

Oracle® E-Records

API User's Guide

Release 11*i*

Part No. B15914-01

January 2005

Oracle E-Records API User's Guide, Release 11i

Part No. B15914-01

Copyright © 2003, 2005 Oracle Corporation. All rights reserved.

Primary Author: Michele-Andrea Fields

Contributors: Bryan So, Swaminathan.x Subramanian

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle*MetaLink*, Oracle Store, Oracle8, Oracle8i, and SQL*Plus are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	vii
Preface.....	ix
Audience for This Guide.....	ix
How To Use This Guide	ix
Documentation Accessibility	xi
Other Information Sources	xi
Online Documentation.....	xi
Related User’s Guides.....	xii
Guides Related to All Products	xii
User Guides Related to This Product	xii
Installation and System Administration	xvii
Other Implementation Documentation.....	xix
Training and Support.....	xx
Do Not Use Database Tools to Modify Oracle Applications Data.....	xxi
About Oracle.....	xxi
Your Feedback	xxiii
1 API Introduction	
Introducing the Oracle E-Records APIs	1-2
Oracle E-Records API Support Policy	1-3
Technical Requirements	1-3
Technical Overview.....	1-3
Basic Business Needs.....	1-3

Input Data Sources	1-4
Wrapper Function.....	1-5
Stored Procedure.....	1-5
Major Features	1-6
Oracle E-Records API Bill of Materials	1-10
2 Oracle E-Records API Usage	
Calling the API Interface Code	2-2
API Call Hints	2-2
3 Technical Overview	
Structure for Oracle E-Records Public APIs	3-2
Standard Parameters.....	3-5
Value-ID Conversion.....	3-6
4 Business Objects	
Evidence Store APIs	4-1
Open Document	4-1
Close Document.....	4-2
Update Document.....	4-2
Change Document Status	4-3
Post Document Parameters	4-3
Cancel Document.....	4-4
Get Document Details	4-4
Request Signature	4-5
Capture Signature.....	4-5
Post Signature.....	4-7
Post Signature Parameters.....	4-7
E-Record Event APIs	4-8
Validate E-record	4-8
Validate Payload	4-8
Validate Payload Forms.....	4-9
Get Event Details	4-9
Raise ERES Event	4-9

Raise Inter Event	4-10
Get E-record ID	4-11
Event Relationship APIs	4-12
Create Relationship	4-12
Validate Relationship	4-13
ERES Transaction Acknowledgement	4-14
Send Acknowledgment	4-14
General Purpose APIs	4-14
Get Psig Status	4-15
Is E-signature Required	4-15
Is E-record Required	4-15
Get Query ID On Events	4-16
Get AME Rule Variable Values	4-16
Display Date	4-17
Display Date Only	4-17
Display Time Only	4-17
Is Audit Value Old	4-17
Get Notification Routing Information	4-18
Get Descriptive Flexfield One Prompt	4-18
Get Descriptive Flexfield All Prompts	4-19
Get Lookup Meaning	4-20
Get Query ID On Parameters	4-20
APIs for Forms-Based Applications	4-21
Add Transaction	4-21

A Messages and Errors

Handling Messages	A-1
Interpreting Error Conditions	A-3
Understanding Error Messages	A-3

Glossary

Index

Send Us Your Comments

Oracle E-Records API User's Guide, Release 11i

Part No. B15914-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appsdoc_us@us.oracle.com
- FAX: 650-506-7200 Attn: Oracle Process Manufacturing
- Postal service:

Oracle Corporation
Oracle Process Manufacturing Documentation
500 Oracle Parkway
Redwood City, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Audience for This Guide

Welcome to Release 11*i* of the *Oracle E-Records API User's Guide*.

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

- The principles and customary practices of your business area.
- Oracle E-Records APIs
- The Oracle Applications graphical user interface.

To learn more about the Oracle Applications graphical user interface, read the *Oracle Applications User's Guide*.

See Other Information Sources for more information about Oracle Applications product information.

How To Use This Guide

This guide contains the information you need to understand and use Oracle E-Records APIs.

- Chapter 1 describes how APIs are used, the basic business need of APIs, and the different E-Records APIs offered.
- Chapter 2 describes how to use the E-Records APIs.
- Chapter 3 describes the technical aspect of the APIs.
- Chapter 4 describes the business objects for each API.
- Appendix A describes messages and error codes.

- A Glossary provides definitions of terms that are used in this guide.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. 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. Standards will continue to evolve over time, and Oracle Corporation 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 additional information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

JAWS, a Windows screen reader, 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, JAWS may not always read a line of text that consists solely of a bracket or brace.

Other Information Sources

You can choose from many sources of information, including online documentation, training, and support services, to increase your knowledge and understanding of Oracle Process Manufacturing System Administration.

If this guide refers you to other Oracle Applications documentation, use only the Release 11*i* versions of those guides.

Online Documentation

All Oracle Applications documentation is available online (HTML or PDF).

- **Online Help** - The new features section in the HTML help describes new features in 11*i*. This information is updated for each new release of Oracle Process Manufacturing System Administration. The new features section also includes information about any features that were not yet available when this guide was printed. For example, if your administrator has installed software from a mini-packs an upgrade, this document describes the new features. Online help patches are available on *OracleMetaLink*.

- **11i Features Matrix** - This document lists new features available by patch and identifies any associated new documentation. The new features matrix document is available on *OracleMetaLink*.
- **Readme File** - Refer to the readme file for patches that you have installed to learn about new documentation or documentation patches that you can download.

Related User's Guides

Oracle E-Records APIs share business and setup information with other Oracle Applications products. Therefore, you may want to refer to other user's guides when you set up and use Oracle E-Records APIs.

You can read the guides online by choosing Library from the expandable menu on your HTML help window, by reading from the Oracle Applications Document Library CD included in your media pack, or by using a Web browser with a URL that your system administrator provides.

If you require printed guides, you can purchase them from the Oracle Store at <http://oraclestore.oracle.com>.

Guides Related to All Products

Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI) available with this release of Oracle Process Manufacturing System Administration (and any other Oracle Applications products). This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

You can access this user's guide online by choosing "Getting Started with Oracle Applications" from any Oracle Applications help file.

User Guides Related to This Product

Accounting Setup User's Guide

The OPM Accounting Setup application is where users set up global accounting attributes about the way financial data will be collected by OPM. These attributes include such things as account keys, financial calendars, and account segments. Since OPM is closely integrated with Oracle General Ledger (GL), much of the

attributes are defined in the Oracle GL instead of OPM, and therefore, the windows are display only within OPM. The *Oracle Process Manufacturing Accounting Setup User's Guide* describes how to setup and use this application.

Cost Management User's Guide

The OPM Cost Management application is used by cost accountants to capture and review the manufacturing costs incurred in their process manufacturing businesses. The *Oracle Process Manufacturing Cost Management User's Guide* describes how to setup and use this application.

Manufacturing Accounting Controller User's Guide

The Manufacturing Accounting Controller application is where users define the impact of manufacturing events on financials. For example, event RCPT (Inventory Receipts) results in a debit to inventory, a credit to accrued accounts payable, a debit or a credit to purchase price variance, etc. These impacts are predefined in the Manufacturing Accounting Controller application so users may begin using OPM to collect financial data out-of-the-box, however, they may also be adjusted per your business needs. The *Oracle Process Manufacturing Manufacturing Accounting Controller User's Guide* describes how to setup and use this application.

Oracle Financials Integration User's Guide

Since OPM is closely integrated with Oracle General Ledger, financial data that is collected about the manufacturing processes must be transferred to the Oracle Financials applications. The OPM Oracle Financials Integration application is where users define how that data is transferred. For example, users define whether data is transferred real time or batched and transferred at intervals. The *Oracle Process Manufacturing Oracle Financials Integration User's Guide* describes how to setup and use this application.

Inventory Management User's Guide

The OPM Inventory Management application is where data about the items purchased for, consumed during, and created as a result of the manufacturing process are tracked. The *Oracle Process Manufacturing Inventory Management User's Guide* includes information to help you effectively work with the Oracle Process Manufacturing Inventory application.

Physical Inventory User's Guide

Performing physical inventory count is the most accurate way to get an accounting of all material quantities purchased, manufactured, and sold, and update your onhand quantities accordingly. The OPM Physical Inventory application automates and enables the physical inventory process. The *Oracle Process Manufacturing Physical Inventory User's Guide* describes how to setup and use this application.

Order Fulfillment User's Guide

The OPM Order Fulfillment application automates sales order entry to reduce order cycle time. Order Fulfillment enables order entry personnel to inform customers of scheduled delivery dates and pricing. The *Oracle Process Manufacturing Order Fulfillment User's Guide* describes how to setup and use this application.

Purchase Management User's Guide

OPM Purchase Management and Oracle Purchasing combine to provide an integrated solution for Process Manufacturing. Purchase orders are entered in Oracle Purchasing and received in OPM. Then, the receipts entered in OPM are sent to Oracle Purchasing. The *Oracle Process Manufacturing Purchase Management User's Guide* describes how to setup and use this integrated solution.

Using Oracle Order Management with Process Inventory Guide

Oracle Process Manufacturing and Oracle Order Management combine to provide an integrated solution for process manufacturers. The manufacturing process is tracked and handled within Oracle Process Manufacturing, while sales orders are taken and tracked in Oracle Order Management. Process attributes, such as dual UOM and lot control, are enabled depending on the inventory organization for the item on the sales order. Order Management accepts orders entered through Oracle Customer Relationship Management (CRM). Within CRM, orders can originate from TeleSales, Sales Online, and iStore, and are booked in Order Management, making the CRM suite of products available to Process customers, through Order Management. The *Oracle Order Management User's Guide* and *Using Oracle Order Management with Process Inventory Guide* describes how to setup and use this integrated solution.

Process Execution User's Guide

The OPM Process Execution application lets you track firm planned orders and production batches from incoming materials through finished goods. Seamlessly integrated to the Product Development application, Process Execution lets you convert firm planned orders to single or multiple production batches, allocate

ingredients, record actual ingredient usage, and then complete and close production batches. Production inquiries and preformatted reports help you optimize inventory costs while maintaining a high level of customer satisfaction with on-time delivery of high quality products. The *OPM Process Execution User's Guide* presents overviews of the tasks and responsibilities for the Production Supervisor and the Production Operator. It provides prerequisite setup in other applications, and details the windows, features, and functionality of the OPM Process Execution application.

Using Oracle Advanced Planning and Scheduling with Oracle Process Manufacturing

Oracle Process Manufacturing and Oracle Advanced Planning and Scheduling (APS) combine to provide a solution for process manufacturers that can help increase planning efficiency. This solution provides for constraint-based planning, performance management, materials management by exception, mixed mode manufacturing that enables you to choose the best method to produce each of your products, and combine all of these methods within the same plant/company. The *Using Oracle Advanced Planning and Scheduling with Oracle Process Manufacturing User's Guide* describes how to setup and use this application.

MPS/MRP and Forecasting User's Guide

The Oracle Process Manufacturing Material Requirements Planning (MRP) application provides long-term "views" of material demands and projected supply actions to satisfy those demands. The Master Production Scheduling (MPS) application lets you shorten that view to a much narrower and immediate time horizon, and see the immediate effects of demand and supply actions. The *Oracle Process Manufacturing MPS/MRP and Forecasting User's Guide* describes how to setup and use this application.

Capacity Planning User's Guide

The OPM Capacity Planning User's Guide describes the setup required to use OPM with the Oracle Applications Advanced Supply Chain Planning solutions. In addition, Resource setup, used by the OPM Production Execution and New Product Development applications, is also described.

Using Oracle Process Manufacturing with Oracle Manufacturing Scheduling

Oracle Process Manufacturing integrates with Oracle Manufacturing Scheduling to manage and utilize resources and materials. Through the Process Manufacturing application, you set up manufacturing, inventory, procurement and sales order

data. Through the Manufacturing Scheduling application, you can optimize the schedule based on resource and component constraints and user predefined priorities. Using different optimization objectives, you can tailor Manufacturing Scheduling to meet your needs.

Using Oracle Manufacturing Scheduling helps you improve productivity and efficiency on your shop floor. By optimally scheduling shop floor jobs, and being able to quickly react to unplanned constraints, you can lower manufacturing costs, increase resource utilization and efficiency, and increase customer satisfaction through improved on-time delivery. The *Using Oracle Process Manufacturing with Oracle Manufacturing Scheduling User's Guide* describes how to setup and use this integrated solution.

Product Development User's Guide

The Oracle Process Manufacturing Product Development application provides features to manage formula and laboratory work within the process manufacturing operation. It lets you manage multiple laboratory organizations and support varying product lines throughout the organization. You can characterize and simulate the technical properties of ingredients and their effects on formulas. You can optimize formulations before beginning expensive laboratory test batches. Product Development coordinates each development function and enables a rapid, enterprise-wide implementation of new products in your plants. The *Oracle Process Manufacturing Product Development User's Guide* describes how to setup and use this application.

Quality Management User's Guide

The Oracle Process Manufacturing Quality Management application provides features to test material sampled from inventory, production, or receipts from external suppliers. The application lets you enter specifications and control their use throughout the enterprise. Customized workflows and electronic record keeping automate plans for sampling, testing, and result processing. You can compare specifications to assist in regrading items, and match customer specifications. Aggregate test results and print statistical assessments on quality certificates. Several preformatted reports and inquiries help manage quality testing and reporting. The *Oracle Process Manufacturing Quality Management User's Guide* describes how to set up and use this application.

Implementation Guide

The *Oracle Process Manufacturing Implementation Guide* offers information on setup. That is, those tasks you must complete following the initial installation of the Oracle

Process Manufacturing software. Any tasks that must be completed in order to use the system out-of-the-box are included in this manual.

System Administration User's Guide

Much of the System Administration duties are performed at the Oracle Applications level, and are therefore described in the *Oracle Applications System Administrator's Guide*. The *Oracle Process Manufacturing System Administration User's Guide* provides information on the few tasks that are specific to OPM. It offers information on performing OPM file purge and archive, and maintaining such things as responsibilities, units of measure, and organizations.

API User's Guides

Public Application Programming Interfaces (APIs) are available for use with different areas of the Oracle Process Manufacturing application. APIs make it possible to pass information into and out of the application, bypassing the user interface. Use of these APIs is documented in individual manuals such as the *Oracle Process Manufacturing Inventory API User's Guide*, *Oracle Process Manufacturing Process Execution API User's Guide*, *Oracle Process Manufacturing Product Development Formula API User's Guide*, *Oracle Process Manufacturing Quality Management API User's Guide*, and the *Oracle Process Manufacturing Cost Management API User's Guide*. Additional API User's Guides are periodically added as additional public APIs are made available.

Installation and System Administration

Oracle Applications Concepts

This guide provides an introduction to the concepts, features, technology stack, architecture, and terminology for Oracle Applications Release 11*i*. It provides a useful first book to read before an installation of Oracle Applications. This guide also introduces the concepts behind Applications-wide features such as Business Intelligence (BIS), languages and character sets, and Self-Service Web Applications.

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11*i*, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications, the Oracle8 technology stack, and the Oracle8*i* Server technology stack by

automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user's guides and implementation guides.

Upgrading Oracle Applications

Refer to this guide if you are upgrading your Oracle Applications Release 10.7 or Release 11.0 products to Release 11*i*. This guide describes the upgrade process and lists database and product-specific upgrade tasks. You must be either at Release 10.7 (NCA, SmartClient, or character mode) or Release 11.0, to upgrade to Release 11*i*. You cannot upgrade to Release 11*i* directly from releases prior to 10.7.

Maintaining Oracle Applications

Use this guide to help you run the various AD utilities, such as AutoUpgrade, AutoPatch, AD Administration, AD Controller, AD Relink, License Manager, and others. It contains how-to steps, screenshots, and other information that you need to run the AD utilities. This guide also provides information on maintaining the Oracle applications file system and database.

Oracle Applications System Administrator's Guide

This guide provides planning and reference information for the Oracle Applications System Administrator. It contains information on how to define security, customize menus and online help, and manage concurrent processing.

Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. It also provides information to help you build your custom Oracle Forms Developer 6*i* forms so that they integrate with Oracle Applications.

Oracle Applications User Interface Standards for Forms-Based Products

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

Other Implementation Documentation

Oracle Applications Product Update Notes

Use this guide as a reference for upgrading an installation of Oracle Applications. It provides a history of the changes to individual Oracle Applications products between Release 11.0 and Release 11*i*. It includes new features, enhancements, and changes made to database objects, profile options, and seed data for this interval.

Multiple Reporting Currencies in Oracle Applications

If you use the Multiple Reporting Currencies feature to record transactions in more than one currency, use this manual before implementing Oracle Process Manufacturing System Administration. This manual details additional steps and setup considerations for implementing Oracle Process Manufacturing System Administration with this feature.

Multiple Organizations in Oracle Applications

This guide describes how to set up and use Oracle Process Manufacturing System Administration with Oracle Applications' Multiple Organization support feature, so you can define and support different organization structures when running a single installation of Oracle Process Manufacturing System Administration.

Oracle Workflow Guide

This guide explains how to define new workflow business processes as well as customize existing Oracle Applications-embedded workflow processes. You also use this guide to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes.

Oracle Applications Flexfields Guide

This guide provides flexfields planning, setup and reference information for the Oracle Process Manufacturing System Administration implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications

product data. This manual also provides information on creating custom reports on flexfields data.

Oracle eTechnical Reference Manuals

Each eTechnical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps you convert data from your existing applications, integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on Oracle*MetaLink*.

Oracle Manufacturing APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes API's and open interfaces found in Oracle Manufacturing.

Oracle Order Management Suite APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes API's and open interfaces found in Oracle Order Management Suite.

Oracle Applications Message Reference Manual

This manual describes all Oracle Applications messages. This manual is available in HTML format on the documentation CD-ROM for Release 11i.

Training and Support

Training

Oracle offers a complete set of training courses to help you and your staff master Oracle E-Records APIs and reach full productivity quickly. These courses are organized into functional learning paths, so you take only those courses appropriate to your job or area of responsibility.

You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many Education Centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your

needs. For example, you may want to use your organization structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle E-Records APIs working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle8i server, and your hardware and software environment.

Do Not Use Database Tools to Modify Oracle Applications Data

*Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications 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 Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using Oracle Applications can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, 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 Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications 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.

About Oracle

Oracle Corporation develops and markets an integrated line of software products for database management, applications development, decision support, and office

automation, as well as Oracle Applications, an integrated suite of more than 160 software modules for financial management, supply chain management, manufacturing, project systems, human resources and customer relationship management.

Oracle products are available for mainframes, minicomputers, personal computers, network computers and personal digital assistants, allowing organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle is the world's leading supplier of software for information management, and the world's second largest software company. Oracle offers its database, tools, and applications products, along with related consulting, education, and support services, in over 145 countries around the world.

Your Feedback

Thank you for using Oracle E-Records APIs and this user's guide.

Oracle values your comments and feedback. At the end of this guide is a Reader's Comment Form you can use to explain what you like or dislike about Oracle E-Records APIs or this user's guide. Mail your comments to the following address or call us directly at (650) 506-7000.

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Or, send electronic mail to **appsdoc_us@oracle.com**.

API Introduction

This document describes the Application Program Interfaces (APIs) that support external interfaces to the Oracle E-Records application. The topics discussed in this chapter are:

- » Introducing the Oracle E-Records APIs
- » Basic Business Needs
- » Major Features
- » Oracle E-Records API Bill of Materials

Introducing the Oracle E-Records APIs

Oracle E-Records APIs let you import information from another system into the Oracle E-Records tables. When you import this information you can include all pertinent information using a tool that does not have cryptic IDs and system specific information. The interface ensures that your imported data contain the same detail as those you enter manually on the Oracle E-Records windows.

This document describes the usage of the business objects that are stored as PL/SQL packages within the database schema, such as:

- Stored procedures used within these packages
- Parameters that these procedures accept and the values that return to the calling program
- Multilingual support
- Error handling methodology

What Is In This Document

This document describes the basic business needs, major features, architecture, and components for the Oracle E-Records APIs insert, update, and delete features. The application is divided into application-specific objects that let you link functionality into your own programs. The interfaces can make use of the standard functionality and logic implemented in the Oracle E-Records application.

Oracle E-Records APIs are written in PL/SQL that can be called by your own programs. To make use of these APIs, code your wrapper function that passes the appropriate parameters to the APIs. Your program is responsible for connecting to a database before calling an API function, and disconnecting from the database upon return. You can choose to write log files before calling and after returning from a function. If there is a problem during execution of a call, then the APIs return one of the following status codes:

- S for success
- E for error
- U for unknown or unexpected status
- N for item requiring a location
- V for inventory shortage exists
- I for incomplete manual transactions exist
- G for over allocation exists

Oracle E-Records API Support Policy

Oracle E-Records APIs are supported by Oracle. This means:

- Oracle provides objects and libraries needed to use the APIs and the documentation for their use.
- Oracle ensures that the APIs function as designed.
- Oracle does not support customer generated programs that use the APIs.

Technical Requirements

Oracle E-Records APIs make use of the following standard Oracle Applications packages:

- FND_API - the standard Oracle Applications API version checking function. This is used by the stored procedure to check for a valid API version number and also contains constant variables such as TRUE and FALSE.
- FND_MESSAGE - the standard Oracle Applications messaging function. This is used by the stored procedure to report status and error handling.
- FND_PUB_MSG - the standard Oracle Applications message retrieval function used to search the procedure messages.

These packages are installed as part of the current release. Refer to the *Oracle Applications Coding Standards* manual for additional details.

Technical Overview

Oracle E-Records APIs are designed to operate in an 11i environment only. They offer the following basic API functionality:

- Creating, updating, and deleting information.
- Proper encapsulation.
- Synchronous processing following the business hierarchy.
- Detailed and translatable error messages.

Basic Business Needs

These APIs let you feed information from shop floor equipment or Manufacturing Execution Systems (MES) directly into Oracle E-Records.

In addition, these APIs serve as a central place to insert, update, or delete batch records in OPM from any source.

Following are some of the important characteristics for these APIs:

Code Reuse

You can maximize code reuse from all application development tools, including PL/SQL, Oracle Forms, and Oracle Reports.

Ease of Integration

You can integrate APIs into other applications and enabling technology, such as Oracle Workflow Server, Oracle Internet Commerce & Oracle WebSystem, and Oracle EDI Gateway.

Insulation from Changes

You can encapsulate the structure of schema to prevent changing schema structures from affecting other applications.

Consistent Behavior

You can hide Object logic specific to an application from other applications, and to ensure that this logic is correctly invoked by other applications and customers.

Robust Validation

You can validate all incoming information into Oracle Applications.

Input Data Sources

Flat File

Input data to the user wrapper function comes from a flat file source. This is processed by the wrapper and header information, and passed as parameters to the stored procedure in a synchronous mode. However, along with the standard parameters, the header information is passed as a PL/SQL table. In this mode, the calling function monitors the success or failure (return code) from the called procedure. It also provides an option to COMMIT work done by the procedure.

Batch File

Input data to the user wrapper function comes from a batch file. This is processed by the wrapper and header information, and passed as parameters to the stored

procedure in an asynchronous mode. In this mode, the calling function does not monitor the success or failure of each individual record. The Oracle Message FND_PUB_MSG functionality is used to audit the calls.

Online User Interface (UI)

Input data to the user wrapper function comes from a window or another user interface. This is processed by the UI and the details passed as parameters to the stored procedure in a synchronous mode. In this mode, the UI calling function monitors the success or failure (return code) from the called procedure. It also provides an option to COMMIT work done by the procedure.

Wrapper Function

Windows are generally used as wrapper functions.

The wrapper function is responsible for collating the details required as input parameters to the stored procedure and forwarding these in the call and monitoring the return code.

The stored procedure returns an return code:

- S for success
- E for error
- U for unknown or unexpected status
- N for item requiring a location
- V for inventory shortage exists
- I for incomplete manual transactions exist
- G for over allocation exists

Based on the return, the wrapper function searches the Oracle Messages File for the stored procedure to determine a COMMIT of the transaction or not.

Stored Procedure

The stored procedure is called with the appropriate parameters forwarded in a PL/SQL table format. The procedure validates each record from this table and then processes the appropriate functional logic as required. The procedure writes appropriate messages to the Oracle Messages table. These are informational as determined by the logic. These can be interrogated by the calling wrapper function through the GET MESSAGES functionality.

The stored procedure calls other validation procedures in the course of its execution; a modular approach is adopted. Functions called by these procedures do not use IN/OUT parameters as these have been removed from the Oracle 8 coding standards.

On successful completion of the procedure, a success unit is in place that can be optionally COMMITTED. The decision as to whether a COMMIT is issued on successful completion is under the control of the calling code and deliberately outside the scope of the API procedures.

Major Features

In order to support the requirements mentioned in the “Basic Business Needs” topic, the new APIs support the following functionality:

Open Document

The Open Document API opens a document for signature and stores the information.

Close Document

The Close Document API closes an opened document after the relevant information is posted.

Update Document

The Update Document API updates a document for signature and stores the information.

Change Document Status

The Change Document Status API changes the document status to the status passed as a parameter.

Post Document Parameters

The Post Document Parameters API associates any name-value pair attributes to the document.

Cancel Document

The Cancel Document API updates the document status to CANCEL.

Get Document Details

The Get Document Details API retrieves information about the document and the relevant signature.

Request Signature

The Request Signature API requests a signature for a document.

Capture Signature

The Capture Signature API captures the signature for a single event and generates the corresponding document ID and signature ID.

Post Signature

The Post Signature API posts signature details about a document.

Post Signature Parameters

The Post Signature Parameters API associates any name-value pair attributes to the specified signature.

Validate E-record

The Validate E-record API determines if the e-record with a given ID exists in the evidence store.

Validate Payload

The Validate Payload API determines if the payload of parameters being passed to raise an ERES event is valid or not.

Validate Payload Forms

The Validate Payload Forms API determines if the payload of parameters being passed to raise an ERES event in forms is valid or not.

Get Event Details

The Get Event Details API retrieves the event name and event key from the evidence store for an e-record ID.

Raise ERES Event

The Raise ERES Event API raises both single events and events that require inter event processing.

Raise Inter Event

The Raise Inter Event API raises a set of ERES events in Inter Event mode.

Get E-record ID

The Get E-record ID API retrieves the e-record ID for a combination of event name and event key from a table of e-record events.

Create Relationship

The Create Relationship API establishes the parent-child relationship between two e-record events.

Validate Relationship

The Validate Relationship API validates a parent-child relationship.

Send Acknowledgement

The Send Acknowledgement API creates an acknowledgement for an e-record in the evidence store.

Get PSIG Status

The Get PSIG Status API retrieves the signature status of an e-record event.

Is E-signature Required

The Is E-signature Required API determines if a signature is required for an e-record event.

Is E-record Required

The Is E-record Required API determines if an e-record is required for the specified ERES event.

Get Query ID on Events

The Get Query ID on Events API retrieves a query ID for a list of events passed to the API as p_eventQuery_recTbl.

Get AME Rule Variable Values

The Get AME Rule Variable Values API retrieves the AME input variable values for a specified transaction and rule.

Display Date

The Display Date API converts an input date value of type DATE to a corresponding string display value.

Display Date Only

The Display Date Only API converts an input date value of type DATE to a corresponding string display value.

Display Time Only

The Display Time Only API converts an input date value of type DATE to a corresponding string display value.

Is Audit Value Old

The Is Audit Value Old API checks if the audited table column was modified during a transaction.

Get Notification Routing Information

The Get Notification Routing Information API retrieves the overriding recipient for a routed notification in a workflow.

Get Descriptive Flexfield One Prompt

The Get Descriptive Flexfield One Prompt API retrieves a single descriptive flexfield prompt for an application ID, Descriptive Flexfield, and column.

Get Descriptive Flexfield All Prompts

The Get Descriptive Flexfield All Prompts API retrieves the prompts of all items in the descriptive flexfield.

Get Lookup Meaning

The Get Lookup Meaning API retrieves the lookup code meaning for a specified lookup code and type.

Get Query ID on Parameters

The Get Query ID on Parameters API retrieves the query ID for events based on an array of event parameters, (NAME, KEY).

Upload File

The Upload File API provides a generic file upload management system for uploading files into the Oracle E-Business Suite.

Oracle E-Records API Bill of Materials

The following are packages and files that are delivered with the Oracle E-Records APIs. These must be on your system for your interface to compile and link properly.

Package Name	File Names
EDR_EVIDENCESTORE_PUB	EDRPEVRS.pls, EDRPEVRB.pls
EDR_ERES_EVENT_PUB	EDRPEVTS.pls, EDRPEVTB.pls
EDR_EVENT_RELATIONSHIP_PUB	EDRPRELS.pls, EDRPRELB.pls
EDR_TRANS_ACKN_PUB	EDRPACKS.pls, EDRPACKB.pls
EDR_STANDARD_PUB	EDRPSTDS.pls, EDRSTNDB.pls
EDR_FILES_PUB	EDRPFILS.pls, EDRFILB.pls
EDR_FILE_UTIL_PUB	EDRFUTLS.pls, EDRFUTLB.pls

Oracle E-Records API Usage

The Oracle E-Records APIs are written in PL/SQL. To use these APIs, code your interface or wrapper. Your program is responsible for connecting to a database before calling an API function. You can write log files before calling and after returning from an API function. Each function returns an error code in the parameter `x_return_status` that indicates whether the API was successful or failed. The values are:

- S for success
- E for error
- U for unknown or unexpected status
- N for item requiring a location
- V for inventory shortage exists
- I for incomplete manual transactions exist
- G for over allocation exists

The topics discussed in this chapter are:

- Calling the API Interface Code

Calling the API Interface Code

The following are part of a sample wrapper and are used to test the API code. Wrappers are written as PL/SQL packages. Wrappers can be written for each API and call the APIs directly. The source of data for the wrapper comes from an ASCII flat file in this example. You can write a similar type of wrapper to call the API code.

These wrappers have the following parameters:

Standard Input Parameters

p_api_version	IN	NUMBER
p_validation_level	IN	NUMBER
p_init_msg_list	IN	BOOLEAN
p_commit	IN	BOOLEAN

Standard Output Parameters

x_message_count	OUT	NUMBER
x_message_list	OUT	VARCHAR2
x_return_status	OUT	VARCHAR2

API Call Hints

For performance improvement, NOCOPY hints are available in the OUT parameters of the APIs. When an API has the same type of parameter defined as IN and OUT, pass in different variables. In addition, check the return status of the API (generally returned through x_return_status parameter) before looking at other OUT variables returned by the API. If the return status is not Success, then do not use any of the OUT parameters passed back from the API.

Technical Overview

The public Oracle E-Records APIs perform all validations necessary on input data supplied in order to prevent the flow of invalid data into OPM. If any validation errors occur, then that particular row is skipped and the process continues to the next record. If the insert fails, then none of the detail records for the item in process are inserted. After finishing validations on input data, the public API performs the required function by calling the necessary routines.

The topics discussed in this chapter are:

- Structure for Oracle E-Records Public APIs
- Standard Parameters

Structure for Oracle E-Records Public APIs

According to API standards, the following are the files, packages, and procedures for the public APIs.

Evidence Store APIs

These are the Evidence Store API standards.

Object Type	Name
Package Specification File	EDRPEVRS.pls,
Package Body File	EDRPEVRB.pls
Package	EDR_EVIDENCESTORE_PUB
Procedure - Open Document	Open_Document
Procedure - Close Document	Close_Document
Procedure - Update Document	Update_Document
Procedure - Change Document Status	Change_DocumentStatus
Procedure - Post Document Parameters	Post_DocumentParameter
Procedure - Cancel Document	Cancel_Document
Procedure - Get Document Details	Get_DocumentDetails
Procedure - Request Signature	Request_Signature
Procedure - Capture Signature	Capture_Signature
Procedure - Post Signature	Post_Signature
Procedure - Post Signature Parameters	Post_SignatureParameters

E-Record Event APIs

These are the E-Record Event API standards.

Object Type	Name
Package Specification File	EDRPEVTS.pls
Package Body File	EDRPEVTB.pls
Package	EDR_ERES_EVENT_PUB
Procedure - Validate E-record	validate_erecord

Object Type	Name
Procedure - Validate Payload	validate_payload
Procedure - Validate Payload Forms	validate_payload_forms
Procedure - Get Event Details	get_event_details
Procedure - Raise ERES Event	raise_eres_event
Procedure - Raise Inter Event	raise_inter_event
Procedure - Get E-record ID	get_erecord_id

Relationship APIs

These are the Relationship API standards.

Object Type	Name
Package Specification File	EDRPRELS.pls,
Package Body File	EDRPRELB.pls
Package	EDR_EVENT_RELATIONSHIP_PUB
Procedure - Create Relationship	create_relationship
Procedure - Validate Relationship	validate_relationship

Transaction Acknowledgment API

These are the Transaction Acknowledgement API standards.

Object Type	Name
Package Specification File	EDRPACKS.pls,
Package Body File	EDRPACKB.pls
Package	EDR_TRANS_ACKN_PUB
Procedure - Send Acknowledgement	send_ackn

General Utility APIs

These are the General Utility API standards.

Object Type	Name
Package Specification File	EDRPSTDS.pls,
Package Body File	EDRSTNDB.pls
Package	EDR_STANDARD_PUB
Procedure - Get PSIG Status	Get_PsigStatus
Procedure - Is E-signature Required	Is_esig_Required
Procedure - Is E-record Required	Is_eRec_Required
Procedure - Get Query ID on Events	Get_QueryId_OnEvents
Procedure - Get AME Rule Variable Values	Get_AmeRule_VarValues
Procedure - Display Date	display_date
Procedure - Display Date Only	display_date_only
Procedure - Display Time Only	display_time_only
Procedure - Is Audit Value Old	Is_AuditValue_Old
Procedure - Get Notification Routing Information	Get_Notif_Routing_Info
Procedure - Get Descriptive Flexfield One Prompt	Get_DescFlex_OnePrompt
Procedure - Get Descriptive Flexfield All Prompts	Get_DescFlex_AllPrompts
Procedure - Get Lookup Meaning	Get_Lookup_Meaning
Procedure - Get Query ID on Parameters	Get_QueryId_OnParams

Document Management APIs

These are the Document Management API standards.

Object Type	Name
Package Specification File	EDRPFILS.pls, EDFUTLS.pls
Package Body File	EDRFILB.pls, EDFUTLB.pls
Package	EDR_FILES_PUB, EDR_FILES_UTIL_PUB
Procedure - Upload File	upload_file

Standard Parameters

API standard parameters are a collection of parameters that are common to most APIs. The following paragraphs explain the standard parameters used in APIs and their interpretation.

Some of the standard parameters apply to all APIs regardless of the nature of the business function they perform. For example, `p_api_version` and `x_return_status` are included in all APIs.

Some parameters are applicable for certain types of APIs and not applicable for other types. For example, `p_commit` is applicable for APIs that change the database state, and not applicable for read APIs.

Standard parameters are included in all APIs whenever applicable.

Standard IN parameters:

- `p_api_version`
- `p_init_msg_list`
- `p_commit`
- `p_validation_level`

Standard OUT parameters:

- `x_return_status`
- `x_msg_count`
- `x_msg_data`

Parameter	Type	IN/OUT	Required	Validation
<code>p_api_version</code>	number	IN	Y	Validates version compatibility. The version sent by the calling function is compared to the internal version of the API and an unexpected error (U) is generated if these do not match.

Parameter	Type	IN/OUT	Required	Validation
p_init_msg_list	boolean	IN	N	Used to specify whether the message list is initialized on entry to the API. It is an optional parameter and if not supplied, defaults to FND_API.G_FALSE which means that the API does not initialize the message list. If multiple APIs are called in the same session, then pass this as TRUE; otherwise messages accumulate on the stack.
p_commit	boolean	IN	N	Used to specify whether the API commits its work before returning to the calling function. If not supplied, then it defaults to FALSE. Ensure that the save_batch procedure is called before performing any commits manually (if p_commit parameter is passed as FALSE).
x_return_status	varchar2	OUT	N	Specifies whether the API was successful or failed. Valid values are S for successful, E for failed due to expected error, or U for failed due to unexpected error.
x_msg_count	number	OUT	N	Specifies the number of messages added to message list.
x_msg_data	varchar2	OUT	N	Returns the messages in an encoded format. These messages are processed by the standard message functions as defined in the Business Object API Coding Standards Document.

Value-ID Conversion

IDs are used to represent primary and foreign entity keys and for internal processing of attributes. They are not meaningful and are hidden. Besides IDs, attributes have values that represent them. These values are meaningful and used for display purposes. In general, APIs operate only on IDs.

For example, an item is represented by an ID, or the number column item_id. This ID is its primary key and is used for all internal processing of the item. Besides this ID, an item is represented by the value varchar2 column item_no. This value is displayed when you choose an item. Therefore, an item can be identified by either its ID or value, in this case item_no.

The following set of rules are for the conversion process:

- ID or value, or both can be passed to an API. But, when both values are passed, ID based parameters take precedence over value based parameters. For example, if both parameters are passed, the value based parameter is ignored and the ID based parameter is used.
- When both the value and ID of an attribute are passed to an API, a message informs the API caller that some of the input has been ignored.
- This message is not an error message. The API continues with its regular processing.
- Each value has to resolve into one ID. Failure to resolve a value into an ID is an error and is associated with an error message. The API aborts processing and returns with a return status of error.

Business Objects

This topic defines the table structure and column definitions for the E-Records APIs. The following topics are discussed:

- Evidence Store APIs
- E-Record Event APIs
- Event Relationship APIs
- ERES Transaction Acknowledgement APIs
- General Purpose APIs
- APIs for Forms-Based Applications

Evidence Store APIs

Oracle E-Records provides standard PL/SQL APIs, that can be used to post data in the evidence store. The APIs are designed as generic and can be called from any location.

Open Document

The Open Document API opens a document for signature and stores the information. The document opened using this APIs remains active until the document ID is closed using another API or the document is cancelled. The `DOCUMENT_ID` returned by this API is the required key for calling other relevant APIs.

Following is the definition of Open Document:

Field/Column	Type	IN/OUT	Default	Description
p_psig_xml	clob	IN	null	The XML file containing the transaction snapshot.
p_psig_document	clob	IN	null	The actual signature document.
p_psig_documentFormat	varchar2	IN	null	The document format, which takes the format of either text or html.
p_psig_requester	varchar2	IN	null	E-signature requester name.
p_psig_source	varchar2	IN	null	Source of e-signature request.
p_event_name	varchar2	IN	null	Event name with which the API is called.
p_event_key	varchar2	IN	null	Event key associated with the event name.
p_wf_notif_id	number	IN	null	Workflow notification ID.
x_document_id	number	OUT	N/A	Contains the ID of the document open.

Close Document

The Close Document API closes an opened document after the relevant information is posted. The closed document cannot be modified further.

Following is the definition of Close Document:

Field/Column	Type	IN/OUT	Default	Description
p_document_id	number	IN	N/A	The ID of the document being closed.

Update Document

The Update Document API updates a document for signature and stores the information.

Following is the definition of Update Document:

Field/Column	Type	IN/OUT	Default	Description
p_document_id	number	IN	N/A	The ID of the document being updated.
p_psig_xml	clob	IN	null	The XML file containing the transaction snapshot.
p_psig_document	clob	IN	null	The e-record document.
p_psig_documentFormat	varchar2	IN	null	The document format which takes the format of either text or html.
p_psig_requester	varchar2	IN	N/A	E-signature requester name.
p_psig_source	varchar2	IN	null	The source of the e-signature request.
p_event_name	varchar2	IN	null	The event name from which the update is called.
p_event_key	varchar2	IN	null	Event key associated with the event name.
p_wf_notif_id	number	IN	null	Workflow notification ID.

Change Document Status

The Change Document Status API changes the document status to the status passed as a parameter. There is no validation performed on the status value.

Following is the definition of Change Document Status:

Field/Column	Type	IN/OUT	Default	Description
p_document_id	number	IN	N/A	The ID of the document whose status is being changed.
p_document_status	varchar2	IN	N/A	The document status value that is set by the API.

Post Document Parameters

The Post Document Parameters API associates any name-value pair attributes to the document.

Following is the definition of Post Document Parameters:

Field/Column	Type	IN/OUT	Default	Description
p_document_id	number	IN	N/A	Document ID.
p_doc_parameters_tbl	EDR_EvidenceStore_PUB.Params_tbl_type	IN	N/A	Table of document parameters being associated.

Cancel Document

This API is used to update the document status to CANCEL. All rows from relevant tables are deleted for a given document ID. This API cannot be called after the document is closed.

Following is the definition of Cancel Document:

Field/Column	Type	IN/OUT	Default	Description
p_document_id	number	IN	N/A	The ID of the document being cancelled.

Get Document Details

The Get Document Details API fetches information about the document and the relevant signature.

Following is the definition of Get Document Details:

Field/Column	Type	IN/OUT	Default	Description
p_document_id	number	IN	N/A	Document ID.
x_document_rec	edr_psig_documents%ROWTYPE	OUT	N/A	Contains the record of document details.
x_doc_parameters_tbl	Edr_EvidenceStore_PUB.params_tbl_type	OUT	N/A	Contains the table of document parameters.

Field/Column	Type	IN/OUT	Default	Description
x_signatures_tbl	edr_evidencestore_pub.signature_tbl_type	OUT	N/A	Contains the table of signature details.

Request Signature

The Request Signature API requests a signature for a document. It creates a new signature row in the signature table for the document and user.

Following is the definition of Request Signature:

Field/Column	Type	IN/OUT	Default	Description
p_document_id	number	IN	N/A	Document id.
p_user_name	varchar2	IN	N/A	User name requesting the signature.
p_original_recipient	varchar2	IN	null	The original recipient of the signature if transferred.
p_overriding_comment	varchar2	IN	null	The user overriding comment if it is a transferred notification.
x_signature_id	number	OUT		Contains the requested signature ID.

Capture Signature

The Capture Signature API captures the signature for a single event and generates the corresponding document ID and signature ID.

Following is the definition of Capture Signature:

Field/Column	Type	IN/OUT	Default	Description
p_psig_xml	clob	IN	null	The XML file containing the transaction snapshot.
p_psig_document	clob	IN	null	The signature document.

Field/Column	Type	IN/OUT	Default	Description
p_psig_docFormat	varchar2	IN	N/A	The document format which takes the format of either text or html.
p_psig_requester	varchar2	IN	N/A	E-signature requester name.
p_psig_source	varchar2	IN	N/A	Source of e-signature request.
p_event_name	varchar2	IN	N/A	Event name from which the API is called.
p_event_key	varchar2	IN	N/A	Event key associated with the event name.
p_wf_notif_id	number	IN	N/A	Workflow notification ID.
p_doc_parameters_tbl	EDR_EvidenceStore_PUB.Params_tbl_type	IN	N/A	Table of document parameters being associated.
p_user_name	varchar2	IN	N/A	The user name calling the API.
p_original_recipient	varchar2	IN	null	The original recipient of the notification.
p_overriding_comment	varchar2	IN	null	User overriding comment if it is a transferred notification.
p_evidenceStore_id	number	IN	N/A	Source evidence store ID.
p_user_response	varchar2	IN	N/A	User response or comment.
p_sig_parameters_tbl	EDR_EvidenceStore_PUB.Params_tbl_type	IN	N/A	Table of signature parameters.
x_signature_id	number	OUT	N/A	Contains the value of the signature ID.
x_document_id	number	OUT	N/A	Contains the value of the document ID.

Post Signature

The Post Signature API posts signature details for a specific document.

Following is the definition of Post Signature:

Field/Column	Type	IN/OUT	Default	Description
p_document_id	number	IN	N/A	Document ID.
p_evidencestore_id	varchar2	IN	N/A	Source Evidence Store ID.
p_user_name	varchar2	IN	N/A	Posting user name.
p_user_response	varchar2	IN	N/A	User response or comment
p_original_recipient	varchar2	IN	null	Original recipient of the notification if it is transferred.
p_overriding_comment	varchar2	IN	null	User overriding comment if it is a transferred notification.
x_signature_id	number	OUT		Contains the signature ID posted.

Post Signature Parameters

The Post Signature Parameters API associates any name-value pair attributes to the specified signature.

Following is the definition of Post Signature Parameters:

Field/Column	Type	IN/OUT	Default	Description
p_signature_id	number	IN	N/A	Contains the signature ID posted.
p_sig_parameters_tbl	EDR_EvidenceStore_PUB.params_tbl_type	IN	N/A	Table of signature parameters.

E-Record Event APIs

Oracle E-Records provides standard PL/SQL APIs, that can be used in e-record events. The APIs are designed as generic and called from any location.

Validate E-record

The Validate E-record API determines if the e-record with a specific ID exists in the evidence store. The return value indicates the result.

Following is the definition of Validate E-record:

Field/Column	Type	IN/OUT	Default	Description
p_erecord_id	number	IN	N/A	The ID of the e-record being searched.

Validate Payload

The Validate Payload API determines if the payload of parameters being passed to raise an ERES event is valid.

Following is the definition of Validate Payload:

Field/Column	Type	IN/OUT	Default	Description
p_event_name	varchar2	IN	null	The event name for which the payload is validated.
p_event_key	varchar2	IN	Null	The event key associated with the event name.
p_payload	fnd_wf_event.param_table	IN	N/A	The name-value pair of parameters being validated.
p_mode	varchar2	IN	null	The mode of validation. Valid values are Strict or Null. This parameter is only used if the payload contains the parameters for inter event processing. Strict is used when the parent e-record ID is passed.

Validate Payload Forms

The Validate Payload Forms API determines if the payload of parameters being passed to raise an ERES event is valid. This API is used from forms.

Following is the definition of Validate Payload Forms:

Field/Column	Type	IN/OUT	Default	Description
p_event_name	varchar2	IN	null	The event name for which the payload is validated.
p_event_key	varchar2	IN	null	The event key associated with the event name.
p_payload	fnd_wf_event.param_table	IN	N/A	The name-value pair of parameters being validated.

Get Event Details

The Get Event Details API retrieves the event name and event key from the evidence store for a specific e-record ID.

Following is the definition of Get Event Details:

Field/Column	Type	IN/OUT	Default	Description
p_erecord_id	number	IN	N/A	The ID of the e-record.
x_event_name	varchar2	OUT	N/A	Contains the event name if e-record ID is valid. Otherwise, the value is null.
x_event_key	varchar2	OUT	N/A	Contains the event name if e-record ID is valid. Otherwise, the value is null.

Raise ERES Event

The Raise ERES Event API raises both single events and events that require inter event processing. For single events, the p_child_erecords parameter must be null and the payload must conform to guidelines set in the *Oracle E-Records Developers Guide*.

If the parent already exists for events that require Inter Event processing, then the payload must contain the details of the parent event to set the parent child

relationship. If the child e-records exist, then the p_child_erecords parameter must contain the e-record IDs of these children.

Following is the definition of Raise ERES Event:

Field/Column	Type	IN/OUT	Default	Description
p_validation_level	number	IN	null	Used to specify the extent to which the API parameters are validated.
p_child_erecords	erecord_Id_Tbl_Type	IN	N/A	This parameter contains the list of e-record IDs of the children to establish parent-child relationship.
x_event	Eres_Event_Tbl_Type	OUT	N/A	Contains the table of e-record event structure with event name, event key, and payload supplied by the caller. At the end, the e-record ID and status are populated.

Raise Inter Event

The Raise Inter Event API raises a set of ERES events in Inter Event mode. It is expected that while using this API at least one parent event is raised along with one or more child events.

Following is the definition of Raise Inter Event:

Field/Column	Type	IN/OUT	Default	Description
p_validation_level	number	IN	null	Used to specify the extent to which the API parameters are being validated.
x_events	Eres_Event_Tbl_Type	OUT	N/A	Contains the table of e-record event structure with event name, event key, and payload supplied by the caller. At the end, the e-record ID and status are populated.
x_overall_status	varchar2	OUT	N/A	The overall status of the e-record processing. Valid values are Error, Complete, Pending, and Indeterminate.

Get E-record ID

The Get E-record ID API retrieves the e-record ID for a combination of event name and event key from a table of e-record events. The updated e-record ID for the combination is returned.

Following is the definition of Get E-record ID:

Field/Column	Type	IN/OUT	Default	Description
p_events	ERES_EVENT_TBL_TYPE	IN	N/A	The e-record event table from which the event name is extracted.
p_event_name	varchar2	IN	N/A	The event name being searched from the table.
p_event_key	varchar2	IN	N/A	The event key associated with the event name.
x_erecord_id	number	OUT	N/A	Returns the most recent e-record ID associated with the specified event name and key. Returns null if the combination is not found in the table.

Event Relationship APIs

Create Relationship

The Create Relationship API creates a parent-child relationship between the specified parent and child events.

Following is the definition of Create Relationship:

Field/Column	Type	IN/OUT	Default	Description
p_commit	varchar2	IN	FND_API.G_FALSE	Used to specify whether the API must commit before returning to the calling function.
p_validation_level	number	IN	FND_API.G_VALID_LEVEL_FULL	Used to specify the extent to which the parameters to the API are being validated.
p_parent_erecord_id	number	IN	N/A	The parent e-record ID.
p_parent_event_name	varchar2	IN	null	The parent event name.
p_parent_event_key	varchar2	IN	null	The parent event key.
p_child_erecord_id	number	IN	N/A	The child e-record ID.
p_child_event_name	varchar2	IN	null	The child event name.
p_child_event_key	varchar2	IN	null	The child event key.
x_relationship_id	number	OUT	N/A	Contains the primary key value of the created row in the table when successful. This is null if there is an error.

Validate Relationship

The Validate Relationship API verifies if a parent-child relationship can be created between the specified parent and child events.

Following is the definition of Validate Relationship:

Field/Column	Type	IN/OUT	Default	Description
p_parent_erecord_id	number	IN	N/A	The parent e-record ID.
p_parent_event_name	varchar2	IN	null	The parent event name.
p_parent_event_key	varchar2	IN	null	The parent event key.
p_child_erecord_id	number	IN	N/A	The child e-record ID.
p_child_event_name	varchar2	IN	null	The child event name.
p_child_event_key	varchar2	IN	null	The child event key.
x_return_status	varchar2	IN	N/A	This is the result status. Valid values are S - Successful Validation, E - Failed Validation.

ERES Transaction Acknowledgement

This API package sends appropriate acknowledgement for a given business transaction in execution. The API is in the EDR_TRANS_ACKN_PUB package.

Send Acknowledgment

The Send Acknowledgment API creates an acknowledgement for an e-record in the evidence store. This acknowledgement indicates whether the business transaction for which the e-record was created was completed successfully.

Following is the definition of Send Acknowledgment:

Field/Column	Type	IN/OUT	Default	Description
p_autonomous_commit	varchar2	IN	FND_API.G_FALSE	Indicates if the API commits changes autonomously. The autonomous commit is done only if the flag has the value Y.
p_event_name	varchar2	IN	N/A	The event name for which the acknowledgment is sent.
p_event_key	VARCHAR2	IN	N/A	The event key associated with the event name.
p_erecord_id	number	IN	N/A	The e-record ID for which the acknowledgment is sent.
p_trans_status	varchar2	IN	N/A	The status of the transaction for which the acknowledgment is being generated. Valid values are Success and Error.
p_ackn_by	varchar2	IN	null	The source of the acknowledgment.
p_ackn_note	varchar2	IN	null	Additional comments about the acknowledgment.

General Purpose APIs

This is a generic standard body package that contains all the general purpose e-record routines.

Get Psig Status

The Get Psig Status API retrieves the signature status for a specified event.

Following is the definition of Get Psig Status:

Field/Column	Type	IN/OUT	Default	Description
p_event_name	varchar2	IN	N/A	The event name for which the signature status is being obtained.
p_event_key	varchar2	IN	N/A	The event key associated with the event name.
x_psig_status	varchar2	OUT	N/A	The signature status for the event. Valid values are Pending, Success, and Error.

Is E-signature Required

The Is E-signature Required API checks if a signature is required for a specified event.

Following is the definition of Is E-signature Required:

Field/Column	Type	IN/OUT	Default	Description
p_event_name	varchar2	IN	N/A	The event name being checked.
p_event_key	varchar2	IN	N/A	The event key associated with the event name.
x_isRequired_eSig	varchar2	OUT	N/A	This is the requirement status. Valid values are True and False.

Is E-record Required

The Is E-record Required API checks if an e-record is required for a specified event.

Following is the definition of Is E-record Required:

Field/Column	Type	IN/OUT	Default	Description
p_event_name	varchar2	IN	N/A	The event name being checked.

Field/Column	Type	IN/OUT	Default	Description
p_event_key	varchar2	IN	N/A	The event key associated with the event name.
x_isRequired_eRec	varchar2	OUT	N/A	This is the requirement status. Valid values are True and False.

Get Query ID On Events

The Get Query ID On Events API retrieves a query ID for a list of events passed on to the API as p_eventQuery_recTbl. This API queries the e-record repository through forms.

Following is the definition of Get Query ID On Events:

Field/Column	Type	IN/OUT	Default	Description
p_eventQuery_recTbl	EventDetail_tbl_type	IN	N/A	The table of event query records.
x_query_id	number	OUT	N/A	The value of the query ID returned.

Get AME Rule Variable Values

The Get AME Variable Values API retrieves the AME input variable values for a specified transaction and rule.

Following is the definition of Get AME Variable Values:

Field/Column	Type	IN/OUT	Default	Description
p_transaction_name	varchar2	IN	N/A	The required transaction name.
p_ameRule_id	number	IN	N/A	The AME rule ID associated with the transaction.
p_ameRule_name	varchar2	IN	N/A	The corresponding AME rule name.
x_inputVar_values_tbl	InputVar_Values_tbl_type	OUT	N/A	The corresponding variable values.

Display Date

The Display Date API converts an input date value of type DATE to a corresponding string display value.

Following is the definition of Display Date:

Field/Column	Type	IN/OUT	Default	Description
p_date_in	DATE	IN	N/A	Input Date.
x_date_out	varchar2	OUT	N/A	The converted date in string format.

Display Date Only

The Display Date Only API converts an input date value of type DATE to a corresponding string display value.

Following is the definition of Display Date Only:

Field/Column	Type	IN/OUT	Default	Description
p_date_in	DATE	IN	N/A	Input Date.
x_date_out	varchar2	OUT	N/A	The converted date in string format.

Display Time Only

The Display Time Only API converts an input date value of type DATE to a corresponding string display value.

Following is the definition of Display Time Only:

Field/Column	Type	IN/OUT	Default	Description
p_date_in	DATE	IN	N/A	Input Date.
x_date_out	varchar2	OUT	N/A	The converted date in string format.

Is Audit Value Old

The Is Audit Value Old API checks if the audited table column was modified during a transaction.

Following is the definition of Is Audit Value Old:

Field/Column	Type	IN/OUT	Default	Description
p_table_name	varchar2	IN	N/A	The audit table name being checked.
p_column_name	varchar2	IN	N/A	The corresponding audit column being checked.
p_primKey_name	varchar2	IN	N/A	The primary key name.
p_primKey_value	varchar2	IN	N/A	The primary key value.
x_isOld_auditValue	varchar2	OUT	N/A	The corresponding status value.

Get Notification Routing Information

The Get Notification Routing Information API retrieves the overriding recipient for a routed notification in a workflow. The routing comments are also returned.

Following is the definition of Get Notification Routing Information:

Field/Column	Type	IN/OUT	Default	Description
p_original_recipient	varchar2	IN	N/A	The original recipient of the notification.
p_message_type	varchar2	IN	N/A	The type of message sent during the notification.
p_message_name	varchar2	IN	N/A	The corresponding message name.
x_final_recipient	varchar2	OUT	N/A	The final recipient of the notification.
x_allrout_comments	varchar2	OUT	N/A	Concatenation of all the routing comments.

Get Descriptive Flexfield One Prompt

The Get Descriptive Flexfield One Prompt API retrieves a single descriptive flexfield prompt for a specific application ID, descriptive flexfield, and column.

Following is the definition of Get Descriptive Flexfield One Prompt:

Field/Column	Type	IN/OUT	Default	Description
p_application_id	number	IN	N/A	The Oracle Applications application ID.
p_descFlex_defName	varchar2	IN	N/A	The flexfield definition name.
p_descFlex_context	varchar2	IN	N/A	The flexfield context.
p_column_name	varchar2	IN	N/A	The column name in the flexfield.
p_prompt_type	varchar2	IN	null	Prompt type.
x_column_prompt	varchar2	OUT	N/A	The returned flexfield column prompt.

Get Descriptive Flexfield All Prompts

The Get Descriptive Flexfield All Prompts API retrieves the prompts of all the items in the descriptive flexfield.

Following is the definition of Get Descriptive Flexfield All Prompts:

Field/Column	Type	IN/OUT	Default	Description
p_application_id	number	IN	N/A	The Oracle Applications application ID.
p_descFlex_defName	varchar2	IN	N/A	The flexfield definition name.
p_descFlex_context	varchar2	IN	N/A	The flexfield context.
p_colnNames_rec	varchar2	IN	N/A	The column name in the flexfield.
p_prompt_type	varchar2	IN	null	Prompt type.
x_colnPrompts_rec	varchar2	OUT	N/A	Contains the values of all 30 flexfield column prompts.

Get Lookup Meaning

The Get Lookup Meaning API retrieves the lookup code meaning using FND_LOOKUPS view for a specific lookup type and code.

Following is the definition of Get Lookup Meaning:

Field/Column	Type	IN/OUT	Default	Description
p_lookup_type	varchar2	IN	N/A	Lookup type.
p_lookup_code	varchar2	IN	N/A	Lookup code.
x_lkup_meaning	varchar2	OUT	N/A	Contains the FND lookup meaning for the specified type and code.

Get Query ID On Parameters

The Get Query ID On Parameters API retrieves a query ID for events based on arrays of event parameters (NAME, KEY).

Following is the definition of Get Query ID On Parameters:

Field/Column	Type	IN/OUT	Default	Description
p_eventNames_tbl	FND_TABLE_OF_varchar2_255	IN	N/A	The array of event names.
p_eventKeys_tbl	FND_TABLE_OF_varchar2_255	IN	N/A	The corresponding array of event keys.
x_query_id	number	OUT	N/A	Contains the query ID obtained for the array of event parameters.

APIs for Forms-Based Applications

The following API is for forms-based applications only and does not use the standard API parameters.

Add Transaction

The Add Transaction API adds a transaction from a form to the internal structures. In most cases, this API is called multiple times to add all the transactions being processed in a single database commit statement.

Following is the definition of Add Transaction:

Field/Column	Type	IN/OUT	Default	Description
p_block	varchar2	IN	N/A	The form block from which this procedure was called.
p_action	varchar2	IN	N/A	The action performed in the transaction. Valid values are Insert, Delete, and Update.
p_Event_Name	varchar2	IN	N/A	The event name associated with the transaction.
p_event_key	varchar2	IN	N/A	The event key associated with the event.
p_payload	EDR_SIGN_ GRP.payload_ table	IN	N/A	The name-value pair of parameters being validated.

Messages and Errors

This topic discusses the use and interpretation of error messages when running the APIs.

The topics discussed in this chapter are:

- Handling Messages
- Interpreting Error Conditions
- Understanding Error Messages

Handling Messages

APIs put result messages into a message list. Programs calling APIs retrieve the messages from the list and process them by either issuing them, loading them in a database table, or writing them to a log file. Messages are stored in an encoded format to enable API callers to find message names by using the standard functions provided by the message dictionary. It also stores these messages in database tables and reports from these tables in different languages. The structure of the message list is not public. Neither API developers nor API callers can access this list except through calling the API message utility routines mentioned below.

The following utility functions are defined in the FND_MSG_PUB package, in the file AFASMSGs.pls:

Initialize

Initializes the API message list

Add

Adds a message to the API message list

Get

Gets a message from the API message list

Count_Msg

Returns the number of messages in the API message list

Delete

Deletes one or more messages from the API message list

Reset

Resets the index used in getting messages

Count_And_Get

Returns the number of messages in the API message list. If this number is one, then it also returns the message data.

To add a message to the API message list, use the regular message dictionary procedures `FND_MESSAGE.SET_NAME` and `FND_MESSAGE.SET_TOKEN` to set the message name and tokens on the message dictionary stack. Then call `FND_MSG_PUB.Add` to fetch the messages from the message dictionary stack and add it to the API message list.

To get a message from the API message list, API callers use the procedure `FND_MSG_PUB.Get`. This procedure operates in five modes:

First

Gets the first message in the API message list

Next

Gets the next message in the API message list

Last

Gets the last message in the API message list

Previous

Gets the previous message in the API message list

Specific

Gets a specific message from the API message list

For overall better performance and reduction in the number of calls a program needs to make in order to execute an API, it is recommended that APIs provide their callers with the following information:

- message count
- message data

The message count holds the number of messages in the API message list. If this number is one, then the message data holds the message in an encoded format.

```
x_msg_count  OUT  NUMBER
x_msg_data   OUT  VARCHAR2
```

Interpreting Error Conditions

The parameter `x_return_status` indicates whether the API was successful or failed. The values are:

- S for success
- E for error
- U for unknown or unexpected status

Understanding Error Messages

These error messages are output to the stored procedure message file and can be monitored through the return `x_msg_count`. In conjunction with the `x_return_status`, this can be used to monitor the success or failure of the procedure call.

Displaying Errors in Languages Other than English

Language translation of error messages is determined by the environment variable `NLS_LANGUAGE`. If the message is not found in the required language, then the message is retrieved in US English.

The following is a complete list of Oracle E-Records API Error Messages. Note that a message that is preceded with Warning is not an API error, just a warning, and a message preceded with Error is an API error.

Any uppercase word preceded by an ampersand (&) is a token, or placeholder, for an actual value that is populated at runtime.

Message Text	Message Name
The e-record ID &ERECORD_ID has been acknowledged. A duplicate acknowledgement cannot be created.	EDR_ACK_DUPLICATE_ACKN
The e-record ID &ERECORD_ID is invalid and an acknowledgement cannot be created.	EDR_ACK_INVALID_ERECORD
The event name &EVENT_NAME or event key &EVENT_KEY are invalid for the e-record ID &ERECORD_ID.	EDR_ACK_INVALID_EVENT
The transaction acknowledgement status &STATUS is invalid for the e-record ID &ERECORD_ID.	EDR_ACK_INVALID_STATUS
An unexpected error has taken place while creating an acknowledgement for the e-record ID &ERECORD_ID.	EDR_ACK_UNEXPECTED_ERROR
Invalid notification routing rule for &NAME	EDR_BAD_NTF_RULE
This file already exists in this category.	EDR_FILES_ALREADY_EXISTS
The commit flag has to be T when the send for approval flag is Y. You cannot send a file for approval if it has not been committed to the database.	EDR_FILES_COMMIT_FLAG_ERROR
The file category is invalid. Please provide a valid file category and try again.	EDR_FILES_INVALID_CATEGORY
The existing file cannot be overwritten because it is either already approved or is pending approval. You can only upload a new version of this file.	EDR_FILES_OVERWRITE_ERROR
The &EVENT Event has more than one active ERES Subscription. Contact your System Administrator to disable the extra Subscriptions.	EDR_MULTI_ERES_SUBSCRIP_ERR
PL/SQL 444087 - EDR API defined error message: The audit has been disabled for the specified table column.	EDR_PLS_STDMSG_AUDIT_DISABLED
You cannot perform any operation on closed documents	EDR_PSIG_DOC_CLOSED
You cannot close a document that has pending e-signatures associated to it.	EDR_PSIG_DOC_SIGNATURES

Message Text	Message Name
Duplicate Signer.	EDR_PSIG_DUPLICATE_SIGNER
Invalid Ad hoc Status.	EDR_PSIG_INVALID_ADHOC_STATUS
Invalid Document ID.	EDR_PSIG_INVALID_DOCUMENT
Invalid Document Status.	EDR_PSIG_INVALID_DOC_STATUS
Document status transition from &FROM to &TO is not allowed.	EDR_PSIG_INVALID_DOC_TRAN
Invalid E-signature Sequence.	EDR_PSIG_INVALID_SIGNATURE_SEQ
Invalid E-signature ID.	EDR_PSIG_INVALID_SIGNID
E-signature was not requested earlier for this document and signer combination. You need to request e-signature before posting the details.	EDR_PSIG_INVALID_SIGN_POST
You cannot add e-signatures to a document in &STATUS status.	EDR_PSIG_INVALID_SIGN_REQUEST
User &USER is invalid.	EDR_PSIG_INVALID_USER
The user you are trying to remove is not an ad hoc signer.	EDR_PSIG_NOT_ADHOC_SIGNER
Cannot sign a document with pending e-signature from a lower signer sequence.	EDR_PSIG_PENDING_SIGNATURE
You can cancel e-signatures only if the e-signature status is PENDING or the document status is in PENDING.	EDR_PSIG_SIGN_COMPLETE
Cannot validate Time zone. The Server Time zone or EDR: Server Time zone profile values are not same or have been set to NULL.	EDR_PSIG_TIMEZONE_ERROR
Invalid Document Format. Valid document formats are NULL, TEXT, HTML, text/html and text/plain.	EDR_PSIG_WFNTF_DOCFORMAT
Cannot generate notification for the given Notification ID. Check the Notification ID and the document format passed. The document format should be null, TEXT or HTML	EDR_PSIG_WFNTF_ERROR

Message Text	Message Name
Event Name: &EVENT_NAME, Event Key: &EVENT_KEY - Provide a valid PARENT_ERECORD_ID in payload. Refer to the Oracle E-Records Cookbook for payload parameter information.	EDR_VAL_INTER_EVENT_DB
The child event name or event key is invalid. The valid value in evidence store for event name is &CHILD_NAME and event key is &CHILD_KEY.	EDR_VAL_INVALID_CHILD_EVENT
The child e-record ID &ERECORD_ID is invalid.	EDR_VAL_INVALID_CHILD_ID
Event Name: &EVENT_NAME, Event Key: &EVENT_KEY - The DEFERRED parameter in the payload has invalid value. Refer to the Oracle E-Records Cookbook for payload parameter information.	EDR_VAL_INVALID_DEFERRED
Parent event name and event key are same as child event name and event key. You cannot establish circular relationship.	EDR_VAL_INVALID_EVENT
Event Name: &EVENT_NAME, Event Key: &EVENT_KEY - The payload parameters for a child event in the context of inter event processing are incorrect. Refer to the Oracle E-Records Cookbook for payload parameter information.	EDR_VAL_INVALID_INTER_EVENT
Event Name: &EVENT_NAME, Event Key: &EVENT_KEY - The value &PARAM_VALUE for parameter &PARAM_NAME is invalid.	EDR_VAL_INVALID_PARAM
Event Name: &EVENT_NAME, Event Key: &EVENT_KEY - The parent event name or event key is invalid. The valid value in evidence store for event name is &CHILD_NAME and event key is &CHILD_KEY.	EDR_VAL_INVALID_PARENT_EVENT
Event Name: &EVENT_NAME, Event Key: &EVENT_KEY - The parent e-record ID &ERECORD_ID is invalid.	EDR_VAL_INVALID_PARENT_ID

Message Text	Message Name
Event Name: &EVENT_NAME, Event Key: &EVENT_KEY - One or more mandatory parameters in the payload are missing or have invalid values. Refer to the Oracle E-Records Cookbook for payload parameter information.	EDR_VAL_INVALID_PAYLOAD
Event Name: &EVENT_NAME, Event Key: &EVENT_KEY - The event has a parent event named - &PARENT_EVENT. The relationship defined in the subscription of the parent event is invalid.	EDR_VAL_INVALID_RELATIONSHIP
Event Name: &EVENT_NAME, Event Key: &EVENT_KEY - The #WF_SOURCE_APPLICATION_TYPE parameter in the payload has invalid value. Refer to the Oracle E-Records Cookbook for payload parameter information.	EDR_VAL_INVALID_SOURCE

Glossary

Application Programming Interface (API)

A documented, supported method for communicating within or between modules.

Business Object

An independent item of significance in the business world. An example of a business object is a sales order.

Business Process API

An API that performs a transaction for the calling module, e.g., to hire an employee, enter an order, or cost a material movement transaction.

Entity

An item of significance in the business world, that has no meaning without reference to a business object. An example of an entity is a sales order header. A Sales Order Header is an entity of the business object sales order.

Group API

An API intended for use by other authorized Oracle Applications.

Module

A module is a collection of one or more business objects and the associated transactions. A module publishes APIs for other modules and accesses other modules through their published APIs. An example of a module is Oracle Inventory.

Public API

An API intended for use by all applications.

Private API

An API intended for use by the owning module only.

Index

A

AFASMSGs.pls, A-1

B

Batch File, 1-4
Business Objects, 4-1

C

Code Re-Use, 1-4
Consistent Behavior, 1-4

E

Ease of Integration, 1-4
Error Messages, A-3

F

FND_API, 1-3
FND_MESSAGE, 1-3
FND_MSG_PUB, A-1
FND_PUB_MSG, 1-3
formula information, importing, 1-2

G

GET MESSAGES, 1-5

I

importing formula information, 1-2
Insulation from Changes, 1-4

Item Cost
Structure, 3-2

N

NLS_LANGUAGE, A-3

O

Online User Interface (UI), 1-5
Oracle Messages, 1-5

P

p_api_version, 2-2, 3-5
p_commit, 2-2, 3-6
p_init_msg_list, 2-2, 3-6
p_validation_level, 2-2
PL/SQL, 1-2

R

Robust Validation, 1-4

S

Standard Input Parameters, 2-2
Stored procedures, 1-2
support policy, 1-3

T

Technical Overview, 3-1

X

x_message_count, 2-2

x_message_list, 2-2

x_msg_count, 3-6

x_msg_data, 3-6

x_return_status, 2-2, 3-6