User should have write permission on the directory from where he or she is 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 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 fired. If you want to create 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 is not printed in logfile. If you want to print system properties in log file, specify `printprops=true`.

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

Deploying and Undeploying Web Services

If a Web service WSDL file has been generated successfully, the administrators can deploy the service from the Integration Repository to the application server.

Deploying Web Service

The screenshot shows the Oracle Integration Repository SDA Monitor interface. On the left is a tree view of the Application Object Library. The main area displays details for the 'PLSQL Interface: Application Context APIs'. It includes fields for Internal Name (FND_GLOBAL), Type (PL/SQL), Product (Application Object Library), Status (Active), and Online Help (See related online help). The Scope is Public and the Interface Source is Oracle. A 'Regenerate WSDL' button is present. Below this is the 'Full Description' section, followed by the 'Web Service - SOA Provider' section which shows the Web Service Status as 'Generated' and the WSDL URL. A 'Deploy' button is located here. The 'Source Information' section shows the Source File, Source Version, and Source Product. The 'Procedures and Functions' section includes a 'Select Object and Create Grant' button and a table of procedures.

Select Details	Name	Internal Name	Status	Description
<input type="checkbox"/> Show Get Conc Appl ID	PROG_APPL_ID	Active	Returns concurrent program Application ID.	
<input type="checkbox"/> Show Get Conc Login ID	CONC_LOGIN_ID	Active	Returns concurrent program login ID.	
<input type="checkbox"/> Show Get Conc Program ID	CONC_PROGRAM_ID	Active	Returns concurrent program ID.	
<input type="checkbox"/> Show Get Conc Request ID	CONC_REQUEST_ID	Active	Returns concurrent Request ID.	
<input type="checkbox"/> Show Get Login ID	LOGIN_ID	Active	Returns login ID (unique per signon).	
<input type="checkbox"/> Show Get User ID	USER_ID	Active	Returns user id.	
<input type="checkbox"/> Show Initialize Globals	APPS_INITIALIZE	Active	Sets up global variables and profile values in a database session.	

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.

To deploy a Web service of an interface type other than Business Service Object, click **Deploy** in the Web Service - SOA Provider region of an interface details page. This deploys the service from Oracle Integration Repository to Oracle Application Server.

Once the Web service is successfully deployed, the Web Service Status field marked as 'Deployed' also appears which indicates that this selected service has been successfully deployed to the application server.

Please note that Business Service Object interface type, formerly known as Service Bean, is service enabled through Web Service Provider. To deploy the Business Service Object in the Web Service - Web Service Provider region, click **Deploy Web Service** to deploy the service. After the service is deployed, **Redeploy Web Service** appears allowing you

to redeploy the service if needed. The deployed WSDL URLs is also displayed in the region.

Reviewing Deployed WSDL

After you have successfully deployed or redeployed a Web service, you will find the 'Deployed' status displayed in the Web Service Status field along with an deployed WSDL link confirming that a deployed service has been successfully created.

Web service deployed WSDL information page

The screenshot shows the Oracle Integration Repository interface. The left sidebar contains a tree view of the repository structure, including categories like Business Event, Business Service Object, Composite - BPEL, Concurrent Program, EDI, Interface View, Java, Open Interface, and PL/SQL. The main content area is titled 'PLSQL Interface : Application Context APIs'. It displays metadata for the interface, including Internal Name (FND_GLOBAL), Type (PL/SQL), Product (Application Object Library), Status (Active), and Scope (Public). Below this, the 'Full Description' section explains that the interface returns system global values. The 'Web Service - SOA Provider' section shows the Web Service Status as 'Deployed' and provides a WSDL link: http://wss50066rems.us.oracle.com:8040/webservices/SOAProvider/plsql/fnd_global?wsdl. The 'Source Information' section lists the Source File as 'patch/115/sql/AFSCGBLS.pls', Source Version as '120.4.12000000.2', and Source Product as 'FND'. The 'Procedures and Functions' section includes a table of available procedures and functions.

Select Details	Name	Internal Name	Status	Description
<input type="checkbox"/> Show/Get Conc Appl ID	PROG_APPL_ID	Active	Returns concurrent program Application ID.	
<input type="checkbox"/> Show/Get Conc Login ID	CONC_LOGIN_ID	Active	Returns concurrent program login ID.	
<input type="checkbox"/> Show/Get Conc Program ID	CONC_PROGRAM_ID	Active	Returns concurrent program ID.	
<input type="checkbox"/> Show/Get Conc Request ID	CONC_REQUEST_ID	Active	Returns concurrent Request ID.	
<input type="checkbox"/> Show/Get Login ID	LOGIN_ID	Active	Returns login ID (unique per signon).	
<input type="checkbox"/> Show/Get User ID	USER_ID	Active	Returns user id.	
<input type="checkbox"/> Show/Initialize Global	APPS_INITIALIZE	Active	Sets up global variables and profile values in a database session.	

Click the deployed **WSDL** link to view the deployed source code, as shown in the following example:

```

<?xml version="1.0"?>
<definitions name="IntegrationRepositoryService"
targetNamespace="http://myhost.us.oracle.com/oracle/apps/fnd/rep/ws/Inte
grationRepositoryService"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns2="http://xmlns.oracle.com/apps/fnd/ServiceBean"
  xmlns:tns1="http://xmlns.oracle.com/apps/fnd/rep/ws"

xmlns:tns="http://myhost.us.oracle.com/oracle/apps/fnd/rep/ws/Integratio
nRepositoryService">
<types>
<xsd:schema>
  <xsd:import namespace="http://xmlns.oracle.com/apps/fnd/rep/ws"
schemaLocation="http://myurl.us.oracle.com:1234/webservices/AppsWSProvid
er/oracle/apps/fnd/rep/ws/IntegrationRepositoryService.xsd"/>
</xsd:schema>
<xsd:schema elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/apps/fnd/ServiceBean" >
  <xsd:element name="ServiceBean_Header">
    <xsd:complexType>
      <xsd:element name="RESPONSIBILITY_NAME" minOccurs="0"
type="xsd:string"/>
      <xsd:element name="RESPONSIBILITY_APPL_NAME" minOccurs="0"
type="xsd:string"/>
      <xsd:element name="SECURITY_GROUP_NAME" minOccurs="0"
type="xsd:string"/>
      <xsd:element name="NLS_LANGUAGE" minOccurs="0" type="xsd:string"/>
    </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
. . .

```

Additionally, you will find the following buttons available in the Web Service - SOA Provider region:

- **Redeploy:** allows you to redeploy the Web service if the service was regenerated from a new interface definition or because the original service was corrupt. Clicking on the **Redeploy** button will update the deployed service with the current system values.

Note: If you want to redeploy a service of the Business Service Object interface type in the Web Service - Web Service region, click **Redeploy Web Service** to redeploy the service.

- **Undeploy:** allows you to undeploy the Web service from Oracle Application Server back to Oracle Integration Repository if necessary.

If you try to undeploy a native service that is not used, Oracle Integration Repository undeploys the native service from the server.

Note: The **Deploy** and **Redeploy** buttons appear only if you are

granted with the Integration Repository Administrator role.

To deploy, undeploy, redeploy a Web service:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility. Select the Integration Repository link.
2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.
3. Expand an interface type node to locate your desired interface definition.
4. Click the interface definition name link to open the interface details page.
5. From the Web Service - SOA Provider region, click **Deploy** to deploy the service from Oracle Integration Repository to Oracle Application Server if the service has been successfully generated.

If you want to deploy a service of the Business Service Object interface type, then click **Deploy Web Service** to deploy the service.

Note: If this selected interface definition has never been generated into a Web service, you need to first generate the service and then deploy it. How to generate a Web service, see Generating Web Services, page 3-4.

6. Click the deployed WSDL link to view the deployed WSDL description.
7. Click **Redeploy** to redeploy the service if necessary.
If it is for Business Service Object interface type, then click **Redeploy Web Service** to deploy the service.
8. Click **Undeploy** to undeploy the service if necessary.

Subscribing to Business Events

For business events, users with 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.

Subscribing to a Business Event

The screenshot displays the Oracle Integration Repository web interface. On the left is a navigation tree with categories like Business Event, Financials, and Lease Management. The 'Lease Management' node is expanded. The main content area shows the 'Business Event Details : Lease Application - Recommended Alternate Offers Accepted'. It includes a 'Subscribe' button, a 'Search' button, and a 'Printable Page' button. Below these are fields for Internal Name, Type, Product, and Status. A 'Full Description' section provides details about the event. A 'Source Information' section lists the source file, version, and product.

Internal Name	Type	Product	Status
oracle.apps.oid.sales.leaseapplication.recommended_alternate_offers_accepted	Business Event	Lease Management	Active

Full Description
Lease Application - Recommended Alternate Offers Accepted Raised when the Alternate offer for a Lease application is accepted

Source Information

Source File	Source Version	Source Product
patch/115/xml/bpel/oksawfe_mod.wfx	12.0	OE

Subscribing to a business event creates an event subscription with Out Agent as WF_BPEL_Q.

Note: If a BPEL process is created with the business event that you have subscribed to it, in order for the subscribed business event to be successfully enqueued to WF_BPEL_Q queue, you need to make sure:

- The consumer name must be unique.
- The BPEL process is deployed before raising the business event.

Once the event subscription has been successfully completed, a confirmation message appears. Additionally, the **Unsubscribe** button appears in the details page allowing you to remove or unsubscribe the event. Clicking **Unsubscribe** removes the WF_BPEL_Q event subscription. A confirmation message also appears after the subscription has been successfully removed.

To subscribe to a business event:

1. Log on to Oracle Integration Repository with the Integration Repository Administrator role through the Integrated SOA Gateway responsibility. Select the Integration Repository link.
2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.
3. Expand the Business Event interface type node to locate your desired event.
4. Click the business event interface that you want to subscribe to it to open the Interface details page for the event.

5. Click **Subscribe** to subscribe to the selected event to create an event subscription with Out Agent as `WF_BPEL_Q`. A confirmation message appears after the event subscription is successfully created.
6. Click **Unsubscribe** to unsubscribe to the subscribed event if needed.

See Generate Web Services, page 3-4 and Deploy, Redeploy, or Undeploy Web Services, page 3-10.

Managing Security Grants

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 with authorized privileges to access or execute certain API methods exposed through Oracle Integration Repository. This protects application data from unauthorized access without security checks.

To accomplish this goal and only allow authorized users to access certain API methods, the interface methods of an inbound API are precreated as permissions and stored in FND function repository. An integration repository administrator can then select one or more methods in a package and then authorize a user, user group, or all users to execute the selected method(s) by creating appropriate security grants.

Note: To create and revoke grants, you must log on to Oracle Integration Repository with the Integration Repository Administrator role.

This security grant can be performed in the Create Grant page for a given interface type to control the method at a very granular level.

Note: Each of the overloaded function contained in an interface can be uniquely granted to a specific user, user group, or all users through the create grant feature. If you select more than one overloaded function in the Procedures and Functions region (or the Methods region), in the Create Grant page an Overloaded column appears in the selected methods table indicating more than one overloaded function is selected for the grant.

To create grants:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility. Select the Integration Repository link.
2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.

3. Expand an interface type node and click an interface definition name link you want to open the interface details page.
4. Select one or multiple method names for which you want to create grants.
5. Click **Create Grant**.

The Create Grants page appears.

Create Grants

ORACLE® Integration Repository

Home Logout Preferences Help Personalize Page Diagnostics

Integration Repository SQA Monitor

View By: Interface Type

Integration Repository > PLSQL Interface : Payment Instrument Registration >

Create Grants

Cancel Apply

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

Grant All Selected

Grantee Type: Specific User

Grantee Name:

6. Select a grantee type:
 - Specific User
 - Group of Users
 - All Users
7. If you selected **Specific User** or **Group of Users**, specify the user or group for which to create the grants in the Grantee Name field.
8. Click **Apply**.

The interface details page reappears.

To view or revoke grants:

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

1. Navigate to the interface details page that you want to view or revoke the grants.
2. In the Methods region, click **Show** for a given method to view its grant details in a table.

Grant details for two methods

<input type="checkbox"/>	Hide	receiveItem	Active	Receives a single purchaser order.
Grant Details				
Grantee	Granted Via	Grantee Type	Revoke	
All Users	Direct	GLOBAL		
<input type="checkbox"/>	Hide	receiveItems	Active	Receives a list of items.
Grant Details				
Grantee	Granted Via	Grantee Type	Revoke	
Feng, Amy	OIC Payment Analyst Manager Group	GROUP		
Jackson, Lou	OIC Payment Analyst Manager Group	GROUP		
OIC Payment Analyst Manager Group	Direct	GROUP		
OSCSUGER,	OIC Payment Analyst Manager Group	GROUP		
SUPER1,	OIC Payment Analyst Manager Group	GROUP		
Wells, Bruce	Direct	USER		

3. If you specified a group of users as the grantee, each member of the group, plus the group itself, is listed as a grantee.

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

Grantee Type can be one of the following values:

- `USER` - The grantee is an individual user who was selected directly.
- `GROUP` - The grantee is a group of users or a member of a group of users.
- `GLOBAL` - The grant was issued to all users.

4. You can revoke a grant by clicking the trash can icon in the **Revoke** column.

A confirmation page appears, where you can click **Apply** or **Cancel** to execute or cancel the action.

Note: For any users who were issued their grants as members of a group, you cannot revoke their grants individually, but only by revoking the grant for the entire group. The trash can icon is disabled for group members.

For information on function security, see *Managing Function Security*, page 6-1.

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

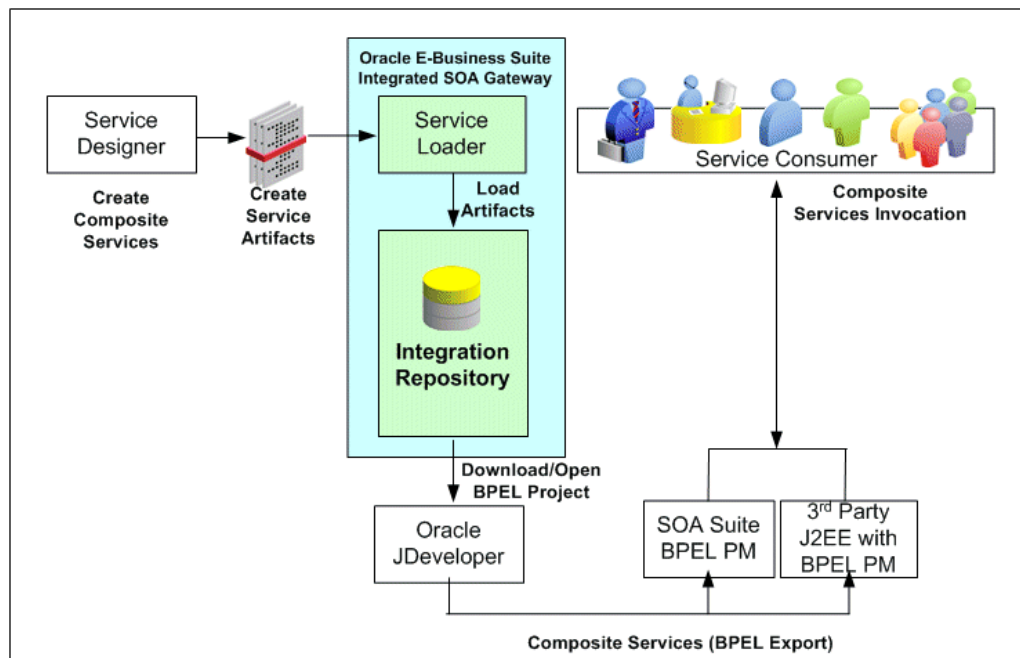
Composite services use native services as building blocks to orchestrate the business invocation sequence from discrete Web services into a meaningful end-to-end business flow through a Web service composition language BPEL. Strictly speaking, this type of interface is comparatively service enabled without 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 *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide* for details.

Composite Service Enablement Process



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

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. By clicking a composite service name link from the navigation tree or search results, you will find the composite service interface details page where displays all composite service details.

The composite service details page allows you to view the selected composite service WSDL file and the BPEL URL in the BPEL Files region.

The administrators can also download a corresponding composite service project file, such as BPEL 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 your local machine.

Important: In general, only integration repository developers and 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 integration analysts) will not find **Download Service** available in the details page.

For more information about how to grant Download Composite Service privilege, see Role-Based Access Control (RBAC) Security, page 6-5.

Composite Details Page with Download Privilege

The screenshot shows the Oracle Integration Repository interface. On the left is a navigation tree with categories like Business Event, Business Service Object, Composite - BPEL, Order Management Suite, Order Entry (selected), Supply Chain Management, Concurrent Program, EDI, Interface View, Java, Open Interface, PL/SQL, Service Data Object, Web Service, and XML Gateway Map. The main content area displays details for 'Composite Service BPEL : Test Process'. It includes a 'Download Service' button, a table of metadata (Internal Name: oracle.apps.oe.TestBPELProcess, Type: Composite - BPEL, Product: Order Entry, Status: Active, Business Entity: Sales Order, Scope: Public, Interface Source: Oracle), a 'Full Description' section stating 'This is Abhishek's Test BPEL File.', and a 'BPEL Files' section with an 'XML Representation' and an 'Abstract WSDL' link to a URL. Search and Printable Page buttons are present at the top and bottom right of the main content area.

To download the relevant files aggregated in a .JAR file for a composite service, navigate to the composite service details page for a service that you want to download, and then click **Download Service** to download the 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.

To download a composite service:

1. Log on to Oracle Integration Repository with the integration repository administrator role through the Integrated SOA Gateway responsibility. Select the Integration Repository link.
2. In the Integration Repository tab, select 'Interface Type' from the View By drop-down list.
3. Expand the Composite Service interface type node to locate your desired composite service.
4. Click the composite service that you want to download it to open the Composite Service Interface Details page.
5. Click **Download Service** to download the selected composite file to your local machine.

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, 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: Please note that custom interface types of EDI, Open Interface Tables and Interface Views are not supported in this release.

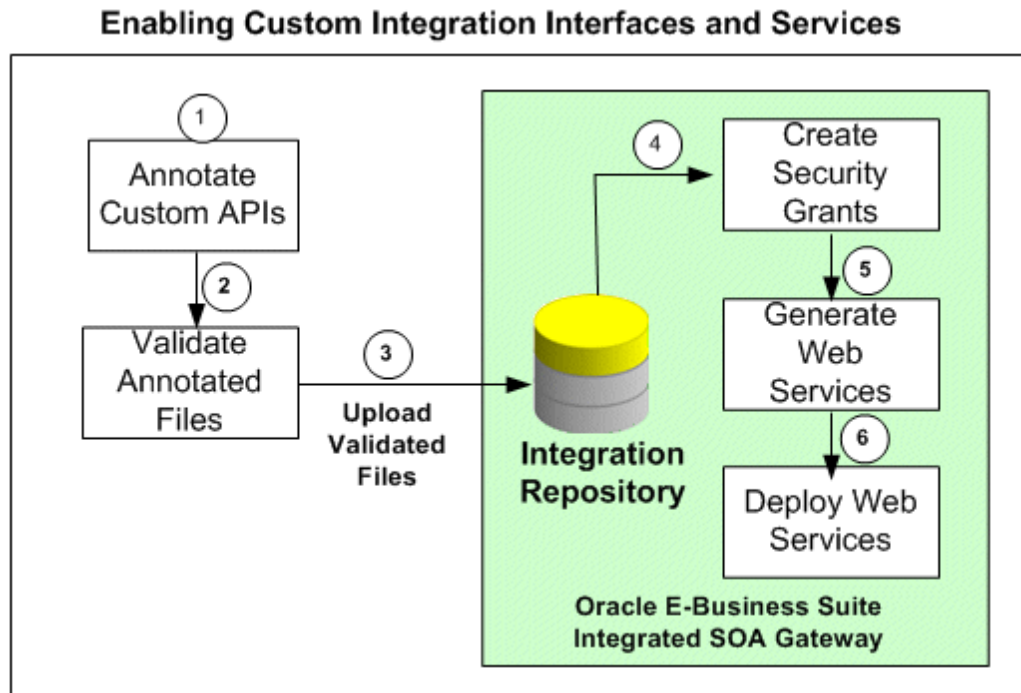
Oracle Integration Repository currently does not support the creation of custom Product Family and custom Business Entity.

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. Hence, 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, generating and deploying Web services.

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.

Enabling Custom Integration Interface Process Flow

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



1. A system integration developer annotates a custom integration interface definition based on the Integration Repository annotation standards for the supported interface types.
2. An integration repository administrator validates 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.
3. An integration repository administrator uploads the generated iLDT file to Oracle Integration Repository.
4. The integration repository administrator then creates necessary security grants for the custom integration interfaces if needed.
5. The administrator generates Web services if the custom interfaces can be service enabled.
6. The administrator deploys the Web services from Oracle Integration Repository to the application server.

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 5-3
- Administering Custom Integration Interfaces and Services, page 5-12

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 and apply the patch described in the prerequisites:

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 11 (5.8)
- 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:

Apply patch 9004712: One-Off for IREP Parser On Top of Oracle E-Business Suite Release 12.1.1.

- ***Prerequisites for Installing Perl Modules on Windows***
 - Apply Windows specific patch 9153106:R12.OWF.B.
 - 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 middle-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_1fc8b3b9a1e18e3b_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='1fc8b3b9a1e18e3b' />
    </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/Ap
ache/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. Search and download the following perl modules that are required to be installed manually from CPAN:

- Compress-Raw-Zlib-2.009
(<http://search.cpan.org/~pmqs/Compress-Raw-Zlib-2.009>)
- Compress-Zlib-2.009
(<http://search.cpan.org/~pmqs/Compress-Zlib-2.009>)
- Class-MethodMaker-1.12
(<http://search.cpan.org/~fluffy/Class-MethodMaker-1.12>)

For example, use the following steps to install

```
:
#gzip -d Compress-Raw-Zlib-2.009.tar.gz
tar -xvf Compress-Raw-Zlib-2.009.tar
cd Compress-Raw-Zlib-2.009
```

- **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 5-11

Generating ILDT Files

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

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

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 5-11.

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 5-9.

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 Interface Views 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 5-10 (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 verbatim into the output iLDT, and otherwise not examined. The `pathname` on the right-hand side must refer to a real file, which can be located anywhere.

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

Options

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

- `-generate`: It 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 5-9 and the `-development` flag, page 5-10.

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

Environment

The following environment variables 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 parser is not able to find a particular class, check for its availability in CLASSPATH.

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

For others, check your platform documentation on how to set `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`.

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. Issue 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/apps@instance_name 0 Y
UPLOAD $FND_TOP/patch/115/import/wfirep.lct SOAIS_pls.ildt`

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.

These administrative tasks include:

- **For Custom Integration Interfaces of Interface Types**

- Creating Security Grants, page 5-15

In addition to verifying if the uploaded custom interfaces can be successfully

retrieved from the Integration Repository, to let appropriate users actually use these custom files, the administrators can create security grants to grant appropriate interface methods access privileges to an appropriate user, a user group, or all users.

- Generating Web Services, page 5-15

Similar to the native integration interfaces, these custom interfaces can become WSDL-based Web services if the interface types they belong to can be service enabled.

- Deploying, Undeploying, and Redeploying Web Services, page 5-16

Once the custom interfaces become Web services, integration repository administrators can deploy the generated Web services to Oracle Application Server. The administrators can also redeploy or undeploy the Web services if needed.

- Subscribing to Custom Business Events, page 5-16

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

- **For Custom Composite Integration Interface**

- Viewing and Downloading Custom Composite Services, page 5-17

Integration repository administrators can view a custom composite service details, and download the .ZIP file for a composite service if it is available for download.

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.

Viewing from Interface List Page

The screenshot shows the Oracle Integration Repository interface. On the left is a tree view of the repository structure, including categories like Business Event, Business Service Object, Composite - BPFL, Concurrent Program, EDI, Interface View, Java, Applications Technology, Application Object Library, User Management, Financial Receivables Suite, Financials, Marketing Suite, Marketing and Sales Suite, Product Lifecycle Management, Public Sector, Sales Suite, Supply Chain Management, Open Interface, PL/SQL, Service Data Object, Web Service, and XML Gateway Map. The 'User Management' interface is selected.

The main area is titled 'Interface List : User Management'. It contains search filters: 'Interface Source' (All, Custom, Oracle), 'Scope' (All), and a 'Go' button. There is also an 'Export' button. Below the filters is a table with the following data:

Name	Internal Name	Product	Type	Source	Status	Description
RegistrationAdmin	oracle.apps.fnd.umx.registration.RegistrationAdmin	User Management	Java	Oracle	Active	RegistrationAdmin includes methods to support registration with Oracle User Management.
RegistrationBean	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.
RegistrationURLFramework	oracle.apps.fnd.umx.util.URLFramework	User Management	Java	Oracle	Active	URLFramework includes methods to support Centralized Account Provisioning System in Oracle Applications.

At the bottom of the page, there is a footer with 'About this Page', 'Privacy Statement', 'Integration Repository', 'Home', 'Logout', 'Preferences', 'Help', 'Diagnostics', and 'Copyright (c) 2006, Oracle. All rights reserved.'

- 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 'Composite' as the interface type to locate the custom composite services.

Viewing from Interface Search Page

ORACLE[®]

Integration Repository

Home Logout Preferences Help Diagnostics

Integration Repository

Search

Browse

Interface Name

Product Family

Product

Internal Name

Interface Type

Business Entity

Hide More Search Options

TIP Select Category before a Category Value

Category

Category Value

Interface Source

Status

Web Service Status

Standard

Standard Specification

Scope

Export

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

Browse

Integration Repository SOA Monitor Home Logout Preferences Help Diagnostics

About this Page Privacy Statement

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

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

Creating Security Grants

To let appropriate users actually 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 an appropriate user, a user group, or all users.

This security grant is performed in the Create Grant page for a selected custom integration interface to control the method at a very granular level.

For more information, see Creating Grants, page 3-16.

Generating Web Services

Once custom integration interfaces are uploaded to Oracle Integration Repository, an integration repository administrator can transform these interface definitions into WSDL descriptions if the interface type they belong to can be service enabled.

To generate a WSDL URL so that a selected interface can become Web service, the administrator who has the privilege must first locate the custom interface that you

desire and then generate the Web service by clicking **Generate WSDL** in the interface details page.

If the Web service is successfully generated, you should notice a WSDL link available along with the 'Generated' Web Service Status field displayed in the Web Service - SOA Provider region of the details page.

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

Note: Because custom integration interface is supported in this release, even if a custom interface is for XML Gateway Map or Business Service Object interface type, there will not be the Web Service - Web Service Provider region in the details page for the backward compatibility.

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

- **Regenerate WSDL**

This lets you regenerate the WSDL if necessary

- **Deploy**

This lets you deploy the generated service from Oracle Integration Repository to the application server

See: *Deploying, Undeploying, and Redeploying Web Services*, page 5-16.

For detailed information on how to generate Web services on native integration interfaces, see *Generating Web Services*, page 3-4.

Deploying, Undeploying, and Redeploying Web Services

Once the Web services is successfully generated for custom integration interfaces, like native packaged interfaces, the administrator will perform the same deployment activities if necessary to deploy the generated services to the application server, and redeploy or undeploy them again from the server.

For detailed information on how to deploy, undeploy, or redeploy Web services, see *Deploying, Undeploying, and Redeploying 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. This creates an event subscription with Out Agent as `WF_BPEL_Q`.

Once the event subscription has been successfully completed, a confirmation message

appears. The **Unsubscribe** button appears in the interface details page allowing you to remove or unsubscribe the event. Clicking **Unsubscribe** removes the WF_BPEL_Q event subscription. A confirmation message also appears after the subscription has been successfully removed.

For more information, see *Subscribing to Business Events*, page 3-14.

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.

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 6-5
- Multiple Organization Access Control Security (MOAC Security), page 6-8
- WS-Service Security (Web Service Security), page 6-12

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 E-Business Suite 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.

Interface Contains One Default Method

ORACLE Integration Repository Home Logout Preferences Help Diagnostics

Integration Repository | SOA Monitor

View: Interface Type Business Service Object Concurrent Program Applications Technology **e-Commerce Gateway** Contracts Suite Contracts Core Discrete Manufacturing Financial Globalizations Suite Financial Payables Suite Financial Receivables Suite Financial Services Applications Financials Marketing Suite Marketing and Sales Suite Order Management Suite Procurement Projects Suite Public Sector Sales Suite Service Suite Supply Chain Management University EDI Interface View Java Open Interface PL/SQL Applications Technology Application Object Library Common Application Calendar

Integration Repository >

Concurrent Program : Transaction Layout Definition Search Printable Page Regenerate

Internal Name: ECRDTLD
Type: Concurrent Program
Product: e-Commerce Gateway
Status: Active
Business Entity: [SQL Transaction Layout Definition Report](#)
Online Help: [See the related online help](#)

Scope: Public
Interface Source: Seeded

Full Description
Reports the layout of a specified transaction data file.

Web Service Redeploy Undeploy Publish
WSDL: <http://ws60066rems.us.oracle.com:8040/webservices/SOAProvider/concurrentprogram/ecrtdtd/wsdl>

Source Information

Parameters

Name	Type	Required	Displayed	Description
Transaction Code	ECE_SRS_DOCUMENT_TYPE_ALL	Yes	Yes	Transaction Code
Include Data Not Mapped	Yes_No	Yes	Yes	Include Data Not Mapped

Methods

Select Object and Create Grant

Select All | Select None

Select Details	Name	Internal Name	Status	Description
<input type="checkbox"/> Show Process	Process	Active	Transaction Layout Definition Report	

Interface types like Concurrent Program and Business Events contain only one method.

- 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 Contains Multiple Methods

ORACLE Integration Repository

Home Logout Preferences Help Diagnostics

Integration Repository > **Service Bean Details : Email Business Object Services**

Search Printable Page Generate

Qualified Name /oracle/apps/ar/hz/service/party/EmailService
 Interface oracle.apps.ar.hz.service.party.EmailService
 Extends oracle.svc.service
 Product Receivables
 Status Active
 Documentation Email Business Object API, Oracle Trading Community Architecture Technical Implementation Guide
 XML Schema EmailService

Scope Private To Application
 Interface Source Seeded

Extracts a particular Email business object from TCA.

Full Description

Extracts a particular Email business object from TCA. You pass the object's identification information to the operation, and it returns the identified business object as it exists in TCA.

Source Information

Source File java/hz/service/party/server/EmailSAM.xml
 Source Version 120.3
 Source Product AR
 Implementation oracle.apps.ar.hz.service.party.server.EmailSAM

Methods

Select Object and Create Grant

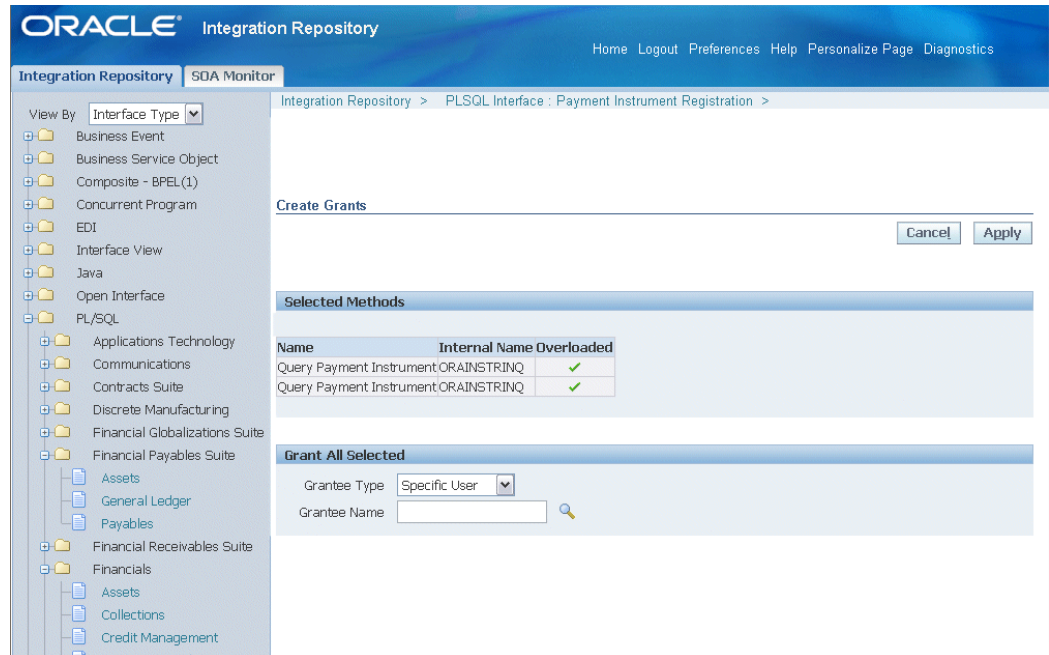
Select All Select None

Select Details	Name	Status	Description
<input type="checkbox"/> Show createEmail	createEmail	Active	Creates a Email business object.
<input type="checkbox"/> Show getEmail	getEmail	Active	Extracts a particular Email business object from TCA.
<input type="checkbox"/> Show saveEmail	saveEmail	Active	Saves a Email business object.
<input type="checkbox"/> Show updateEmail	updateEmail	Active	Updates a Email business object.

Interface types containing multiple methods are PL/SQL, Business Service Object, and Java interfaces.

To create a grant

For example, in the PL/SQL interface details page, select appropriate method name check boxes in the Methods region and click **Create Grant** to open the Create Grants page.



Note: Each overloaded function contained in an API can be uniquely granted to a specific user, user group, or all users through the create grant feature. If you select more than one overloaded function, an Overloaded column appears in the selected methods table indicating more than one overloaded function is selected for the grant.

From here, you can select a grantee type and grantee name if applicable and click **Apply**.

To revoke a grant

In the interface details page, select the Show link for the method that you want to view or revoke the grant. The Grant Details section of the selected method name appears detailing the grantee and grantee type information. Click the **Revoke** icon for the grant that you want to revoke to revoke the grant.

Note: To create and revoke grants, you must log into Oracle Integration Repository with the integration repository administrator role.

Detailed information on how to create, review, and revoke security grants, see Managing Security Grants, page 3-16.

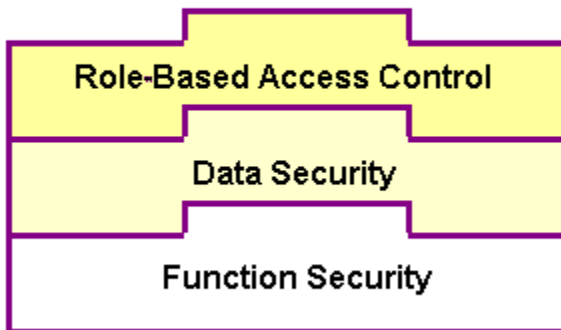
Managing Role-Based Access Control Security

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
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 its 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

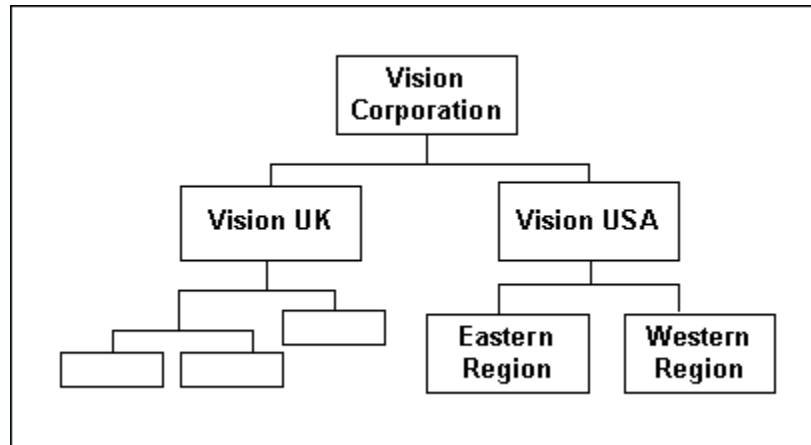
Multiple organizations can be sets of books, business groups, legal entities, operating units, or inventory organizations. You can define multiple organizations and the relationships between them in a single installation of Oracle E-Business Suite.

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.

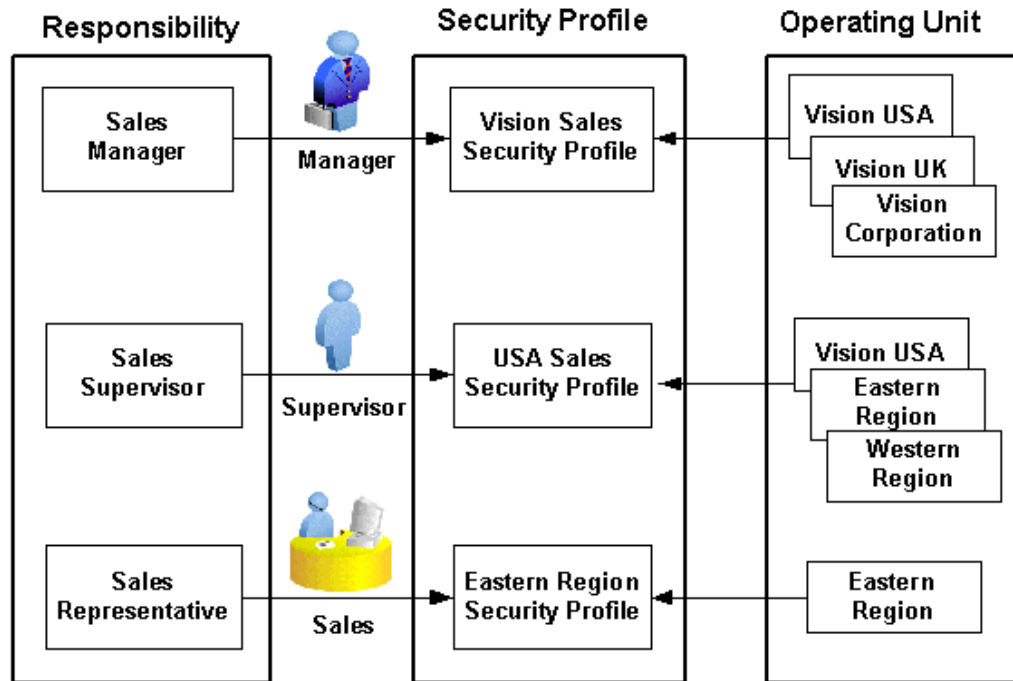
Security profiles are defined based on organization hierarchies. For example, a sales company consists of USA and UK operating units; the USA operating unit has Western Region Sales and East Region Sales. Sales managers are responsible for both USA and UK sales, supervisors are responsible for either 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.

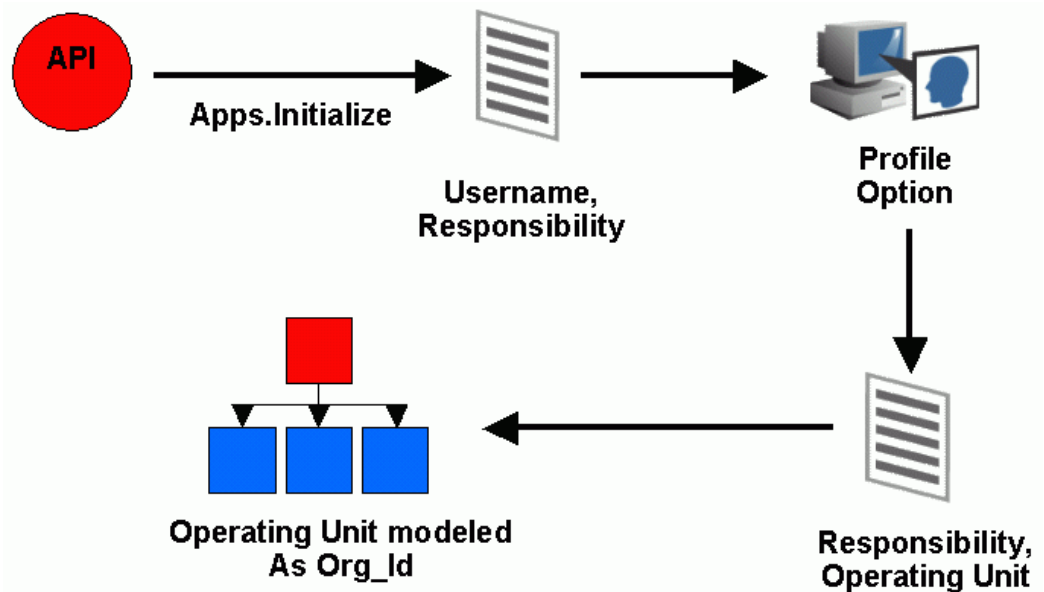
Note: The security profile allows you to assign multiple operating units for the same business group; global security profile allows you to assign multiple operating units across business groups. Based on your HR implementation and how you want to partition the data, you can decide which security profile 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 use the profile options in a multi-organization environment.

Building 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 standard for securing Web services. It describes enhancements to SOAP messages to provide quality protection through message integrity and single message authentication. It also describes how to attach security tokens to SOAP messages to enhance security features.

To secure Web service content and authenticate Web service operation, Oracle E-Business Suite Integrated SOA Gateway supports WS-Security through UsernameToken based security. This security mechanism includes UsernameToken profile which provides username and password information in the security header for a Web service provider to use in authenticating the SOAP request. The authentication is performed by passing a username and password in the SOAP Header of a SOAP message sent to the Web Service Producer.

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).

Please note that the username/password information discussed in this UsernameToken based security model is the concept of Oracle E-Business Suite username/password (or the username/password created through the Users window in defining an application user).

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>Zoe</wsse:Username>
        <wsse:Password>IloveDogs</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-wsswssec
urity-secext-1.0.xsd">
      <wsse:UsernameToken>
        <wsse:Username>Kwalker</wsse:Username>
        <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-t
oken-profile-1.0#PasswordText">welcome</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </env:Header>

  <env:Body>
    ...
  </env:Body>
</env:Envelope>

```

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

See Implementing Service Invocation Framework, page 8-6.

Monitoring and Managing SOAP Messages Using SOA Monitor

This chapter covers the following topics:

- SOA Monitor Overview
- Searching SOAP Requests and Responses
- Viewing SOAP Request Details
- Viewing SOAP Response Details
- Viewing Method Details
- Purging SOAP Requests and Responses
- Enabling and Disabling Web Service Monitoring

SOA Monitor Overview

If the SOA monitoring feature is turned on, all the SOAP requests and responses that SOA Provider and Web Service Provider process can be captured and passed to SOA Monitor where integration repository administrators can monitor and audit these SOAP requests and responses as well as identify errors or exceptions if occur.

Important: To enable the interaction between SOA Provider/Web Service Provider and SOA Monitor, you must turn on the Web service monitoring feature by setting the profile option "SOA: Service Monitor".

SOA Provider also uses this profile option to detect whether the monitoring feature is enabled. If it is enabled, the SOAP message that SOA Provider and Web Service Provider receives will be passed to SOA Monitor.

The administrator can perform the following activities by using SOA Monitor:

- Monitor service operation status
- View service operation details
- View service operation request-response information
- Archive service operation instances
- Monitor service operation status

Accessing SOA Monitor

To access SOA Monitor, log on to Oracle E-Business Suite with the username that is granted with the integration repository administrator role. Select the *Integrated SOA Gateway responsibility* from the navigation menu and then select the SOA Monitor link to open the SOA Monitor search page.

ORACLE Integration Repository

Integration Repository SOA Monitor

Monitor Search

* Indicates required field

Search

* From: 22-Feb-2008 (example: 22-Feb-2008)

To: 23-Feb-2008 (example: 22-Feb-2008)

Show More Search Options

Go Clear

Last Updated : 02-22-2008 10:02:18

Indicates no response

Indicates failed response

Indicates successful response

Request Received	Web Service Name	Operation Name	Username	Host	Status	Details
						Request Response
22-Feb-2008 10:02:05	/oracle/apps/fnd/rep/ws/IntegrationRepositoryService	invokeServiceBatch		11.22.33.44	✓	oo oo
22-Feb-2008 05:02:15				1.2.3.4	✓	oo oo
22-Feb-2008 05:02:15				0.0.0.0	✓	oo oo
22-Feb-2008 10:02:14	/oracle/apps/fnd/rep/ws/IntegrationRepositoryService	invokeServiceBatch		11.22.33.44	✓	oo oo
22-Feb-2008 10:02:43	/oracle/apps/fnd/rep/ws/IntegrationRepositoryService	invokeServiceBatch		11.22.33.44	✓	oo oo

Turn Off Web Service Monitoring Purge

This section includes the following topics:

- Searching SOAP Requests and Responses, page 7-3
- Viewing SOAP Request Details, page 7-5
- Viewing SOAP Response Details, page 7-6
- Viewing Method Details, page 7-8
- Purging SOAP Requests and Responses, page 7-9
- Enabling and Disabling Web Service Monitoring, page 7-13

Searching SOAP Requests and Responses

SOA Monitor provides an audit environment and management tool allowing you to perform searches on saved SOAP requests and responses received through SOA Provider based on a certain period of time.

Note: To accommodate these SOAP messages, SOA Monitor uses the following profile options to buffer sufficient room of possible SOAP requests and responses and then efficiently interacts with SOA Provider to save them in the database as a batch update operation:

- **SOA: Maximum Number of Requests - Responses to Buffer**

This profile option sets the maximum number of Web service requests and responses for SOA Monitor to log before saving them to database. The default value is set to 5.

- **SOA: Maximum Monitor Save Interval (Seconds)**

This profile option sets the maximum save interval in seconds between two consecutive saves for SOA Monitor to pass the logged SOAP messages and save them to the database. The default value is set to 120 seconds.

You can optionally enter more search criteria including username or response status along with a selected time frame for your search. 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 date and time the request was received, Web service name, operation name, username, host IP address, message response status, request details and response details.

Note: The default value in the From field is the last seven days from today; the To field should show the current date. However, you can update these From and To fields if needed to reflect your search.

ORACLE Integration Repository

Integration Repository SOA Monitor

Monitor Search

Indicates required field

Search

* From 22-Feb-2008 To 23-Feb-2008

Show More Search Options

Go Clear

Last Updated : 02-22-2008 10:02:18

Indicates no response
Indicates failed response
Indicates successful response

Request Received	Web Service Name	Operation Name	Username	Host	Status	Request	Response
22-Feb-2008 10:02:05	/oracle/apps/fnd/rep/vs/IntegrationRepositoryService	invokeServiceBatch		11.22.33.44	✓	00	00
22-Feb-2008 05:02:15				1.2.3.4	✓	00	00
22-Feb-2008 05:02:15				0.0.0.0	✓	00	00
22-Feb-2008 10:02:14	/oracle/apps/fnd/rep/vs/IntegrationRepositoryService	invokeServiceBatch		11.22.33.44	✓	00	00
22-Feb-2008 10:02:43	/oracle/apps/fnd/rep/vs/IntegrationRepositoryService	invokeServiceBatch		11.22.33.44	✓	00	00

Turn Off Web Service Monitoring Purge

From the search result page, you can perform the following tasks:

- View status of each audited SOAP request and response
- View SOAP request details by clicking the **Request** icon for a given SOAP request
See: Viewing SOAP Request Details, page 7-5.
- View a corresponding SOAP response details by clicking the **Response** icon for a given SOAP response
See: Viewing SOAP Response Details, page 7-6.
- Purge SOAP requests and responses collected over a period of time by clicking **Purge**
See: Purging SOAP Requests and Responses, page 7-9.
- Enable or Disable Web Service Monitoring feature by clicking **Turn On Web Service Monitoring** or **Turn Off Web Service Monitoring**
See: Enabling and Disabling Web Service Monitoring, page 7-13.

To perform a search:

1. Log on to Oracle E-Business Suite with the username that is granted with the integration repository administrator role. Select the Integrated SOA Gateway responsibility. From the navigation menu, select the SOA Monitor link to open the SOA Monitor search page.
2. Enter appropriate information in the From and To fields 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:
 - Username: Search and select an appropriate username.
 - Response Status: Select an appropriate value from the drop-down list including 'Any (default)', 'Failed', and 'Success'.

Click **Go** to execute your search.

4. All SOAP requests and responses that match your search criteria appear.
5. Click the **Request** icon for a given message to view request details.
6. Click the **Response** icon for a given message to view SOAP response details.
7. Click **Purge** to purge all SOAP requests and responses listed in the table.

Viewing SOAP Request Details

After a search, you can view the SOAP request details by clicking the **Request** icon for a given message listed in the search result table. The Web Service Request Details page appears. From this page, you can find the SOAP request information including Web service name, username, responsibility, number of attachments, date and time the request was received, SOAP request XML information, host IP address, interface details, and method names.

SOAP Request Details Page

The screenshot displays the 'SOAP Request Details Page' within the Oracle Integration Repository. The page header includes the Oracle logo and 'Integration Repository', with navigation links for 'Diagnostics', 'Home', 'Logout', and 'Preferences'. A breadcrumb trail shows 'SOA Monitor > Response Details > Method Details >'. The main section, titled 'Request Details', contains a table with the following information:

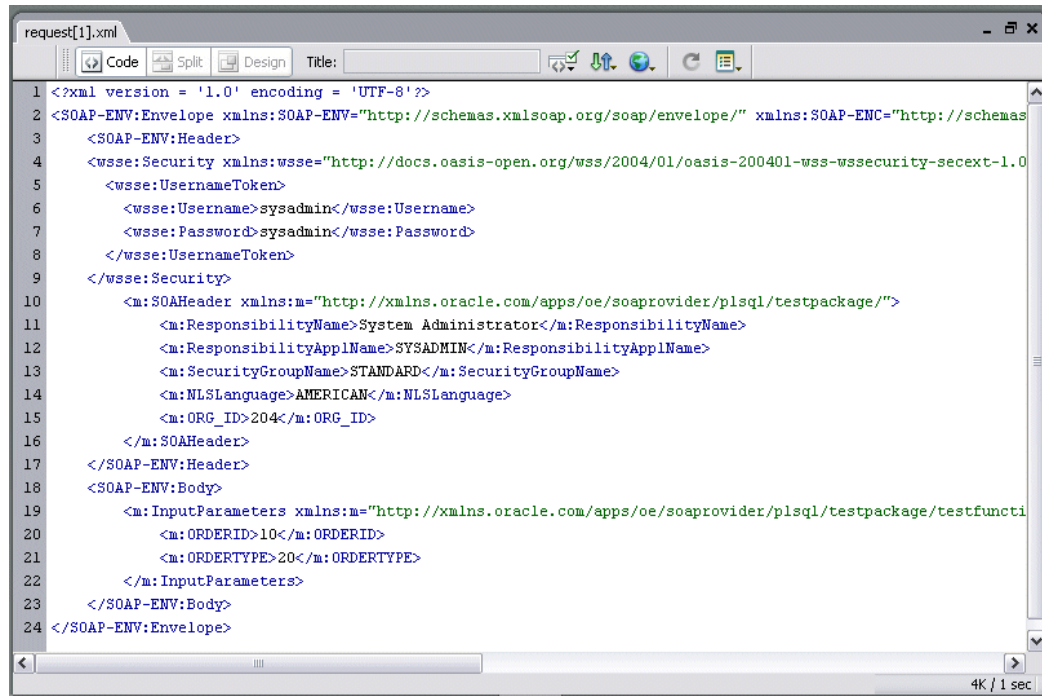
Web Service Name	TESTPACKAGE	Date	Feb 26, 2008 08:02:15
Username	SYSADMIN	Host	ap690ses.us.oracle.com
Responsibility		Interface	TESTPACKAGE
Number of Attachments	0	Method Called	PLSQL:TESTPACKAGE:TESTFUNCTION
Soap Request	View		

The footer of the page includes links for 'About this Page' and 'Privacy Statement', and a copyright notice: 'Copyright (c) 2006, Oracle. All rights reserved.'

If the SOAP request has attachments associated with it, the Attachment region will appear to include all attachment names and MIME Type information.

Additionally, you can view and save the actual XML file of this request in a pop-up window by clicking the SOAP Request **View** link.

SOAP Request XML File



```
1 <?xml version = '1.0' encoding = 'UTF-8'?>
2 <SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/' xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap-enc/'>
3   <SOAP-ENV:Header>
4     <wsse:Security xmlns:wsse='http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0'>
5       <wsse:UsernameToken>
6         <wsse:Username>sysadmin</wsse:Username>
7         <wsse:Password>sysadmin</wsse:Password>
8       </wsse:UsernameToken>
9     </wsse:Security>
10    <m:SOAHeader xmlns:m='http://xmlns.oracle.com/apps/oe/soapprovider/plsql/testpackage/'>
11      <m:ResponsibilityName>System Administrator</m:ResponsibilityName>
12      <m:ResponsibilityApplName>SYSADMIN</m:ResponsibilityApplName>
13      <m:SecurityGroupName>STANDARD</m:SecurityGroupName>
14      <m:NLSLanguage>AMERICAN</m:NLSLanguage>
15      <m:ORG_ID>204</m:ORG_ID>
16    </m:SOAHeader>
17  </SOAP-ENV:Header>
18  <SOAP-ENV:Body>
19    <m:InputParameters xmlns:m='http://xmlns.oracle.com/apps/oe/soapprovider/plsql/testpackage/testfunction'>
20      <m:ORDERID>10</m:ORDERID>
21      <m:ORDERTYPE>20</m:ORDERTYPE>
22    </m:InputParameters>
23  </SOAP-ENV:Body>
24 </SOAP-ENV:Envelope>
```

To view SOAP request details:

1. Log on to Oracle E-Business Suite with the username that is granted with the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

From the navigation menu, select the SOA Monitor link to open the SOA Monitor search page.
2. Perform a search to display search results. See: Searching SOAP messages, page 7-3.
3. Click the **Request** icon for a given request to view SOAP request details. The Web Service Request Details page appears allowing you to view the request details.
4. If there is any attachment associated with it, you can find attachment information listed in the Attachment region.
5. Click the SOAP Request **View** link to view the actual XML file displayed in a pop-up window.

Viewing SOAP Response Details

Similar to the view request details, SOA Monitor allows you to view a corresponding response details.

From the search result table, click the **Response** icon for the response that you want to view. This opens the Web Service Response Details page where you can find the SOAP response information including Web service name, username, responsibility, number of attachments, execution time, status, date and time the request was received, SOAP response XML information, host IP address, interface details, and method names.

In addition to the header region, you can find the following regions displayed in the details page if certain conditions are met:

- **Error Information:** If the response has failure status, the Error Information region appears to show the error description and details.
- **Method:** If the value of the Method Called field is 'Invoke Service Batch', then the Method region appears. The relevant method information including method name, status, and details is listed in a table.

You can drill down to see each method details by clicking the **Details** icon for the method you want to view. This opens the Method Details page. If the method status is 'Failed', then the Error Information region appears letting you to view the error details.

See: Viewing Method Details, page 7-8

- **Attachment:** If the SOAP response has attachments associated with it, the Attachment region will appear to list the attachment details including all attachment names and MIME Type information.

Additionally, you can view the actual XML file of this response in a pop-up window by clicking the SOAP Response **View** link.

SOAP Response Details Page

The screenshot displays the 'SOAP Response Details Page' in the Oracle Integration Repository. The page header includes the Oracle logo and navigation links like 'Diagnostics', 'Home', 'Logout', and 'Preferences'. Below the header, the 'Response Details' section provides metadata about the response, such as the web service name, username, responsibility, number of attachments, execution time, and status. A 'Method' section contains a table listing individual methods and their execution status.

Method Name	Status	Details
getDataObjectDescription	Success	⌵
getInterfaceClassName	Failed	⌵

To view SOAP response details:

1. Log on to Oracle E-Business Suite with the username that is granted with the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

From the navigation menu, select the SOA Monitor link to open the SOA Monitor search page.

2. Perform a search to display search results. See: Searching SOAP messages, page 7-3.
3. Click the **Response** icon for a given message to view SOAP response details. The Web Service Response Details page appears allowing you to view the response details.
4. If there is any attachment associated with it, you can find attachment information listed in the Attachment region.
5. If this response status is 'Failed', then you will find the error details in the Error Information region.
6. If the value of the Method Called field is 'Invoke Service Batch', then you will find the Method region appears letting you to view method information.

You can click the **Details** icon for the method that you want to view to see the method details.
7. Click the SOAP Response **View** link to view the actual XML file displayed in a pop-up window.

Viewing Method Details

If you have method 'Invoke Service Batch' called in a SOAP response message, the Method region appears in the Response Details page allowing you to view relevant method information including method name and status.

Additionally, you can drill down to see each method details by clicking the **Details** icon in the Method region for the method you want to view. This opens the Method Details page for your selected method where you can find the method details including Web service name, username, date, interface name, and status information.

If the method status is 'Failed', then the Error Information region appears letting you to view the error details.

Method Details Page

ORACLE[®] Integration Repository

SOA Monitor > Response Details >

Method Details

Web Service Name	/oracle/apps/fnd/rep/ws/IntegrationRepositoryService	Interface	/oracle/apps/fnd/rep/ws/IntegrationRepositoryService
Username		Method Name	getInterfaceClassByName
Date	Feb 22, 2008 10:02:43	Status	FAILED

Error Information

Error Description	Unknown Error
Error Details	Unknown Error

Diagnosics Home Logout Preferences

About this Page Privacy Statement Copyright (c) 2006, Oracle. All rights reserved.

To view SOAP response method details:

1. Log on to Oracle E-Business Suite with the username that is granted with the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

From the navigation menu, select the SOA Monitor link to open the SOA Monitor search page.
2. Perform a search to display search results. See: Searching SOAP messages, page 7-3.
3. Click the **Response** icon for a given message to view SOAP response details. The Web Service Response Details page appears allowing you to view the response details.
4. If the value of the Method Called field is 'Invoke Service Batch', then you will find the Method region appears letting you to view method information.
5. Click the **Details** icon for the method that you want to view to see the method details.

Purging SOAP Requests and Responses

Oracle E-Business Suite Integrated SOA Gateway allows you to purge SOAP requests and responses 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

The screenshot shows the 'Name' window of the 'Purge Schedule Request' in the Oracle Integration Repository. The breadcrumb trail indicates the current step is 'Name', followed by 'Parameters', 'Schedule', 'Layout', 'Notifications', 'Printing', and 'More'. The 'Schedule Request: Name' section includes a note that an asterisk indicates a required field. The 'Program Name' is 'Purge Obsolete SOA Monitor Data' and the 'Request Name' is 'Demo1'. Below this is a 'Language Settings' section with 'Select All' and 'Select None' options. A table lists language settings with columns for 'Language', 'Territory', and 'Numeric Character'. The 'Language' column has checkboxes for 'American English' (checked), 'Arabic', and 'Korean'. The 'Territory' column has dropdowns for 'AMERICA', 'UNITED ARAB EMIRATES', and 'KOREA'. The 'Numeric Character' column has dropdowns for '.,', '.,', and '.,'. At the bottom, there are 'Cancel', 'Submit', and 'Next' buttons, and a 'Step 1 of 7' indicator.

Select	*Language	*Territory	*Numeric Character
<input checked="" type="checkbox"/>	American English	AMERICA	.,
<input type="checkbox"/>	Arabic	UNITED ARAB EMIRATES	.,
<input type="checkbox"/>	Korean	KOREA	.,

- **Parameters:** You must enter the Start Date and End Date fields to identify the date range for your purge.

Purge Schedule Request: Parameters Window

The screenshot shows the 'Parameters' window of the 'Purge Schedule Request' in the Oracle Integration Repository. The breadcrumb trail indicates the current step is 'Parameters', followed by 'Name', 'Schedule', 'Layout', 'Notifications', 'Printing', and 'More'. The 'Schedule Request: Parameters' section includes a note that an asterisk indicates a required field. The 'Program Name' is 'Purge Obsolete SOA Monitor Data' and the 'Request Name' is 'demo1'. Below this are two required fields: '* Start Date' with a value of '01-Mar-2008 14:04:00' and '* End Date' with a value of '04-Mar-2008 14:04:04'. At the bottom, there are 'Cancel', 'Submit', and 'Back' buttons, and a 'Step 2 of 7' indicator.

- **Schedule:** Specify when you would like your request to be run.

Purge Schedule Request: Schedule Window

The screenshot shows the 'Purge Schedule Request: Schedule Window' in the Oracle Integration Repository. The window has a blue header with the Oracle logo and 'Integration Repository' text. On the right, there are links for 'Diagnostics', 'Home', 'Logout', 'Preferences', and 'Help'. Below the header is a navigation bar with tabs: 'Name', 'Parameters', 'Schedule' (which is active), 'Layout', 'Notifications', 'Printing', and 'More'. The main content area is titled 'Schedule Request: Schedule' and contains the following sections:

- Program Name:** Purge Obsolete SOA Monitor Data
- Request Name:** demo1
- Schedule:** A section with a 'Specify when you would like your request to run.' label and an 'Advanced Schedule' button.
- Start Date:** Radio buttons for 'As soon as possible' (selected) and 'Start at specific date and time'. Below are fields for 'Start Date' (05-Mar-2008) and 'Start Time' (08:05 AM).
- Recurrence:** Radio buttons for 'Never repeat' and 'Repeat'. Below are fields for 'Every' (1) 'Days', 'End Date', and 'End Time' (00:00 AM).
- Apply the Interval:** Radio buttons for 'From the completion of the Prior run' and 'From the start of the Prior run' (selected).
- Schedule Name:** A text field with a search icon and a checkbox for 'Increment Date Parameters' (labeled 'Recurring request date parameters will be incremented by the repeat interval').
- Save Schedule:** A checkbox for 'Save Schedule' and fields for 'Schedule Name' and 'Description'.

At the top right of the main content area, there are buttons: 'Cancel', 'Submit', 'Back', 'Step 3 of 7', and 'Next'.

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.

Instead of creating a new schedule for your purge, you can use a previously saved schedule by retrieving the schedule name in the Schedule Name region. Click the **Increment Date Parameters** check box to make the selected schedule become repeatable with the recurrence intervals specified in the Recurrence region.

To reuse a schedule, click the **Save Schedule** check box, and specify the name and description.

- **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

Program Name: Purge Obsolete SOA Monitor Data
Request Name: demo1

Select Recipients: Delete

Select All | Select None

Select Recipients	Normal	Warning	Error
<input type="checkbox"/> A. Banerjee	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add Another Row

Cancel Submit Back Step 5 of 7 Next

About this Page Privacy Statement

Diagnostics Home Logout Preferences Help

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

- **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 and responses during your specified date range.

The audited SOAP requests and responses stored in the following four 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_RESPONSE

This deletes all SOAP responses for the specified date range.

3. Purging FND_SOA_BODY_PIECE

This deletes the SOAP body pieces corresponding to those SOAP requests or responses that have been purged (for the specified date range).

4. Purging FND_SOA_RESPONSE_METHOD

This deletes all SOAP response methods for the specified date range.

5. Purging FND_SOA_ATTACHMENT

This deletes all attachments associated with the SOAP requests and responses for the specified date range.

To purge SOAP requests and responses:

1. Log on to Oracle E-Business Suite with the username that is granted with the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

From the navigation menu, select the SOA Monitor link to open the SOA Monitor search page.
2. Click **Purge** to launch a concurrent program.
3. Enter the following information in the Schedule Request window:
 1. The current 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 Monitoring

SOA Provider can automatically detect whether the SOA Monitor feature is enabled or not by checking a profile value set in the 'SOA: Service Monitor' profile option. If it is enabled, then all incoming SOAP requests and corresponding responses that SOA Provider receives can be logged or saved in SOA Monitor.

In addition to the profile option, integration repository administrators can enable or disable the monitoring feature directly through the SOA Monitor UIs. By clicking **Turn On Web Service Monitoring** or **Turn Off Web Service Monitoring** in the Monitor

Search page will override the 'SOA: Service Monitor' profile value to enable or disable the feature.

For more information about this profile option, see Setting Profile Options, page 2-6.

ORACLE Integration Repository

Integration Repository SOA Monitor

Monitor Search

Turn Off Web Service Monitoring Purge

* Indicates required field

Search

From 22-Feb-2008 To 23-Feb-2008

Show More Search Options

Go Clear

Last Updated : 02-22-2008 10:02:18

Indicates no response
Indicates failed response
Indicates successful response

Request Received	Web Service Name	Operation Name	Username	Host	Status	Request	Response
22-Feb-2008 10:02:05	/oracle/apps/fnd/rep/vs/IntegrationRepositoryService	invokeServiceBatch		11.22.33.44	Success	00	00
22-Feb-2008 05:02:15				1.2.3.4	Success	00	00
22-Feb-2008 05:02:15				0.0.0.0	Success	00	00
22-Feb-2008 10:02:14	/oracle/apps/fnd/rep/vs/IntegrationRepositoryService	invokeServiceBatch		11.22.33.44	Success	00	00
22-Feb-2008 10:02:43	/oracle/apps/fnd/rep/vs/IntegrationRepositoryService	invokeServiceBatch		11.22.33.44	Success	00	00

Turn Off Web Service Monitoring Purge

To enable or disable the Web Service Monitoring feature:

1. Log on to Oracle E-Business Suite with the username that is granted with the integration repository administrator role. Select the Integrated SOA Gateway responsibility.

From the navigation menu, select the SOA Monitor link to open the SOA Monitor search page.

2. Click **Turn On Web Service Monitoring** or **Turn Off Web Service Monitoring** in the Monitor Search page to enable or disable the feature.
3. A confirmation message appears indicating the feature is enabled or disabled.

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 WSDL-described service to be invoked.

Note: To have the Web service invocation feature work properly, Oracle E-Business Suite Integrated SOA Gateway has dependency on the Oracle Application Server WSIF and Oracle Workflow Java Business Event System.

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.

This invocation framework used in Oracle E-Business Suite allows updated implementations of a binding to be plugged into WSIF at run time. As a result, it not only facilitates a stubless or completely dynamic Web service invocation, but also allows the calling service to defer choosing a service binding until run time. More importantly, this enhances the seamless business integration between loosely coupled applications and accelerates service execution and consumption.

Note: WSIF is a simple Java API for invoking Web services. It is

supported by Oracle 10g Application Server (Oracle 10gAS) 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 Release 3 Patch Set 4 (10.1.3.4.0). See *Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12*, My Oracle Support Knowledge Document 556540.1 for details.

Please note that the service invocation framework discussed here only supports document-based Web service invocation. Oracle E-Business Suite Integrated SOA Gateway does not support RPC (remote procedure call) style Web service invocation through this invocation framework.

Note: The document-based Web service 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 8-2
- Understanding Service Invocation Framework Major Features, page 8-5
- Implementing Service Invocation Framework, page 8-6

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.

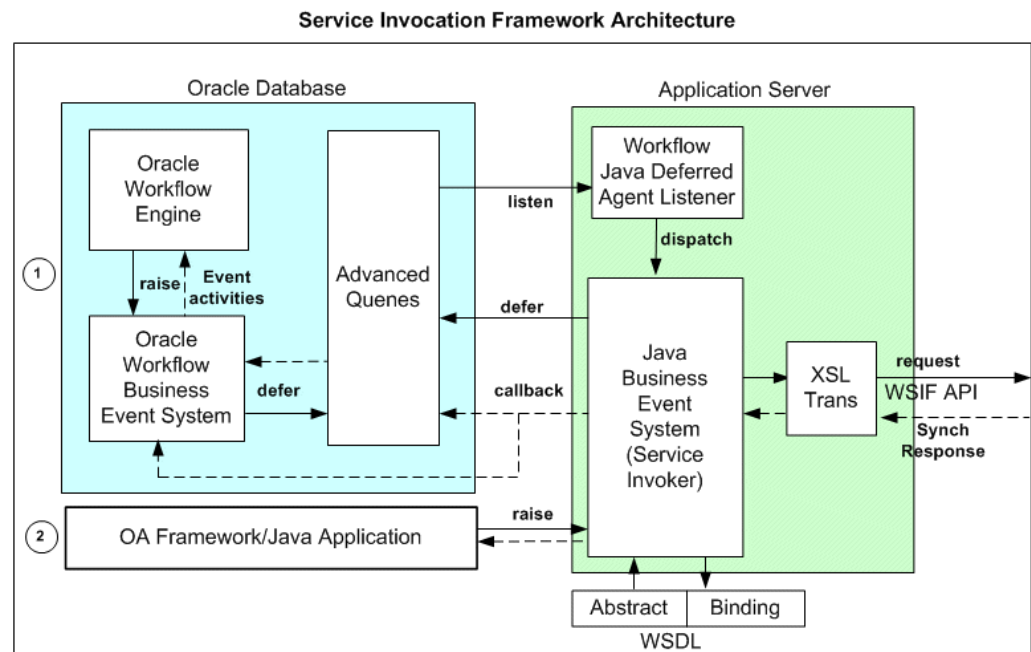
For example, to invoke a Web service through Oracle Workflow JBES, the description of WSDL URL representing the Web service must be consumed through the event subscription definition so that Web service metadata can be parsed and stored as subscription parameters.

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:



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 Manager within the call to the `WF_EVENT.Raise` API, or the Event Manager can obtain the event data or

message payload by calling the generate function for the event if the data or payload is required for a subscription.

Note: See *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 enqueueing 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 8-6
- Setup Tasks for Invoking SSL-based Web Services Over HTTPS, page 8-8
- Implementing Service Invocation Framework, page 8-12

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 *.us.oracle.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 8-8
- Setup SSL Proxy Host and Port, page 8-11

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:

- **Use openssl Utility:**

```
$ openssl s_client -connect <server>:<port> -showcerts
```

For example: `$ openssl s_client -connect host.domain.com:443 -showcerts`

A sample output of above **openssl** command can be like:

```

Server certificate
-----BEGIN CERTIFICATE-----
MIIFVjCCBD6gAwIBAgIQBVWzfUyIcCa5LtuV+f9WvjANBgkqhkiG9w0BAQUFADCB
sDELMakGA1UEBhMCVVMxMzFzAVBGRVNBAAOTDlZlcm1TaWduLCBjbmuMR8wHQYDVQQL
ExZWZlXJpU2lnbiBucnVzdCB0ZXR3b3JrMTswOQYDVQQLZXJUZlJtcyBvZiB1c2Ug
YXQgaHR0cHM6Ly93d3cudmVyaXNpZ24uY29tL3JwYSAoYkNwNTEqMCGA1UEAaMh
VmVyaXNpZ24gQ2xhc3MgMyBTZW51cmUgU2VydmVyIENBMB4XDTA5MDQyMTAwMDAw
MfoXDTEwMDUwNTIzNTk1OVowgckxXzAJBgNVBAYTALVTRMRWEQYDVQQIEwpDYWxp
MzYybmlhMRCwFjYDVQVHFA5SZWR3b3b29kIFNob3JlcE9mbGkGA1UEChQST3JhY2x1
IENvcnBvcnF0aW9uMR8wHQYDVQQLFBZjbmZvcmlhdGlvbiBUZWNobm9sb2d5MTMw
MQYDVQQLFjFjYXZlcm1TaWduLmNvb3Y9cGEgKGMp
MDUxGTAXBgNVBAMUECoub3JhY2x1Y29ycC5jb20wgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBAL/EBxxt2keWtuJbo4SogWmiaJxThYDMvy8nWkpVkiP3s7OCQW0G
t17sAirfbkUirbGRlcWi5fi0RReruGXgYxFnfl2fBNaimRWVo3mjeQo8BpRBM27n
3YcTZUlaIE77FvB3913jzD9c4sbjIe2fGpVmx+X9PZmDKSY9KPGjDbFNAGMBAAGj
ggHTMIIBZzAJBgNVHRMEAIAAMAsGA1UdDwQEAwIFoDBEBGNVHR8EPTA7MDmgN6A1
hjNodHRwOi8vU1ZSU2VjdXJlLWVybC52ZXJpc2lnbi5jb20vU1ZSU2VjdXJlMjAw
NS5jcmwwRAYDVROgBD0wOzA5BgtghkgBhvFAQcXAZAqMCGGCCsGAQUFBwIBFhxo
dHRwc3ovL3d3dy52ZXJpc2lnbi5jb20vcnBhMB0GA1UdJQQWMBQGCCsGAQUFBwMB
BggrBgEFBQcDAjAFBgNVHSMGDAWgBRv7K+g3Yqk7/UqEGctPlWCvNfvJTB5Bggr
BgEFBQcBAQRtMGswJAYIKwYBBQUHMAAGGGH0dHA6Ly9vY3NwLnZlcm1zaWduLmNv
bTBDBDggBgEFBQcwAoY3aHR0cDovL1NWU1NlY3VyZS1haWEudmVyaXNpZ24uY29t
L1NWU1NlY3VyZTIwMDUyYWIhLmNlcjBuBggrBgEFBQcBDARiMGChXqBcMFowWDBW
FglpbWFnZS9naWYwITAFMacGBSs0AwIaBBRLa7kolgYMu9BS0JsprEsHiyEFGDAm
FiRodHRwOi8vbG9nb352ZXJpc2lnbi5jb20vbnNsb2dvM5naWYwDQYJKoZIhvcN
AQEFBQADAgEBAD69NfljQLuDT2Nol3pmQ1717rc8kKmpLEyO6u5MxIK0+L2Ms1V
4NElqNxd1dfIoW68HHXtPS5KtKfLYk9EOokBd7oMpFv31RANV3LpdAHZC9EaK
cA/cOkB2RrSu7ZmaUovRb+3v5FdhAmgtOy6Wlj0yxMvXVf/TOeXqK18C/r1gSzyC
s/jVmy6N810e1eqtozzt/aJNGu7xu/MdtP13eyu7RSEBRGJwEwTXH+rTUKK8mle0
Kz15DgJ6ByK2XZmD4Z+O8DTUhUhiHR1OhuLR7zjGp9W7wQuCizUcTvuKEGzVf5D/
y7orhV0U+AOXn1/5wntVMZc/Tmqr/Fkb8+g=
-----END CERTIFICATE-----

```

- **Use Web Browser:**

Access the WSDL file available through HTTPS URL (such as `https://<hostname>:<port>/OA_HTML/XMLGatewayWSDL`) 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 8-6.

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 changeit  
-alias xabbott_bugdbcert -file my_cert.cer
```

Note: This must be typed as a single line.

The default value of StoreKey Password (-storepass) is "changeit" The file (-file) is the exported certificate file i.e. 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 Manager (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_appsjureopts">="-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>
```

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 8-13
 - WFBES_SOAP_USERNAME
 - WFBES_SOAP_PASSWORD_MOD
 - WFBES_SOAP_PASSWORD_KEY
- Message transformation parameters to support XSL transformation, page 8-15
 - 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 8-14
- Setting Other Web Service Input Message Parts, page 8-16

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 `ReceiveDocument` in Oracle E-Business Suite generic XML Gateway service (`http://hostname:port/OA_HTML/XMLGatewayWSDL`) as described below.

```
<definitions
targetNamespace="http://xmlns.oracle.com/oracle/apps/fnd/XMLGateway"
>
...
<message name="ReceiveDocument_Request">
  <part name="header" element="tnsl:XMLGateway_Header"/>
  <part name="body" element="tnsl:ReceiveDocument"/>
</message>
...
<binding name="XMLGatewaySOAPBinding" type="tns:XMLGatewayPortType">
<soap: binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="ReceiveDocument">
    <soap:operation
soapAction="http://hostname:port/webservices/AppsWSProvider/oracle/a
pps/fnd/XMLGateway"/>
    <input>
      <soap:header message="tns:ReceiveDocument_Request" part="header"
use="literal"/>
      <soap:body parts="body" use="literal"/>
    </input>
    <output>
      <ssoap:body use="literal"/>
    </output>
  </operation>
</binding>
...
</definitions>
```

The operation `ReceiveDocument` requires input message `ReceiveDocument_Request`, which has two parts:

- **Body:** The value of `ReceiveDocument` type to be set as SOAP body is sent as business event payload.
- **Header:** The value of `XMLGateway_Header` 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:XMLGateway_Header\n" +
    xmlns:ns1=\"http://xmlns.oracle.com/apps/fnd/XMLGateway\" \" +
    \"env:mustUnderstand=\"0\" \"\n" +
    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\" +
    \"\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:XMLGateway_Header>\n\";\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_CALLBACK_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-Business 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-Business 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

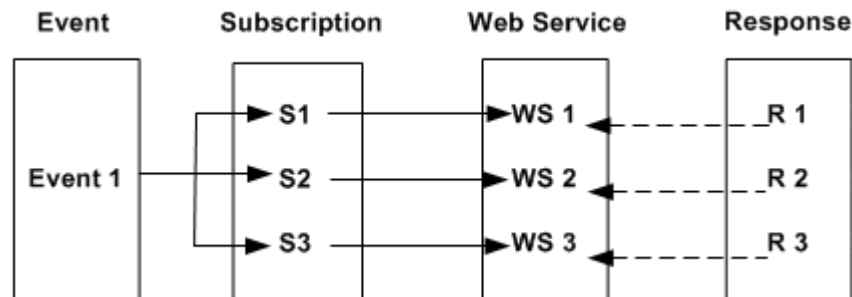
The service invocation framework supports invoking only document-based Web services. The RPC (remote procedure call) style remote Web service invocation is not supported in this release.

- 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.

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 E-Business Suite 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 with the username that has the User Management responsibility access privilege.
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 field 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 with the username that 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_request` event name for the test).

Prerequisites:

Before running this diagnostic test, ensure the following information is in place:

- An event (`oracle.apps.wf.sif.test_fnd_request`) has been created, and a subscription to invoke `FND_REQUEST` service for operation "Submit_Request" has also been defined.
- If the grant for `FND_REQUEST.Submit_Request` has been given to the SOAP username `SYSADMIN` only, then ensure that the password corresponding to the username `SYSADMIN` has been stored in DB vault using a PL/SQL script `$FND_TOP/sql/afvltput.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=ConfirmReceipt`.

For example, a PL/SQL script can be like:

```
sqlplus apps/***db $FND_TOP/sql/afvltput.sql PO Acme_ConfirmReceipt  
<Sysadmin Pwd>
```

For more information on how to invoke Web services using the service invocation framework, see *Implementing Service Invocation Framework*, page 8-6.

Input Required:

WSDL URL: You can enter a valid WSDL URL if you want to override the default.

It checks and reports the test result if it gets a correct string response.

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.

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 predefined 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.

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.

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.

Index

A

Administering Composite Services
 downloading composite services, 4-4
 viewing composite services, 4-3

C

create user account
 configure user in technology stack, 2-5
 create account steps, 2-3
 grant UMX\APPS_SCHEMA_CONNECT
 role, 2-4
Custom Integration Interfaces
 Administrative Tasks
 Create Grants, 5-15
 Deploy Web Services, 5-16
 Download Custom Composite Services,
 5-17
 Generate Web Services, 5-15
 Subscribe to Custom Business Events, 5-
 16
 overview, 5-1
 setup and use parser, 5-3

D

Diagnostic Tests
 Executing Tests, A-1
 Overview, A-1
 Tests, A-3

I

Implementing and Administering Composite
Services

 administering composite services, 4-3
 overview, 4-1
 understanding enablement process, 4-1

Implementing and Administering Native
Services

 Administering Native Services, 3-2
 overview, 3-1

Implementing Service Invocation Framework
Implementation tasks, 8-6

 Overview, 8-1

Implementing Service Invocation Framework
Overview

 major feature, 8-5
 SIF architecture overview, 8-2

Implementing Service Invocation Framework
Tasks

 implementation tasks, 8-12
 Setup tasks, 8-6
 SSL Setup tasks, 8-8

M

Managing SOAP Messages Using SOA Monitor
 using SOA Monitor UIs, 7-1

O

Oracle E-Business Suite Integrated SOA Gateway
 component features, 1-3
 Major Features, 1-2
 native service enablement architecture
 overview, 1-5

Overview, 1-1

S

Securing Web Services

function security, 6-1

moac security, 6-8

overview, 6-1

Role-Based Access Control security, 6-5

Web Service Security, 6-12

service enablement architecture

Design Time, 1-7

Run Time, 1-9

Setting Up

ASADMIN, 2-1

create user account, 2-2

overview, 2-1

profile options, 2-6

U

understanding composite services, 4-1

using SOA Monitor UIs

method details, 7-8

purge, 7-9

search, 7-3

view SOAP request, 7-5

view SOAP response, 7-6

Web service monitoring, 7-13

W

Web service

deploying and undeploying, 3-10

generating, 3-4, 3-16

subscribing to events, 3-14