

Oracle® Process Manufacturing

Process Planning API User's Guide

Release 11i

Part No. B10636-01

February 2003

Oracle Process Manufacturing Process Planning API User's Guide, Release 11i

Part No. B10636-01

Copyright © 2003 Oracle Corporation. All rights reserved.

Primary Author: Michele-Andrea Fields

Contributors: Sridhar Gidugu

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 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
Accessibility of Code Examples in Documentation	xi
Other Information Sources	xi
Online Documentation.....	xi
Related User's Guides.....	xii
Guides Related to All Products	xii
Oracle Applications User's Guide	xii
User Guides Related to This Product	xii
Accounting Setup User's Guide	xii
Cost Management User's Guide	xiii
Manufacturing Accounting Controller User's Guide	xiii
Oracle Financials Integration User's Guide.....	xiii
Inventory Management User's Guide.....	xiii
Physical Inventory User's Guide.....	xiv
Order Fulfillment User's Guide	xiv
Purchase Management User's Guide	xiv
Using Oracle Order Management with Process Inventory Guide	xiv
Process Execution User's Guide.....	xiv
Integration with Advanced Planning and Scheduling User's Guide	xv
MPS/MRP and Forecasting User's Guide	xv

Capacity Planning User's Guide	xv
Using Oracle Process Manufacturing with Oracle Manufacturing Scheduling.....	xv
Product Development User's Guide.....	xvi
Quality Management User's Guide	xvi
Regulatory Management User's Guide	xvi
Implementation Guide.....	xvii
System Administration User's Guide	xvii
API User's Guides	xvii
Installation and System Administration	xvii
Oracle Applications Concepts	xvii
Installing Oracle Applications	xviii
Upgrading Oracle Applications	xviii
Maintaining Oracle Applications	xviii
Oracle Applications System Administrator's Guide	xviii
Oracle Alert User's Guide	xviii
Oracle Applications Developer's Guide.....	xix
Oracle Applications User Interface Standards for Forms-Based Products	xix
Other Implementation Documentation	xix
Oracle Applications Product Update Notes	xix
Multiple Reporting Currencies in Oracle Applications	xix
Multiple Organizations in Oracle Applications	xix
Oracle Workflow Guide.....	xx
Oracle Applications Flexfields Guide.....	xx
Oracle eTechnical Reference Manuals	xx
Oracle Manufacturing APIs and Open Interfaces Manual	xx
Oracle Order Management Suite APIs and Open Interfaces Manual.....	xx
Oracle Applications Message Reference Manual.....	xx
Training and Support	xxi
Training.....	xxi
Support.....	xxi
Do Not Use Database Tools to Modify Oracle Applications Data	xxi
About Oracle	xxii
Your Feedback.....	xxii

1 API Introduction

Introducing the Process Planning APIs	1-2
Process Planning API Support Policy.....	1-3
Technical Requirements	1-3
Technical Overview.....	1-4
Basic Business Needs	1-4
Input Data Sources	1-5
Wrapper Function	1-5
Stored Procedure	1-6
Major Features	1-6
Insert Resources.....	1-6
Insert Resource Detail.....	1-6
Update Resources.....	1-6
Update Resource Detail.....	1-6
Delete Resources.....	1-7
Delete Resource Detail.....	1-7
Process Planning API Bill of Materials	1-7

2 Technical Overview

Structure for Process Planning Public APIs	2-2
Technical Details	2-3
Standard Parameters	2-4
Value-ID Conversion	2-5

3 Business Objects

Resources API	3-1
Validations.....	3-1
API Functions.....	3-3

A Messages and Errors

Handling Messages	A-1
Interpreting Error Conditions	A-5
Understanding Error Messages	A-5

Glossary

Index

Send Us Your Comments

Oracle Process Manufacturing Process Planning API User's Guide, Release 11i

Part No. B10636-01

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:

- FAX: 650-506-7200 Attn: Oracle Process Manufacturing
- Postal service:
Oracle Corporation
Oracle Process Manufacturing
500 Oracle Parkway
Redwood City, CA 94065
U.S.A.
- Electronic mail message to appsdoc_us@oracle.com

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 11i of the *Oracle Process Manufacturing Process Planning 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 Process Manufacturing Process Planning 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 Process Manufacturing Process Planning APIs.

- Chapter 1 describes how APIs are used, the basic business need of APIs, and the different Process Execution APIs offered.
- Chapter 2 describes how to use the Process Execution 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.

- Appendix B lists the full GMEPAPIS.pls file.
- 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 MetaLink.

- **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 MetaLink.
- **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 Process Manufacturing Process Planning 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 Process Manufacturing Process Planning 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.

Integration with Advanced Planning and Scheduling User's Guide

Oracle Process Manufacturing and Oracle Advanced Planning and Scheduling (APS) combine to provide an integrated solution for process manufacturers that can help increase planning efficiency. The integration 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 *Oracle Process Manufacturing Integration with Advanced Planning and Scheduling 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.

Regulatory Management User's Guide

The Oracle Process Manufacturing Regulatory Management application generates the Material Safety Data Sheets (MSDSs) required by authorities to accompany hazardous materials during shipping. You can create MSDSs from OPM Formula

Management with Regulatory or Production effectivities. The *Oracle Process Manufacturing Regulatory Management User's Guide* describes how to setup 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 Product Development Recipe 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 11i. 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 6i 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 11i. 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 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 Process Manufacturing Process Planning 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 Process Manufacturing Process Planning 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 Process Manufacturing Process Planning 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 Process Manufacturing Process Planning 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 Process Planning application. The topics discussed in this chapter are:

- Introducing the Process Planning APIs
- Basic Business Needs
- Major Features
- Process Planning API Bill of Materials

Introducing the Process Planning APIs

The Process Planning APIs let you import information from another system into the OPM 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 Process Planning windows.

This document describes the usage of the business objects that are stored as PL/SQL packages within the OPM 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

Resources are the assets you use to produce batches, including production equipment and employee labor. You can define each resource generally (OVENS) or specifically (OVEN 1, OVEN 2). For each resource, you must assign a component classification code for costing purposes. Optionally, you can group resources into resource classifications. For example, Ovens and Stoves may be grouped into Cooking Units.

What Is In This Document

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

Process Planning 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

Included API Functions

The following is a list of API functions, the business object name, and a brief explanation of each function:

Function	Business Object Name	Description
Insert, Update, and Delete Resource	GMP_RESOURCES_PUB	Creates, modifies, or deletes generic resource information.
Insert, Update, and Delete Resource Detail	GMP_RESOURCE_DTL_PUB	Creates, modifies, or deletes the resource detail information.

Process Planning API Support Policy

Process Planning 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

Process Planning 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

Process Planning APIs are designed to operate in an OPM 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 directly into Process Planning.

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, 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 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 three possible return code:

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

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 has been 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:

Insert Resources

This function can be used to add resource information.

Insert Resource Detail

This function is used to add resource detail information to the CR_RSRC_DTL table. It requires the resource information to be passed in for the resources being inserted, updated, or deleted.

Update Resources

This function lets you update resource information.

Update Resource Detail

This function lets you update a resource detail record. It requires the resource information to be passed as an input parameter.

Delete Resources

GMP_RESOURCES_PUB.resources uniquely identifies a resource and must be passed to the API so the resource can be deleted. This function then calls the update_resources function and sets the delete_mark to 1.

Delete Resource Detail

CR_RSRC_DTL.orgn_code and CR_RSRC_DTL.resources uniquely identifies a resource detail table. They must be passed to the API so that the resource detail can be deleted.

Process Planning API Bill of Materials

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

Package Name	File Names	Description
GMP_RESOURCES_PUB	GMPGRESS.pls, GMPGRESB.pls	A public resource package that the user defined function calls. The business API can be used for creating, modifying, or deleting a resource master.
GMP_RESOURCE_DTL_PUB	GMPRSDTS.pls, GMPRSDTB.pls	A public resource package that the user defined function calls. The business API can be used for creating or modifying the resource detail associated to the resource master.

Technical Overview

The public Process Planning 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 Process Planning Public APIs
- Technical Details
- Standard Parameters

Structure for Process Planning Public APIs

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

Resources API

Object Type	Name
Package Specification File	GMPGRESS.pls
Package Body File	GMPGRESB.pls
Package	gmp_resources_pub
Procedure - Insert Resources	insert_resources
Procedure - Update Resources	update_resources
Procedure - Delete Resources	delete_resources

Resource Detail API

Object Type	Name
Package Specification File	GMPRSDTS.pls
Package Body File	GMPRSDTB.pls
Package	gmp_resource_dtl_pub
Procedure - Insert Resource Detail	insert_resource_dtl
Procedure - Update Resource Detail	update_resource_dtl
Procedure - Delete Resource Detail	delete_resources

Technical Details

The Resources APIs are invoked from a user wrapper calling function with resource attributes passed to the procedure through a record or table format. The wrapper function is responsible for connecting to the database as an appropriate user with the necessary privileges. It passes the appropriate parameters into the stored procedure and is responsible for handling the return code from the procedure.

Business Object APIs have the following procedures:

- Creation of new resources (Generic and Plant Resources)
- Modification of a resources (Generic and Plant Resources)
- Deletion of a resources (Generic and Plant Resources)

These stored procedures are called from a user wrapper function, which executes the procedure and receives the return status and messages from the execution.

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`

Standard OUT parameters:

- `x_return_status`
- `x_message_count`
- `x_message_list`

Parameter	Type	IN/OUT	Required	Validation
<code>p_api_version</code>	<code>varchar2</code>	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.
<code>p_init_msg_list</code>	<code>boolean</code>	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 <code>FND_API.G_FALSE</code> which means that the API does not initialize the message list. If multiple APIs are called in same session, then you must pass this as <code>TRUE</code> , otherwise messages accumulate on the stack.

Parameter	Type	IN/OUT	Required	Validation
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. 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_message_count	number	OUT	N	Specifies the number of messages added to message list.
x_message_list	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 usually 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. Those values are meaningful and are used for display purposes. In general, APIs operate only on IDs.

For example, an item is represented by an ID, 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 a value, the 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:

- Either 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 standard 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 status of error.

Business Objects

Resources API

The APIs in this document are business objects that can create or change the resources information. This information is comprised of a header that provides the resources, resources description, resource_class, std_usage_um, cost component code, minimum_capacity, maximum_capacity, capacity_uom, capacity_constraints, utilization, and efficiency.

Following are the actions that can be taken on resources:

- Insert Resources (Generic and Plant Resources)
- Update Resources (Generic and Plant Resources)
- Delete Resources (Generic and Plant Resources)

Validations

Resource Header

The Resource Header contains information applicable to the entire resource.

Parameter	Type	Length	Default	Required	Validation
Resources	Varchar2	16	N/A	Y	The primary key for the resource header.
Resource_desc	Varchar2	40	N/A	Y	This field cannot be NULL.
Resource_Class	Varchar2	8	N/A	N	Valid values from cr_rsrc_cls where delete_mark is equal to 0.
Cost_cmpntcls_id	Number	10	N/A	Y	Valid values from cm_cmpt_mst where delete_mark is equal to 0.

Parameter	Type	Length	Default	Required	Validation
Minimum_capacity	Number	variable	N/A	N	Value must be less than the maximum capacity.
Maximum_capacity	Number	variable	N/A	N	Value must be greater than the minimum capacity.
Capacity_uom	Varchar2	4	N/A	N	Valid Values from sy_uoms_mst where delete_mark is equal to 0 and is a required field if the minimum value or maximum value is entered.
Capacity_constraint	Number	5	N/A	N	Values must be 0 or 1.
Capacity_tolerance	Number	variable	null	N	Cannot be a negative value.
Utilization	Number	variable	null	N	Cannot be a negative value.
Efficiency	Number	variable	null	N	Cannot be a negative value.

Resource Details

The Resource Details contains the resources that are associated with a plant. The GMP_RESOURCE_DTL API defaults the minimum, maximum values, capacity tolerance, utilization, and efficiency values from the generic resources if the resource was already defined in generic resources.

Parameters	Type	Length	Default	Required	Validation
Orgn_code	Number	4	N/A	Y	Must be a valid orgn_code from sy_orgn_mst.
Resources	Varchar2	16	N/A	Y	Must be a valid value from cr_rsrc_mst.
Group_Resource	Varchar2	16	N/A	Y	Must be a valid value from cr_rsrc_mst.
Assigned_qty	Number	variable	N/A	Y	This parameter cannot be less than 0.
Daily_avail_use	Number	variable	N/A	Y	This parameter cannot be less than 0.
Nominal_cost	Number	variable	N/A	Y	This parameter cannot be less than 0.
Usage_um	Varchar2	4	N/A	Y	Must be a valid value from sy_uoms_mst.

Parameters	Type	Length	Default	Required	Validation
schedule_ind	Number	variable	null	N	Valid values are 0, 1, and 2.
Inactive_ind	Number	variable	N/A	N	Valid values are 0 and 1.
Min_capacity	Number	variable	null	N	Cannot be greater than maximum and less than 0.
Max_Capacity	Number	variable	null	N	Cannot be less than minimum capacity.
Ideal_capacity	Number	variable	null	N	Value should be between minimum and maximum.
Capacity_constraint	Number	5	null	N	Valid values must be 0 or 1.
Capacity_Tolerance	Number	variable	null	Y	Cannot be a negative value.
Utilization	Number	variable	null	N	Cannot be a negative value.
Efficiency	Number	variable	null	N	Cannot be a negative value.

API Functions

Insert Resources

This function can be used to add resources information. The resources information is stored as a record and this function calls the check_data function which returns the success code S after the data in the record is validated. The record is inserted by calling the CR_RSRC_MST_PKG.insert_row function.

```
GMP_RESOURCES_PUB.insert_resources
( p_api_version          IN          NUMBER    => 1
, p_init_msg_list       IN          BOOLEAN   => TRUE
, p_commit              IN          BOOLEAN   => FALSE
, p_resources           IN          cr_rsrc_mst%ROWTYPE
, x_message_count       OUT         NUMBER
, x_message_list        OUT         VARCHAR2
, x_return_status       OUT         VARCHAR2
);
```

Insert Resource Detail

This function is used to add resource detail information to the CR_RSRC_DTL table. It requires the resource_id to be passed in for the resources being inserted, updated, or deleted. The resource_id and the plant_code used must be unique as they define the primary key for the CR_RSRC_DTL table.

```
GMP_RESOURCES_PUB.insert_resource_dtl
( p_api_version          IN          NUMBER    => 1
, p_init_msg_list        IN          BOOLEAN   => TRUE
, p_commit               IN          BOOLEAN   => FALSE
, p_resource_dtl         IN          cr_rsrc_dtl%ROWTYPE
, p_rsrc_instances       IN          resource_instaces_tbl
, x_message_count        OUT         NUMBER
, x_message_list         OUT         VARCHAR2
, x_return_status        OUT         VARCHAR2
);
```

Update Resources

This function lets you update resources. The record that is passed is validated and then the update CR_RSRC_MST.update_row function is invoked.

```
GMP_RESOURCES_PUB.update_resources
( p_api_version          IN          NUMBER    => 1
, p_init_msg_list        IN          BOOLEAN   => TRUE
, p_commit               IN          BOOLEAN   => FALSE
, p_resources            IN          cr_rsrc_mst%ROWTYPE
, x_message_count        OUT         NUMBER
, x_message_list         OUT         VARCHAR2
, x_return_status        OUT         VARCHAR2
);
```

Update Resource Detail

This function lets you update a resource detail record. It requires the cr_rsrc_dtl.resource_id, or plant_code and resource_id to be passed in. Additionally, the column to be updated and the value of that column must be passed to the function.

```
GMP_RESOURCE_DTL_PUB.update_resource_dtl
( p_api_version          IN          NUMBER    => 1
, p_init_msg_list        IN          BOOLEAN   => TRUE
, p_commit               IN          BOOLEAN   => FALSE
, p_resources            IN          cr_rsrc_dtl%ROWTYPE
, x_message_count        OUT         NUMBER
, x_message_list         OUT         VARCHAR2
, x_return_status        OUT         VARCHAR2
);
```

Delete Resources

The GMP_RESOURCES_PUB.resources function uniquely identifies a resource and must be passed to the API so the resource can be deleted.

This function then calls the `update_resources` function and sets the `delete_mark` to 1.

```
GMP_RESOURCES_PUB.delete_resources
( p_api_version          IN          NUMBER    => 1
, p_init_msg_list        IN          BOOLEAN   => TRUE
, p_commit               IN          BOOLEAN   => FALSE
, p_resources            IN          cr_rsrc_mst.resources%TYPE => NULL
, x_message_count        OUT         NUMBER
, x_message_list         OUT         VARCHAR2
, x_return_status        OUT         VARCHAR2
);
```

Delete Resource Detail

The `CR_RSRC_DTL.resource_id` or `orgn_code` and `resource_id` uniquely identifies a resource detail table. It must be passed to the API so the resource detail can be deleted.

```
GMP_RESOURCES_PUB.delete_resources
( p_api_version          IN          NUMBER    => 1
, p_init_msg_list        IN          BOOLEAN   => TRUE
, p_commit               IN          BOOLEAN   => FALSE
, p_orgn_code            IN          cr_rsrc_dtl.orgn_code%TYPE
, p_resources            IN          cr_rsrc_dtl.resources%TYPE
, x_message_count        OUT         NUMBER
, x_message_list         OUT         VARCHAR2
, x_return_status        OUT         VARCHAR2
);
```

Messages and Errors

Handling Messages

APIs put result messages into a message list. Programs calling APIs can then get 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 out message names by using the standard functions provided by the message dictionary. It also stores these messages in database tables and reports off 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, developers can 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. They can then call `FND_MSG_PUB.Add` to fetch the messages off the message dictionary stack and add it to the API message list.

To get a message from the API message list, API callers can 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.

<code>x_msg_count</code>	OUT	NUMBER
<code>x_msg_data</code>	OUT	VARCHAR2

Example:

```
PROCEDURE allocate_line (
    p_material_details IN      gme_material_details%ROWTYPE
    ,p_alloc_type       IN      NUMBER DEFAULT 9
    ,p_api_version      IN      NUMBER := gme_api_pub.api_version
    ,p_validation_level IN      NUMBER := gme_api_pub.max_errors
    ,p_init_msg_list    IN      BOOLEAN := FALSE
    ,p_commit           IN      BOOLEAN := FALSE
    ,x_message_count    OUT     NUMBER
    ,x_message_list     OUT     VARCHAR2
    ,x_return_status    OUT     VARCHAR2
    ,p_del_exist_alloc  IN      BOOLEAN := FALSE) IS
```

```

l_api_name    CONSTANT VARCHAR2 (30)                := 'ALLOCATE_LINE';
l_material_details  gme_material_details%ROWTYPE;
l_batch_hdr    GME_BATCH_HEADER%ROWTYPE;
BEGIN
  gme_debug.log_initialize('AllocateLine');
  -- Initialize message list and count if needed
  IF p_init_msg_list THEN
    fnd_msg_pub.initialize;
    gme_api_pub.error_count := 0;
  END IF;

  IF NOT gme_api_pub.setup_done THEN
    gme_api_pub.setup_done := gme_api_pub.setup;

    IF NOT gme_api_pub.setup_done THEN
      RAISE FND_API.g_exc_error;
    END IF;
  END IF;

  -- Make sure we are call compatible
  IF NOT FND_API.compatible_api_call (
    gme_api_pub.api_version
    ,p_api_version
    ,l_api_name
    ,'gme_api_allocate_line') THEN
    RAISE FND_API.g_exc_error;
    gme_api_pub.log_message (l_package_name||'.'||l_api_
name||'.'||'INVALID_VERSION');
  END IF;

  gme_api_grp.set_timestamp;
  l_material_details := p_material_details;
  gme_api_allocate_line_pvt.allocate_line (
    p_gme_material_details => p_material_details
    ,p_alloc_type    => p_alloc_type
    ,x_gme_material_details => l_material_details
    ,x_return_status => x_return_status
    ,p_del_exist_alloc => p_del_exist_alloc);

  gme_debug.put_line(l_package_name||'.'||l_api_name||'.'||'Return status
from private allocate_line is '||x_return_status);
  IF x_return_status = FND_API.G_RET_STS_SUCCESS THEN
    IF p_commit = TRUE THEN
      gme_debug.put_line (l_package_name||'.'||l_api_name||'.'||'Calling

```

```

Save_batch');
    l_batch_hdr.batch_id := p_material_details.batch_id;
    save_batch(l_batch_hdr, x_return_status);
    IF x_return_status = FND_API.G_RET_STS_SUCCESS THEN
        COMMIT;
    ELSE
        gme_api_pub.log_message ('BATCH_SAVE_FAILED');
        RAISE FND_API.g_exc_error;
    END IF;
END IF;
IF error_count = 0 THEN
    log_message ('GME_API_LINE_ALLOCATED');
END IF;
END IF;
gme_debug.put_line('Completed '||l_api_name||' at '||to_
char(sysdate,'MM/DD/YYYY HH24:MI:SS'));

fnd_msg_pub.count_and_get (
    p_count => x_message_count
    ,p_data => x_message_list);
EXCEPTION
    WHEN FND_API.g_exc_error THEN
        x_return_status := FND_API.G_RET_STS_ERROR;
        fnd_msg_pub.count_and_get (
            p_encoded => FND_API.g_false
            ,p_count => x_message_count
            ,p_data => x_message_list);
    WHEN FND_API.g_exc_unexpected_error THEN
        x_return_status := FND_API.g_ret_sts_unexp_error;
        fnd_msg_pub.count_and_get (
            p_encoded => FND_API.g_false
            ,p_count => x_message_count
            ,p_data => x_message_list);
    WHEN OTHERS THEN
        x_return_status := FND_API.g_ret_sts_unexp_error;
        fnd_msg_pub.add_exc_msg (l_package_name, l_api_name);
        fnd_msg_pub.count_and_get (
            p_encoded => FND_API.g_false
            ,p_count => x_message_count
            ,p_data => x_message_list);
END allocate_line;

```


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 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
Resource Not found	CR_BAD_RESOURCE
Capacity is not within given range.	CR_CAPACITY_NOT_IN_RANGE
Invalid Resource Class	CR_INVALID_RSRC_CLASS
Value must be larger than zero.	CR_POSITIVE
Unexpected &ERROR occurred.	GMD_UNEXPECTED_ERROR
&FIELD = &VALUE is a Not a Valid Value	GMP_INVALID_VALUE
Minimum Capacity is greater than Maximum Capacity	GMP_MIN_MAX_CAPACITY
&VALUE_REQUIRED is Required	GMP_VALUE_REQUIRED

Message Text	Message Name
&Value is Required	GMP_VALUE_REQUIRED
Invalid Organization Code	IC_ORGNCODERR
End Date must be greater than Start Date.	MR_STARTENDDATERR
Duplicate Record	PS_DUP_REC
Minimum has to be less than maximum.	QC_MIN_MAX_SPEC
You cannot duplicate a record which is marked for purge	SY_INVALID_DUPLICATION
Invalid UOM Code	SY_INVALID_UM_CODE
Entry is required	SY_REQUIRED

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
API Functions, 3-3
Application Programming Interface (API), 1

B

Batch File, 1-5
Business Object, 1
Business Objects, 3-1
Business Process API, 1

C

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

D

Delete Resource Detail, 1-7, 3-5
Delete Resources, 1-7
delete_resources, 2-2

E

Ease of Integration, 1-4
Entity, 1
Error Messages, A-5

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, 1-6
GMP_RESOURCE_DTL_PUB, 1-3, 1-7
gmp_resource_dtl_pub, 2-2
GMP_RESOURCES_PUB, 1-3, 1-7
gmp_resources_pub, 2-2
GMPGRESB.pls, 1-7, 2-2
GMPGRESS.pls, 1-7, 2-2
GMPRSDTB.pls, 2-2
GMPRSDTS.pls, 1-7, 2-2
Group API, 1

I

importing formula information, 1-2
Insert Resource Detail, 1-6, 3-3
Insert Resources, 1-6, 3-3
insert_resource_dtl, 2-2
insert_resources, 2-2
Insulation from Changes, 1-4
Item Cost
 Structure, 2-2

M

Module, 1

N

NLS_LANGUAGE, A-5

O

Online User Interface (UI), 1-5
Oracle Messages, 1-6

P

p_api_version, 2-4
p_commit, 2-5
p_init_msg_list, 2-4
PL/SQL, 1-2
Private API, 2
Public API, 1

R

Resource Detail API, 2-2
Resource Details, 3-2
Resource Header, 3-1
Resources API, 2-2, 3-1
Robust Validation, 1-4

S

Standard Parameters, 2-4
Stored procedures, 1-2
support policy, 1-3

T

Technical Details, 2-3
Technical Overview, 2-1

U

Update Resource Detail, 1-6, 3-4
Update Resources, 1-6, 3-4
update_resource_dtl, 2-2
update_resources, 2-2

X

x_msg_count, 2-5
x_msg_data, 2-5
x_return_status, 2-5