Oracle® Process Manufacturing

Oracle Order Management with Process Inventory API User's Guide

Release 11i

Part No. B10662-01

January 2003

ORACLE®

Oracle Process Manufacturing Oracle Order Management with Process Inventory API User's Guide, Release 11i

Part No. B10662-01

Primary Author: Michele-Andrea Fields

Contributors: Neelima Chekuri

Contents

1 API Introduction

2 Technical Overview

3 Business Objects

A Messages and Errors

Glossary

Index

Send Us Your Comments

Oracle Process Manufacturing Order Order Management with Process Inventory API User's Guide, Release 11*i*

Part No. B10662-01

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 11*i* of the *Oracle Order Management with Process Inventory 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 Order Management with Process Inventory 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 Order Management with Process Inventory APIs.

- Chapter 1 describes how APIs are used, the basic business need of APIs, and the different Oracle Order Management with Process Inventory APIs offered.

- Chapter 2 describes the technical aspect of the APIs.

- Chapter 3 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 11i. 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 Order Management with Process Inventory 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 Order Management with Process Inventory 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 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 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 Order Management with Process Inventory 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 Order Management with Process Inventory 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 Oracle8*i* 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 Order Management with Process Inventory 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 Order Management with Process Inventory 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**.

1

API Introduction

This document describes the Application Program Interfaces (APIs) that support external interfaces to Oracle Order Management with Process Inventory application. The topics discussed in this chapter are:

- Introducing the Order Management with Process Inventory APIs
- Basic Business Needs
- Major Features
- Order Management with Process Inventory API Bill of Materials

Introducing the Order Management with Process Inventory APIs

The Order Management with Process Inventory 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 Order Management with Process Inventory 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

What Is In This Document

This document describes the basic business needs, major features, architecture, and components for the Order Management with Process Inventory 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 Order Management with Process Inventory application.

Order Management with Process Inventory 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

Order Management with Process Inventory API Support Policy

Order Management with Process Inventory APIs are supported by Oracle. This means:

n   Oracle provides objects and libraries needed to use the APIs and the documentation for their use.

n   Oracle ensures that the APIs function as designed.

n   Oracle does not support customer generated programs that use the APIs.

Technical Requirements

Order Management with Process Inventory APIs make use of the following standard Oracle Applications packages:

n   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.

n   FND_MESSAGE - the standard Oracle Applications messaging function. This is used by the stored procedure to report status and error handling.

n   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

Order Management with Process Inventory APIs are designed to operate in an OPM 11i environment only. They offer the following basic API functionality:

n   Creating, updating, and deleting information.

n   Proper encapsulation.

n   Synchronous processing following the business hierarchy.

n   Detailed and translatable error messages.

Basic Business Needs

These APIs let you feed information directly into Order Management with Process Inventory.

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:

n   S for success

n   E for error

n   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:

Allocate OPM Orders

The Allocate OPM Orders API is a business object that can create, modify, or delete OPM reservation (allocation) information for Order Management.

Pick Confirm

The Pick Confirm API is a business object that pick confirms, or stages the inventory for, a Process Move Order Line or a Delivery Detail line.

Order Management with Process Inventory API Bill of Materials

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

n   GMIOMAPB.pls: GMI_OM_ALLOC_API_PUB

n   GMIPCAPB.pls: GMI_PICK_CONFIRM_PUB

2

Technical Overview

The public Order Management with Process Inventory 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 Order Management with Process Inventory Public APIs
- Technical Details
- Standard Parameters

Structure for Order Management with Process Inventory Public APIs

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

Allocate OPM Orders API

Object Type	Name
Package Specification File	GMIOMAPB.pls
Package Body File	GMIOMAPS.pls
Package	gmi_om_alloc_api_pub
Procedure - Allocation	allocate_opm_orders

Pick Confirm API

Object Type	Name
Package Specification File	GMIPCAPB.pls
Package Body File	GMIPCAPS.pls
Package	gmi_pick_confirm_pub
Procedure - Pick Confirm	pick_confirm

Technical Details

Following are the technical details for these APIs:

Allocate OPM Orders

The Allocate OPM Orders API is invoked from a user wrapper calling function with allocation attributes passed to the procedure through a format that is detailed in the "Business Objects" section. 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.

Allocate OPM Orders API handles the following:

1. Creation of a new reservation

2. Modification of an exiting reservation

3. Deletion of a reservation

Only pending transactions that are not staged or deleted with a doc type of OMSO can be updated or deleted with the API.

If there is a move order associated with the sales order lines for which the API is called, then it is updated with the changed quantities. If there is no associated move order, then no change is made.

Pick Confirm

The Pick Confirm API is invoked from a user wrapper calling function with allocation attributes passed to the procedure through a format that is detailed in the "Business Objects" section. 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.

Pick Confirm API handles the pick confirming of a move order line or a single delivery detail line.

The following events take place at the time of pick confirming.

n The inventory is marked as staged (staged_ind in IC_TRAN_PND is updated to 1).

- The quantity_delivered field on the move order line is updated. If all the deliveries are staged for the move order, then the status of the move order is updated to closed.

- The released status of the delivery detail line is updated from S - Released to Warehouse to Y - Staged.

- The Shipping API is called to update or split the delivery line.

If a particular delivery line is partially allocated, then a back order line with the status B is created in wsh_delivery_details for the remaining quantity.

If the move order is partially allocated, then the following steps take place at the time of ship confirming:

1. A back order delivery detail line is created in wsh_delivery_details with the unallocated quantity.

2. The original delivery detail line has a status of Y - Staged.

3. The move order line is updated with quantity delivered and the status of the move order line becomes closed.

If the delivery detail is passed to the API and move order line ID is Null then,

   a. If the delivery line is partially allocated, then a back order is created in wsh_delivery_details with remaining quantity, and move order quantities are updated. The delivery line status becomes Y - Staged.

   b. If the delivery line is fully allocated, then the move order is updated with the quantity. The delivery line has a status of Y - Staged.

In both cases, if this is the only delivery detail line for the move order line, then the move order line status is closed. If there are multiple delivery detail lines associated with the move order line, then there is no change in the status of the move order.

If a move order line is passed and the delivery detail is null, then only delivery details that have allocations against them are staged, and the move order quantities are updated. The move order is not closed unless all the delivery detail lines are transacted or back ordered.

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
p_api_version	varchar2	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	varchar2	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 same session, then you must pass this as TRUE, otherwise messages accumulate on the stack.
p_commit	varchar2	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_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 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:

n   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.

n   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.

n   This message is not an error message. The API continues with its standard processing.

n   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.

3

Business Objects

Allocate OPM Orders

The Allocate OPM Orders API is a business object that can create, modify, or delete OPM reservation (allocation) information for Order Management.

Procedure Specification

```
GMI_OM_API_PUB.Allocate_OPM_Orders
( p_api_version     IN   NUMBER      => 1
, p_init_msg_list   IN   BOOLEAN     => TRUE
, p_commit          IN   BOOLEAN     => FALSE
,p_trans_rec        IN   GMI_OM_API_PUB.trans_rec
, x_message_count   OUT  NUMBER
, x_message_list    OUT  VARCHAR2
, x_return_status   OUT  VARCHAR2
);
```

Input Record Structure

The following is the type definition of an input record.

```
TYPE trans_rec is RECORD(
    trans_id        IC_TRAN_PND.TRANS_ID%TYPE
  , line_id         IC_TRAN_PND.LINE_ID%TYPE
  , lot_id          IC_LOTS_MST.LOT_ID%TYPE
  , lot_no          IC_LOTS_MST.LOT_NO
  , sublot_no       IC_LOTS_MST.SUBLOT_NO
  , location        IC_LOCT_MST.LOCATION%TYPE
  , trans_qty       IC_TRAN_PND.TRANS_QTY%TYPE
  , trans_qty2      IC_TRAN_PND.TRANS_QTY%TYPE
  , trans_um        IC_TRAN_PND.TRANS_UM%TYPE
  , reason_code     SY_REAS_CDS.REASON_CODE%TYPE
```

```
        , trans_date     DATE
        , line_detail_id  NUMBER
        , action_code     VARCHAR2
    )
```

API Definition

Following is the definition of Allocate OPM Orders:

Parameter	Type	Length	Default	Required	Validation
Trans_id	number	10	0	N	For Update and Delete, ensure that the transaction is not staged, deleted, or completed.
Line_id	number	10	N/A	Y	Validate the line_id against oe_order_lines_all and ensure that the line_status is not cancelled, shipped, or closed.
Lot_no	varchar2	32	0	N	Must be blank if ic_item_mst.lot_ctl is equal to 0. If item is lot controlled, then lot_no must not equal IC$DEFAULT_LOT (or lot_id is equal to 0).
sublot_no	varchar2	32	0	N	Sublot no must not be passed if the item is not sublot controlled.
lot_id	Number	10	0	N	Verify from ic_lots_mst that the item, lot, sublot (if the item is sublot controlled) combination exists, the lot is active (inactive_ ind), not expired (expiration_date), and not deleted (delete_ mark).
Location	Varchar2	16	N/A	N	Verify that the item and the warehouse are location controlled. Validate that location exists for the warehouse in ic_loct_mst, and that it is active (ic_loct_mst.delete_mark) If whse is non-validated location_controlled (ic_whse_mst.loct_ ctl = 2), then do not validate the location, but verify that the length is less than or equal to 16.
Reason_ code	Varchar2	4	NULL	N	Validate against sy_reas_cds. Verify that delete_mark is not set (sy_reas_cds.delete_mark).
Trans_date	Date	variable	Sched_ ship_ date from the order line	N	If trans_date is not passed, then use sched_ship_date from the order_line. Verify that the date does not fall in a closed period.
Trans_qty	Number	variable	0	N	Must be greater than 0 if the action is insert or update, but 0 is valid if the action is delete.

Parameter	Type	Length	Default	Required	Validation
Trans_qty2	Number	variable	0	N	Must be greater than 0 if the item is dual controlled and the action is insert or update. If this is not supplied, then it defaults for dual 1 and 2 items. If it is not supplied for dual3, then it errors out. If it is passed, then check for deviation.
Trans_um	Varchar2	3	UOM	Y	This errors out if the value UOM is not passed.
Line_detail_id	Number	variable	NULL	N	If this is passed, then it verifies for that wsh_delivery_details.source_line_id from this line_detail_id (delivery_detail_id) = line_id of the order line If this is not passed, then it errors out if there are multiple delivery details for the source_line id. If there is only one, then it takes it as default.
Action_code	Varchar2	6	N/A	Y	Valid values are INSERT, UPDATE, and DELETE.

Pick Confirm

The Pick Confirm API is a business object that pick confirms, or stages the inventory for, a Process Move Order Line or a Delivery Detail line.

Procedure Specifications

```
GMI_OM_API_PUB.Allocate_OPM_Orders
( p_api_version      IN      NUMBER    => 1
, p_init_msg_list    IN      BOOLEAN   => TRUE
, p_commit           IN      BOOLEAN   => FALSE
 ,p_mo_line_id       IN      NUMBER
 ,p_line_detail_id   IN      NUMBER
, x_message_count    OUT     NUMBER
, x_message_list     OUT     VARCHAR2
, x_return_status    OUT     VARCHAR2
);
```

API Definition

Parameter	Type	Length	Default	Required	Validation
delivery_detail_id	number	10	0	N	This is required if mo_line_id is equal to 0.
					The delivery_detail_id must exist in the wsh_delivery_details table.
					The status of this delivery detail line must be S - released.
					Allocations must exist in ic_tran_pnd for the delivery line.
					If a move order line is passed, then this delivery_detail_id must belong to that mo_line_id.
mo_line_id	number	10	0	N	This is required if line_detail_id is not passed.
					The move order line must exist in the ic_txn_request_lines table.
					The status of this move order is open and the move order type is 3.
					There must be at least one delivery detail line whose status is S, associated with the move order line and the quantity_detailed field cannot be NULL.
					The organization must be process enabled.
					If a line_detail_id is passed, then the move order line must belong to the delivery detail line.

Special Considerations

- If only a move_order line is passed, then all the delivery details with allocations that belong to the move order line are pick confirmed.

- If only the delivery_detail_id is passed, then only that is pick confirmed and the rest of the delivery details are unaffected.

- If a move order line and a delivery detail line are passed, then only the delivery detail line is pick confirmed, not the entire move order.

A

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:

n   message count

n   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
```

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 apersand (&) is a token, or placeholder, for an actual value that is populated at runtime.

Message Text	Message Name
Action Code &ACTION_CODE is not valid. Line_id : &LINE_ID.	GMI_API_INVALID_ACTION_CODE
Apps UOM not found for OPM UOM &OPM_UOM. Line_id : &LINE_ID.	GMI_API_APPS_UOM_NOT_FOUND
Cannot retrieve the Fiscal Year from the company's OPM Inventory Calendar.	IC_CAL_FISCALYR_ERR
Deviation Error. Trans_qty : &QTY, Trans_qty2 : %QTY2, Item_no : &ITEM_NO, Line_id : &LINE_ID.	GMI_API_DEVIATION_ERR
Invalid Company Code.	IC_COCODEERR

Message Text	Message Name
Invalid line status &LINE_STATUS for line_id &LINE_ID	GMI_API_INVALID_LINE_STATUS
Invalid Line_detail_id &LINE_DETAIL_ID for Line_id &LINE_ID	GMI_API_INVALID_LINE_DETAIL_ID
Invalid Trans_id.The transaction for trans_id &TRANS_ID is already completed. Line_id : &LINE_ID	GMI_API_TRANS_COMPLETED
Invalid Trans_id.The transaction for trans_id &TRANS_ID is already deleted. Line_id : &LINE_ID	GMI_API_TRANS_DELETED
Invalid Trans_id.The transaction for trans_id &TRANS_ID is already staged. Line_id : &LINE_ID	GMI_API_TRANS_STAGED
Invalid warehouse code &WHSE_CODE	IC_API_INVALID_WHSE_CODE
INVENTORY CALENDAR - an unknown error has occurred while attempting to validate the date against the inventory calendar. Please notify your Systems administrator.	ICCAL_GENL_ERR
Item &ITEM_NO is not lot controlled. Line_id : &LINE_ID	GMI_API_NOT_LOT_CONTROL_ITEM
Item &ITEM_NO or warehouse &WHSE_CODE is not location controlled. Location : &LOCATION, Line_id : &LINE_ID.	GMI_API_NON_LOCT_CTL_ITEM_WHSE
Location &LOCATION is not active . Line_id : &LINE_ID	GMI_API_INACTIVE_LOCATION
Location &LOCATION is not valid . Line_id : &LINE_ID	GMI_API_INVALID_LOCATION
Location is Required. Item &ITEM_NO and the warehouse &WHSE_CODE are location controlled. Line_id : &LINE_ID	GMI_API_LOCATION_REQUIRED
Lot &LOT_NO is deleted. Item_no : &ITEM_NO, Line_id : &LINE_ID.	GMI_API_DELETED_LOT
Lot &LOT_NO is expired. Item_no : &ITEM_NO, Line_id : &LINE_ID.	GMI_API_EXPIRED_LOT
Lot &LOT_NO is not active. Item_no : &ITEM_NO, Line_id : &LINE_ID.	GMI_API_INACTIVE_LOT

Message Text	Message Name
Lot &LOT_NO is indivisible, Over-picking is not enabled and the trans_qty is less than the available quantity. Item_no :&ITEM_NO, Line_id : &LINE_ID.	GMI_API_LOT_INDIV_NO_OVERPIC
Lot &LOT_NO is not Nettable. Line_id : &LINE_ID.	GMI_API_LOT_NOT_NETTABLE
Lot &LOT_NO is not orderable. Line_id : &LINE_ID.	GMI_API_LOT_NOT_ORDERABLE
Lot &LOT_NO is rejected. Line_id : &LINE_ID.	GMI_API_REJECTED_LOT
Lot not found for Lot_no : &LOT_NO, Sublot_no &SUBLOT_NO, Lot_id &LOT_ID. Item_no : &ITEM_NO, Line_id : &LINE_ID.	GMI_API_LOT_NOT_FOUND
Lot status &LOT_STATUS is not found for Lot : &LOT_NO. Line_id = &LINE_ID.	GMI_API_LOT_STATUS_NOT_FOUND
Non validated Location &LOCATION length can not exceed 16 chars. Line_id : &LINE_ID.	GMI_API_LOCT_INVALID_LENGTH
OPM item not found for Inventory_item_id &INVENTORY_ITEM_ID, Organization_id &ORGANIZATION_ID , Line_id &LINE_ID	GMI_API_ITEM_NOT_FOUND
OPM UOM not found for Apps UOM &APPS_UOM. Line_id : &LINE_ID.	GMI_API_OPM_UOM_NOT_FOUND
Over picking is not allowed. Item_no : &ITEM_NO, Line_id : &LINE_ID.	GMI_API_OVERPICK_NOT_ALLOWED
Quantity needs to be entered for action code &ACTION_CODE . Line_id : &LINE_ID.	GMI_API_QTY_REQ_FOR_ACT_CODE
Sublot is not required to be specified. Item &ITEM_NO is not sublot controlled. Line_id &LINE_ID	GMI_API_SUBLOT_NOT_REQUIRED
The Item &ITEM_NO is a no controls item Line_id : &LINE_ID.	GMI_API_NO_CONTROL_ITEM
The Item &ITEM_NO is a Non inventory item. Line_id : &LINE_ID.	GMI_API_NONINV_ITEM
The Item &ITEM_NO is not active. Line_id : &LINE_ID.	GMI_API_INACTIVE_ITEM
The Line_id &LINE_ID is not valid.	GMI_API_INVALID_LINE_ID

Message Text	Message Name
The Organization was found to be invalid while trying to validate the OPM Inventory Calendar.	IC_INVALID_ORGN_ERR
The Period cannot be retrieved from the OPM Inventory Calendar.	IC_CAL_CLOSED_IND_ERR
The Reason_code &REASON_CODE is not valid . Line_id : &LINE_ID, Trans_id : &TRANS_ID, Line_detail_id : &LINE_DETAIL_ID	GMI_API_INVALID_REASON_CODE
The Warehouse &WHSE_CODE is not location controlled. Item_no : &ITEM_NO, Line_id : &LINE_ID.	GMI_API_NOT_LOCT_CONTROL_WHSE
This Allocation would drive the Inventory -ve. Line_id : &LINE_ID	GMI_API_INVQTYNEG
Trans_id &TRANS_ID is not valid. Line_id : &LINE_ID.	GMI_API_INVALID_TRANS_ID
Trans_id is NOT Required for action code : &ACTION_CODE , Line_id : &LINE_ID	GMI_API_TRANS_ID_NOT_REQUIRED
Trans_id is Required for action code : &ACTION_CODE , Line_id : &LINE_ID	GMI_API_TRANS_ID_REQUIRED
Trans_qty2 is required to be passsed for a dual3 item &ITEM_NO. Line_id &LINE_ID.	GMI_API_QTY2_REQUIRED_DUAL3
Trans_um is required . Item_no : &ITEM_NO, Line_id : &LINE_ID	GMI_API_UOM_REQUIRED
UOM conversion not found from &FROM_UOM to &TO_UOM. Item_no : &ITEM_NO, Line_id : &LINE_ID	GMI_API_UOM_CONV_NOT_FOUND
Warehouse &WAREH is closed for date &DATE. The transaction is not allowed.	IC_API_TXN_POST_CLOSED
Warehouse code not found for organization &ORG, Line_id &LINE_ID.	GMI_API_WHSE_NOT_FOUND
WARNING : This Allocation would drive the Inventory -ve. Line_id : &LINE_ID	GMI_API_WARN_INVQTYNEG
You must supply a warehouse.	IC_BLANKWHSE

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