

Oracle® Process Manufacturing

Product Development Formula API User's Guide

Release 11*i*

Part No. A97652-04

August 2004

Oracle Process Manufacturing Product Development Formula API User's Guide, Release 11i

Part No. A97652-04

Copyright © 2002, 2004, Oracle. All rights reserved.

Primary Author: Michele-Andrea Fields

Contributors: Shyamsunder Sitaraman, Bill Stearns

The Programs (which include both the software and documentation) contain proprietary information; 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. This document is not warranted to be 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.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth 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 we disclaim liability for any damages caused by such use of the Programs.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

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

Contents

Send Us Your Comments	v
Preface.....	vii
1 Formula API Introduction	
Introducing the Formula API	1-2
Formula API Features	1-3
Formula API Support Policy.....	1-3
Oracle Applications Packages Supplied	1-4
Formula API Bill of Materials	1-4
Wrapper Function and Stored Procedures	1-6
Validating the User Name	1-6
2 Formula API Business Objects	
Using Formula API Business Objects.....	2-2
Creating a New Formula.....	2-3
Importing Formula Data Structures.....	2-5
Understanding the General Input Structure.....	2-5
Using Input Parameters for Each Formula API	2-6
Change Status API.....	2-20
A Messages and Errors	
Handling Messages	2
Interpreting Error Conditions	3

Understanding Error Messages 3

B How to Get Your OPM Formula APIs Running

Send Us Your Comments

**Oracle Process Manufacturing Product Development Formula API User's Guide, Release 11i
Part No. A97652-04**

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

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

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

- Electronic mail: appsdoc_us@oracle.com
- FAX: (650) 506-7200 Attn: Oracle Applications Documentation Manager
- Postal service:
Oracle Corporation
Oracle Applications Documentation Manager
500 Oracle Parkway
Redwood Shores, CA 94065
USA

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

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

Preface

Welcome to the Oracle Process Manufacturing Product Development Formula Management API User's Guide, Release 11i.

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

- The principles and customary practices of your business area.
- Oracle Process Manufacturing.

If you have never used Oracle Process Manufacturing, Oracle suggests you attend one or more of the Oracle Applications training classes available through Oracle University.

- Oracle Self-Service Web Applications.

To learn more about Oracle Self-Service Web Applications, read the *Oracle Self-Service Web Applications Implementation Manual*.

- 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

The Oracle Process Manufacturing Product Development Formula Management API User's Guide contains the information you need to understand and use Oracle Process Manufacturing. This guide contains four chapters:

- Chapter 1 describes the Application Program Interfaces (APIs) that support external interfaces to the OPM Product Development Formula Management tables including: packages supplied, Formula API bill of material, wrapper function, stored procedures, and validating the user name.
- Chapter 2 provides the relationships between Formula API table structure and its entities. Discusses Formula API business objects, the entity relationship diagram, business object interface design, creating a new formula and importing formula data structures.
- Appendix A provides message handling, the interpretation of error conditions, and an understanding of error messages.
- Appendix B provides a useful guide and examples for using the APIs.

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 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.

Accessibility of Links to External Web Sites in Documentation

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

Other Information Sources

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

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).

- **PDF Documentation**- See the Online Documentation CD for current PDF documentation for your product with each release. This Documentation CD is also available on Oracle*MetaLink* and is updated frequently.
- **Online Help** - You can refer to Oracle Applications Help for current HTML online help for your product. Oracle provides patchable online help, which you can apply to your system for updated implementation and end user documentation. No system downtime is required to apply online help.
- **Release Content Document** - See the Release Content Document for descriptions of new features available by release. The Release Content Document is available on Oracle*MetaLink*.
- **About document** - Refer to the About document for information about your release, including feature updates, installation information, and new documentation or documentation patches that you can download. The About document is available on Oracle*MetaLink*.

Related Guides

Oracle Process Manufacturing shares business and setup information with other Oracle Applications products. Therefore, you may want to refer to other guides when you set up and use Oracle Process Manufacturing.

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). 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.

Guides Related to This Product

Accounting Setup User's Guide

The OPM Accounting Setup application is where users set up global accounting attributes about the way financial data will be collected by OPM. These attributes include such things as account keys, financial calendars, and account segments. Since OPM is closely integrated with Oracle General Ledger (GL), much of the attributes are defined in the Oracle GL instead of OPM, and therefore, the windows are display only within OPM. The *Oracle Process Manufacturing Accounting Setup User's Guide* describes how to setup and use this application.

Cost Management User's Guide

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

Manufacturing Accounting Controller User's Guide

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

Oracle Financials Integration User's Guide

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

Inventory Management User's Guide

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

Physical Inventory User's Guide

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

Order Fulfillment User's Guide

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

Purchase Management User's Guide

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

Using Oracle Order Management with Process Inventory Guide

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

Process Execution User's Guide

The OPM Process Execution application lets you track firm planned orders and production batches from incoming materials through finished goods. Seamlessly integrated to the Product Development application, Process Execution lets you convert firm planned orders to single or multiple production batches, allocate ingredients, record actual ingredient usage, and then complete and close production batches. Production inquiries and preformatted reports help you optimize inventory costs while maintaining a high level of customer satisfaction with on-time delivery of high quality products. The *OPM Process Execution User's Guide* presents overviews of the tasks and responsibilities for the Production Supervisor and the Production Operator. It provides prerequisite setup in other applications, and details the windows, features, and functionality of the OPM Process Execution application.

Using Oracle Advanced Planning and Scheduling with Oracle Process Manufacturing

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

MPS/MRP and Forecasting User's Guide

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

Capacity Planning User's Guide

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

Using Oracle Process Manufacturing with Oracle Manufacturing Scheduling

Oracle Process Manufacturing integrates with Oracle Manufacturing Scheduling to manage and utilize resources and materials. Through the Process Manufacturing application, you set up manufacturing, inventory, procurement and sales order data. Through the Manufacturing Scheduling application, you can optimize the schedule based on resource and component constraints and user predefined priorities. Using different optimization objectives, you can tailor Manufacturing Scheduling to meet your needs.

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

Product Development User's Guide

The Oracle Process Manufacturing Product Development application provides features to manage formula and laboratory work within the process manufacturing operation. It lets you manage multiple laboratory organizations and support varying product lines throughout the organization. You can characterize and simulate the technical properties of ingredients and their effects on formulas. You can optimize formulations before beginning expensive laboratory test batches. Product Development coordinates each development function and enables a rapid,

enterprise-wide implementation of new products in your plants. The *Oracle Process Manufacturing Product Development User's Guide* describes how to setup and use this application.

Quality Management User's Guide

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

Implementation Guide

The *Oracle Process Manufacturing Implementation Guide* offers information on setup. That is, those tasks you must complete following the initial installation of the Oracle Process Manufacturing software. Any tasks that must be completed in order to use the system out-of-the-box are included in this manual.

System Administration User's Guide

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

API User's Guides

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

“About” Document

For information about implementation and user documentation, instructions for applying patches, new and changed setup steps, and descriptions of software updates, refer to the “About” document for your product. “About” documents are available on Oracle *MetaLink* for most products starting with Release 11.5.8.

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 and describes the Oracle Application Object Library components that are needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. This manual also provides information to help you build your custom Oracle Forms Developer forms so that the forms 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.

Oracle Workflow Administrator's Guide

This guide explains how to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes, as well as how to monitor the progress of runtime workflow processes.

Oracle Workflow Developer's Guide

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

Oracle Workflow User's Guide

This guide describes how Oracle Applications users can view and respond to workflow notifications and monitor the progress of their workflow processes.

Oracle Workflow API Reference

This guide describes the APIs provided for developers and administrators to access Oracle Workflow.

Oracle Applications Flexfields Guide

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

Oracle eTechnical Reference Manuals

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

Oracle Applications Message Manual

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

Training and Support

Training

Oracle offers a complete set of training courses to help you and your staff master Oracle Process Manufacturing 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 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 Oracle 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 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 and this user guide.

Oracle values your comments and feedback. In this guide is a reader's comment form that you can use to explain what you like or dislike about Oracle Process Manufacturing or this user 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.

Formula API Introduction

This document describes the Application Program Interfaces (APIs) that support external interfaces to the Oracle Process Manufacturing (OPM) Formula Management tables. The topics discussed in this topic are:

- Introducing the Formula API
- Oracle Applications Packages Supplied
- Formula API Bill of Materials
- Wrapper Function and Stored Procedures
- Validating the User Name

Introducing the Formula API

Formula APIs let you import formula information from an existing formula management system into the Oracle Process Manufacturing Formula Management tables. When you import formulas, you can now include all pertinent information using a friendly tool that does not have cryptic IDs and system specific information. Formula APIs can process virtually all formula types. The interface ensures that your imported formulas contain the same detail as those you enter manually on the OPM Formulas window. However, every formula must include at least one product and one ingredient.

What Is In This Document

This document describes the basic business needs, major features, architecture, and components for the Insert, Update, and Delete features for the Formula APIs. 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 Formula Management application.

Formula APIs are written in PL/SQL that can be called by your own programs. To make use of these APIs, you code the wrapper function that passes the appropriate parameters to the APIs. Your program is also responsible for connecting to a database before calling an API function, and disconnecting from the database upon return. You can also write to 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
- Q for could not calculate the total input and output quantities

Included API Functions

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

Function	Business Object Name	Description
Formula Create	GMD_FORMULA_XXX	Creates formula header, detail, and effectivity information. It can be used for modification of the formula header information.
Formula Detail	GMD_FORMULA_DETAIL_XXX	Creates or modifies the formula detail information.

Where xxx in the business object name is PVT (private) or PUB (public).

Formula API Features

These APIs offer the following features:

- Creating Updating and Deleting Information
- Proper Encapsulation
- Synchronous Processing Following the Business Hierarchy
- Detailed and Translatable Error Messages

Formula API Support Policy

Formula APIs are supported by Oracle. This means:

- Oracle provides objects and libraries needed to use to 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.

Oracle Applications Packages Supplied

Formula 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 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 interrogate the procedure messages.

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

Formula API Bill of Materials

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

Package Name	File Names	Description
GMD_FORMULA_PUB	GMDPFMHS.pls GMDPFMHB.pls	Public Formula Header package that the user defined function calls. The business API can be used for creating, modifying, or deleting a formula header. While creating a Formula header, the API also creates Detail and Effectivity associated with this header.
GMD_FORMULA_DETAIL_PUB	GMDPFMDS.pls GMDPFMDB.pls	Public Formula Detail package that the wrapper or user defined function calls. The business API can be used for creating, modifying, or deleting a formula detail.
GMD_FORMULA_PVT	GMDVFMHS.pls GMDVFMHB.pls	Private Formula Header package that cannot be called directly by the user defined function. The private API is called by public formula header API after performing the necessary validations.

Package Name	File Names	Description
GMD_FORMULA_DETAIL_PVT	GMDVFMDS.pls GMDVFMDB.pls	Private Formula Detail package that cannot be called directly by the user defined function. The private API is called by public formula detail API after performing the necessary validations.
Public Package for Surrogates	GMDPSUMRS.pls GMDPSURB.pls	Public Formula Surrogate package used primarily for creating surrogate keys for formula header, detail lines, and effectivity.
Public Validation Packages	GMDPFMVB.pls GMDPFMVS.pls	Public Formula Validation package used by other procedures in the APIs to perform all business rules validations. For example, prior to the formula line insertion the procedure 'Detail_line_val' (included in this package) is called to check if the same line exists in the OPM system. It also uses lookup tables for validations.
GMD_FORMULA_COMMON_PKG	GMDPFMCS.pls	Common formula package that defines common data type structure (PL/SQL table and record type) for inserts or updates.
GMD_STATUS_PUB	GMDPSTSB.pls, GMDPSTSS.pls	Public API that modifies the status for routings, recipes, operations, and validity rules.

Wrapper Function and Stored Procedures

A wrapper function is used for collating the formulated data in a specific structure (PL/SQL table). This wrapper function calls the Formula API by passing this PL/SQL table as a parameter. Depending on the action (creation/inserts or modification/updates) and the entity (header/detail/effectivity) these wrapper functions can call the public procedures defined in the APIs. For example, to insert all Formula details you can call the procedure 'Insert_FormulaDetail' defined in the public package GMD_FORMULA_DETAIL_PUB. The stored procedures return one of these the status codes:

- S for success
- E for error
- U for unknown or unexpected status
- Q for could not calculate the total input and output quantities

Depending on the status returned, decide if the operation needs to be committed. Optionally, the defined calling function makes the API commit after performing the operation.

Validating the User Name

The Formula API validates the user name that has been passed to the API against the `fnr_user` table. If the user you enter does not have a valid ID, then the API returns an error without performing any other functions or validations.

Formula API Business Objects

In order to use Formula API business objects it is important to understand the relationships of the table structure with all its entities. You must also understand the fundamental API business object interface design before you import formula data structures and manage any errors that arise during these operations.

The following topics are discussed:

- Using Formula API Business Objects
- Creating a New Formula
- Importing Formula Data Structures

Using Formula API Business Objects

A formula is a method to prescribe ingredients and quantities that are required to manufacture particular products. Formulas occasionally contain by-products and coproducts. The information included in a formula is used in various other Oracle Manufacturing modules, including Supply Chain Planning (Material Requirement Planning and Advanced Planning and Scheduling), Production Management, and Cost Management.

Formula APIs are business objects that can create or change the formula information. Formula information within the OPM application is comprised of:

- A header that provides the formula number, version, description, and status or validity of a formula.
- One or more detail lines for each header. Each detail line provides the product, coproduct, ingredient, byproduct specification for a formula.

Creating a New Formula

In order to use the formula business object APIs effectively, follow these steps:

Step 1: Collate formula data into a PL/SQL table format.

Formula APIs can be called through different types of sources. For example, you can use an interface like Oracle forms, or optionally create a wrapper function that calls these APIs. However, ensure that relevant formula information is structured in a PL/SQL table format *before* passing it as a parameter to these APIs.

A test function, `Create_formula`, provided with the APIs, describes a typical wrapper function that creates a PL/SQL table from a comma-delimited text file. The function parses through this flat file and creates the table structure that is passed as parameter to the `Insert_Formula` procedure within the API `GMD_FORMULA_PUB`. Since the test function is a representation of a typical wrapper function, the actual wrapper function can be designed differently.

The APIs also require certain standard parameters such as the API version that needs to be passed by the wrapper function. After performing the appropriate tasks, the APIs return the status code. Depending on this status code, the calling function may decide to commit the work. If the return code is an error, then the function retrieves the error message text from the error stack.

Step 2: Call the public API - main validation performed.

- a. The API checks for the existence of an appropriate User ID in the `FND_USR` table. If there is no valid user, then the API puts error messages in an error stack, and prevents the creation of a formula.
- b. The API checks for valid formula number and version. For all updates or changes to the Formula Header information, Formula ID, or Formula Number, and Formula Version need to be provided. For changes to Formula Detail, the Formula Line ID information needs to be provided.
- c. The API checks every Formula Header to determine that there is at least one product and one ingredient associated with it.
- d. After the formula header is created successfully, formula details are created. If a header has more than one line, then the API creates a header only once (for the first line), and creates all Formula Details as separate lines.
- e. No changes or updates are performed for any formulas that have cost rollups done on them.

Step 3: Review any error messages.

The API returns the status code as one of its parameters after it is executed. The status code represents S for success, E for error, or U Unexpected or Unknown status. If an error E has occurred, then the calling function or wrapper function analyzes the errors from the message stack.

The API can also return a status of Q. Status Q is when the API can not calculate the total input and output quantity for a formula. This occurs when a product or ingredient unit of measure does not convert to the GMD:Yield Type UOM profile option value. Status Q is not an error condition. Formulas can be saved with this status.

For more information on error messages, refer to Appendix A.

Step 4: Check if relevant formula information has been created.

After executing these APIs successfully, and committing the work, the formula information must be tested using a different user interface. For example, after the formula (header and detail) is created, commit the work, and run the appropriate Oracle application to test whether or not a new formula has been created. Check to determine if at least one product and one ingredient is associated with the newly created formula.

Importing Formula Data Structures

The following topic demonstrates the mechanics of importing data structures for:

- Formula Header
- Formula Detail

A single general input structure is described in the following topic.

Understanding the General Input Structure

All stored procedure API are called with PL/SQL parameters. Examination of both the `x_return_status` and `x_msg_count` indicates the pass or fail status of the call. Standard parameters that are common to all API activities and their details are summarized in the following:

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>varchar2</code>	IN	N	Used to specify whether the message list must be initialized on entry to the API. It is an optional parameter, and if not supplied, then it defaults to <code>FND_API.G_FALSE</code> which means that the API does not initialize the message list.
<code>p_commit</code>	<code>varchar2</code>	IN	N	Used to specify whether the API must commit its work before returning to the calling function. If not supplied, then it defaults to <code>FND_API.G_FALSE</code> .
<code>p_called_from_forms</code>	<code>varchar2</code>	IN	N	The default value is No.
<code>x_return_status</code>	<code>varchar2</code>	OUT	N	Specifies whether the API was successful or failed: S = Successful E = failed due to expected error U = failed due to unexpected error Q = Could not calculate the total input and output quantities
<code>x_msg_count</code>	<code>number</code>	OUT	N	Specifies number of messages added to message list.

Parameter	Type	IN/OUT	Required	Validation
x_msg_data	varchar2	OUT	N	Returns the messages in an encoded format. These messages can then be processed by the standard message functions as defined in business object API Coding Standards Document.

Using Input Parameters for Each Formula API

The Formula API requires data to pass in as PL/SQL tables. The wrapper functions create these tables by referencing the table types in the API specifications. Although the table types are similar, the default values set for these table types are different depending on the DML operation.

If the wrapper function needs to insert or create new formula header detail, then it needs to create a table type equal to:

`p_formula_insert_tbl_type`

The specification for this is provided in this topic.

If the wrapper function needs to update or delete a formula header detail, then it needs to create a table type equal to:

`p_formula_update_tbl_type`

The table type provides default values for all optional parameters.

These parameters pass the item-specific data required to create an inventory item. However, parameters can have formula data related to formula detail and effectivity. Since the header data can be in multiple instances it is passed as a PL/SQL table.

If any column must be updated to NULL, then you must pass in `FND_API.G_MISS_CHAR`, `FND_API.G_MISS_NUM`, or `FND_API.G_MISS_DATE` variables to the API column values to update the column to NULL in the database.

Additionally, while inserting or creating new formula headers, you can pass an optional parameter, `p_allow_zero_ing_qty`, with character(10) and Default (False). This parameter lets you create a formula where the total ingredient quantity equals 0.

Parameter Specification for p_formula_insert_tbl_type

The parameter specification for p_formula_insert_tbl_type is described below. This specification is used for inserts. No validation is applied to the descriptive flexfield segments.

Parameters	Type	Length	Required	Validation
RECORD_TYPE	VARCHAR2	1	Y	Action code set to indicate if the records are a insert or update mode. If no value is passed, then it defaults to I.
FORMULA_NO	VARCHAR2	32	Y	Must be non space.
FORMULA_VERS	NUMBER	4	Y	Must be greater than or equal to 0. The formula_no and formula_vers combination must not exist on fm_form_mst. Duplicates are not allowed.
FORMULA_TYPE	NUMBER	5	Y	Valid values are 0 - standard formula, 1 - non-standard formula. If no value is passed, then it defaults to 0.
FORMULA_DESC1	VARCHAR2	70	Y	Description of the formula.
FORMULA_DESC2	VARCHAR2	70	N	Description of the formula.
FORMULA_CLASS	VARCHAR2	32	N	Must exist on fm_form_cls with delete_mark=0.
FMCONTROL_CLASS	VARCHAR2	32	N	Not currently used; must be null.
INACTIVE_IND	NUMBER	5	Y	Valid values are 0 = active (default), 1 = inactive. If no value is passed, then it defaults to 0.
ORGN_CODE	VARCHAR2	4	Y	Organization code for the creator of the formula. If no value is passed, then the default orgn_code set in profile options, for the user ID, is passed.
TOTAL_INPUT_QTY	NUMBER	Variable	N	The total ingredient quantity calculated in GMD:Yield Type UOM.
TOTAL_OUTPUT_QTY	NUMBER	Variable	N	The total product/byproduct quantity calculated in GMD:Yield Type UOM.
FORMULA_UOM	VARCHAR2	4	N	UOM expressed in GMD:Yield Type UOM.

Parameters	Type	Length	Required	Validation
BY_PRODUCT_TYPE	VARCHAR2	1	N	Used for Quality Management integration for samples.
FORMULA_STATUS	VARCHAR2	30	Y	Based on the status_code values in GMD_STATUS table. If no value is passed, then it defaults to 100, defined as New Status..
OWNER_ID	NUMBER	15	Y	User ID of the formula owner. If no value is passed, then it defaults to user_id.
FORMULA_ID	NUMBER	10	N	Formula ID.
FORMULALINE_ID	NUMBER	10	N	Formulaline ID.
LINE_TYPE	NUMBER	5	Y	Valid values are -1 = ingredient, 1 = product, 2 = byproduct. Line type of 1 (product) must be in place against a particular formula before line types of -1 and 2 can be accepted.
LINE_NO	NUMBER	5	Y	Must be greater than 0. Duplicates are not allowed within line_type.
ITEM_NO	VARCHAR2	32	Y	Must exist in ic_item_mst with experimental_ind=0, inactive_ind=0, delete_mark=0.
QTY	NUMBER	Variable	Y	Must be greater than or equal to 0.
ITEM_UM	VARCHAR2	4	Y	Must exist on sy_uoms_mst. Conversion to ic_item_mst.item_um (primary UOM) must be defined.
RELEASE_TYPE	NUMBER	5	Y	Valid values are 0 = automatic release, 1 = partial release. If no value is passed, then it defaults to 0.
SCRAP_FACTOR	NUMBER	Variable	Y	Must be greater than or equal to 0, and less than or equal to 10,000. Default value is 0. If no value is passed, then it defaults to 0.
SCALE_TYPE_HDR	NUMBER	5	Y	Valid values are 0 = formula cannot be scaled, 1 = formula can be scaled. If no value is passed, then it defaults to 1.
SCALE_TYPE_DTL	NUMBER	5	Y	Valid values are 0 = fixed, 1 = linear scaling, 2 = step scaling. If no value is passed, then it defaults to 1.

Parameters	Type	Length	Required	Validation
COST_ALLOC	NUMBER	Variable	N	Must be in the range 0 to 1, inclusive.
PHANTOM_TYPE	NUMBER	5	Y	Valid values are 0 = not a phantom (default), 1 = automatic phantom replacement, 2 = manual phantom. If no value is passed, then it defaults to 0.
REWORK_TYPE	NUMBER	5	Y	Valid values are 0 = no rework (default), 1 = default quantity as stated. If no value is passed, then it defaults to 0.
ATTRIBUTE1	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE2	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE3	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE4	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE5	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE6	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE7	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE8	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE9	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE10	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE11	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE12	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE13	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE14	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE15	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.

Parameters	Type	Length	Required	Validation
ATTRIBUTE16	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE17	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE18	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE19	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE20	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE21	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE22	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE23	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE24	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE25	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE26	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE27	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE28	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE29	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE30	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
DTL_ATTRIBUTE1	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE2	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE3	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE4	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE5	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.

Parameters	Type	Length	Required	Validation
DTL_ATTRIBUTE6	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE7	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE8	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE9	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE10	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE11	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE12	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE13	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE14	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE15	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE16	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE17	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE18	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE19	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE20	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE21	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE22	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE23	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE24	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE25	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.

Parameters	Type	Length	Required	Validation
DTL_ATTRIBUTE26	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE27	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE28	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE29	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE30	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
ATTRIBUTE_CATEGORY	VARCHAR2	30	N	Descriptive Flexfield Attribute Category for Formula Header.
DTL_ATTRIBUTE_CATEGORY	VARCHAR2	30	N	Descriptive Flexfield Attribute Category for Formula Detail.
TPFORMULA_ID	NUMBER	10	N	Technical parameter formula ID.
IAFORMULA_ID	NUMBER	10	N	Formula analysis formula ID.
SCALE_MULTIPLE	NUMBER	Variable	N	Field associated with integer scale. The scaled quantity must be multiples of the value of this field.
CONTRIBUTE_YIELD_IND	VARCHAR2	1	N	Indicates if the ingredient/product contributes to the formula yield. If no value is passed, then it defaults to Y.
SCALE_UOM	VARCHAR2	4	N	Scale unit of measure.
CONTRIBUTE_STEP_QTY_IND	VARCHAR2	1	N	Indicates if the ingredient/product contributes to the routing/recipe step quantity. If no value is passed, then it defaults to Y.
SCALE_ROUNDING_VARIANCE	NUMBER	Variable	N	Field associated with integer scale. The value is shown as a percentage of the total quantity.
ROUNDING_DIRECTION	NUMBER	Variable	N	Indicates the acceptable direction of rounding for the scale routing variance. Valid values are UP, DOWN, or EITHER.
TEXT_CODE_HDR	NUMBER	10	N	Text code for the formula header.
TEXT_CODE_DTL	NUMBER	10	N	Text code for the formula detail.
USER_ID	NUMBER	32	N	ID for the creator of the formula.
CREATION_DATE	DATE	Date	Y	Standard Who Column. If no value is passed, then it defaults to SYSDATE.

Parameters	Type	Length	Required	Validation
CREATED_BY	NUMBER	15	Y	Standard Who Column. If no value is passed, then it defaults to USER_ID.
LAST_UPDATED_BY	NUMBER	15	Y	Standard Who Column. If no value is passed, then it defaults to USER_ID.
LAST_UPDATE_DATE	DATE	Date	Y	Standard Who Column. If no value is passed, then it defaults to SYSDATE.
LAST_UPDATE_LOGIN	NUMBER	15	N	Standard Who Column.
USER_NAME	VARCHAR2	100	Y	Must exist on fnd_user.
DELETE_MARK	NUMBER	5	Y	Logical flag to indicate that the formula header has been marked for purge. If no value is passed, then it defaults to 0.

Parameter Specification for p_formula_update_tbl_type

The parameter specification for p_formula_update_tbl_type is described below. This specification is used for updates. No validation is applied to descriptive flexfield segments. If a column value is not passed, then it is populated with a value from the database.

Parameter	Type	Length	Required	Validation
RECORD_TYPE	VARCHAR2	1	Y	Action code set to indicate if the records are an insert or update mode. If no value is passed, then it defaults to U.
FORMULA_NO	VARCHAR2	32	Y	Must be non space.
FORMULA_VERS	NUMBER	4	Y	Must be greater than or equal to 0. The formula_no and formula_vers combination must not exist on fm_form_mst. Duplicates are not allowed.
FORMULA_TYPE	NUMBER	5	N	Default value is 0 - standard formula.
FORMULA_DESC1	VARCHAR2	70	N	Description of the formula.
FORMULA_DESC2	VARCHAR2	70	N	Description of the formula.
FORMULA_CLASS	VARCHAR2	32	N	Must exist on fm_form_cls with delete_mark=0.
FMCONTROL_CLASS	VARCHAR2	32	N	Not currently used; must be null.
INACTIVE_IND	NUMBER	5	N	Valid values are 0 = active (default), 1 = inactive
ORGN_CODE	VARCHAR2	4	N	Organization code for the creator of the formula.
TOTAL_INPUT_QTY	NUMBER	Variable	N	The total ingredient quantity calculated in GMD:Yield Type UOM.
BY_PRODUCT_TYPE	VARCHAR2	1	N	Used for Quality Management integration for samples.
TOTAL_OUTPUT_QTY	NUMBER	Variable	N	The total product/byproduct quantity calculated in GMD:Yield Type UOM.
FORMULA_UOM	VARCHAR2	4	N	UOM expressed in GMD:Yield Type UOM.
FORMULA_STATUS	VARCHAR2	30	N	Based on the status_code values in GMD_STATUS table.

Parameter	Type	Length	Required	Validation
OWNER_ID	NUMBER	15	N	User ID of the formula owner.
FORMULA_ID	NUMBER	10	N	Formula ID.
FORMULALINE_ID	NUMBER	10	N	Formulaline ID.
LINE_TYPE	NUMBER	5	N	Valid values are -1 = ingredient, 1 = product, 2 = byproduct. Line-type of 1 (product) must be in place against a particular formula before line types of -1 and 2 can be accepted.
LINE_NO	NUMBER	5	N	Must be greater than 0. Duplicates are not allowed within line_type.
ITEM_NO	VARCHAR2	32	N	Must exist in ic_item_mst with experimental_ind=0, inactive_ind=0, delete_mark=0.
QTY	NUMBER	Variable	N	Must be greater than or equal to 0.
ITEM_UM	VARCHAR2	4	N	Must exist on sy_uoms_mst. Conversion to ic_item_mst.item_um (primary UOM) must be defined.
RELEASE_TYPE	NUMBER	5	N	Valid values are 0 = automatic release, 1 = partial release.
SCRAP_FACTOR	NUMBER	Variable	N	Must be greater than or equal to 0, and less than or equal to 10,000. Default value is 0.
SCALE_TYPE_HDR	NUMBER	5	N	Valid values are 0 = formula cannot be scaled, 1 = formula can be scaled.
SCALE_TYPE_DTL	NUMBER	5	N	Valid values are 0 = fixed, 1 = linear scaling, 2 = step scaling.
COST_ALLOC	NUMBER	Variable	N	Must be in the range 0 to 1, inclusive.
PHANTOM_TYPE	NUMBER	5	N	Valid values are 0 = not a phantom (default), 1 = automatic phantom replacement, 2 = manual phantom.
REWORK_TYPE	NUMBER	5	N	Valid values are 0 = no rework (default), 1 = default quantity as stated.
ATTRIBUTE1	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE2	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.

Parameter	Type	Length	Required	Validation
ATTRIBUTE3	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE4	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE5	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE6	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE7	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE8	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE9	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE10	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE11	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE12	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE13	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE14	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE15	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE16	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE17	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE18	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE19	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE20	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE21	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE22	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.

Parameter	Type	Length	Required	Validation
ATTRIBUTE23	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE24	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE25	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE26	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE27	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE28	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE29	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
ATTRIBUTE30	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Header.
DTL_ATTRIBUTE1	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE2	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE3	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE4	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE5	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE6	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE7	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE8	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE9	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE10	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE11	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE12	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.

Parameter	Type	Length	Required	Validation
DTL_ATTRIBUTE13	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE14	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE15	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE16	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE17	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE18	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE19	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE20	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE21	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE22	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE23	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE24	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE25	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE26	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE27	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE28	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE29	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
DTL_ATTRIBUTE30	VARCHAR2	240	N	Descriptive Flexfield Segment for Formula Detail.
ATTRIBUTE_CATEGORY	VARCHAR2	30	N	Descriptive Flexfield Attribute Category for Formula Header.
DTL_ATTRIBUTE_CATEGORY	VARCHAR2	30	N	Descriptive Flexfield Attribute Category for Formula Detail.

Parameter	Type	Length	Required	Validation
TPFORMULA_ID	NUMBER	10	N	Technical parameter formula ID.
IAFORMULA_ID	NUMBER	10	N	Formula analysis formula ID.
SCALE_MULTIPLE	NUMBER	Variable	N	Field associated with integer scale. The scaled quantity must be multiples of the value of this field.
CONTRIBUTE_YIELD_IND	VARCHAR2	1	N	Indicates if the ingredient/product contributes to the formula yield.
SCALE_UOM	VARCHAR2	4	N	Scale unit of measure.
CONTRIBUTE_STEP_QTY_IND	VARCHAR2	1	N	Indicates if the ingredient/product contributes to the routing/recipe step quantity.
SCALE_ROUNDING_VARIANCE	NUMBER	Variable	N	Field associated with integer scale. The value is shown as a percentage of the total quantity.
ROUNDING_DIRECTION	NUMBER	Variable	N	Indicates the acceptable direction of rounding for the scale routing variance. Valid values are UP, DOWN, or EITHER.
TEXT_CODE_HDR	NUMBER	10	N	Text code for the formula header.
TEXT_CODE_DTL	NUMBER	10	N	Text code for the formula detail.
USER_ID	NUMBER	32	N	ID for the creator of the formula.
CREATION_DATE	DATE	Date	N	Standard Who Column.
CREATED_BY	NUMBER	15	N	Standard Who Column.
LAST_UPDATED_BY	NUMBER	15	N	Standard Who Column.
LAST_UPDATE_DATE	DATE	Date	N	Standard Who Column.
LAST_UPDATE_LOGIN	NUMBER	15	N	Standard Who Column.
USER_NAME	VARCHAR2	100	Y	Must exist on fnd_user.
DELETE_MARK	NUMBER	5	N	Logical flag to indicate that the formula header has been marked for purge.

Change Status API

When a formula is created, it has a status of NEW. You can change the status on these objects using the Change Status API. The Update APIs do not let you change the status of an object.

This API must not be called from within another API. Rather, any updates to the other attributes must be done first. Then the Change Status API can be used.

```
GMD_STATUS_PUB.modify_status
( p_api_version IN NUMBER=> 1
, p_init_msg_list IN BOOLEAN => TRUE
, p_entity_name IN VARCHAR2
, p_entity_id IN NUMBER
, p_entity_no IN VARCHAR2
, p_entity_version IN NUMBER
, p_status_to IN VARCHAR2
, p_ignore_flag IN BOOLEAN => FALSE
, x_message_count OUT NUMBER
, x_message_list OUT VARCHAR2
, x_return_status OUT VARCHAR2
);
```

General Input Structure

Field/Column	In/ Out	Type	Required	Description
p_entity_name	IN	Varchar2	Yes	The object for which the status change occurs. Following are the valid names that can be passed in as a parameter (mixed case is allowed): Recipe, Formula, Routing, Operation, Validity.
p_entity_id	IN	Number	Yes/ Optional if a valid Entity Name and version are provided	Depends on the entity that is being updated. For example, if the Entity Name is equal to Recipe, then this represents the recipe_id.
p_entity_no	IN	Varchar2	Yes/ Optional if a valid Entity Id is provided.	Depends on the entity that is being updated. For example, if the Entity Name is equal to Recipe, then this represents the recipe_no.

Field/Column	In/ Out	Type	Required	Description
p_entity_version	IN	Number	Yes/ Optional if a valid Entity Id is provided.	Depends on the entity that is being updated. For e.g. for Entity Name = 'Recipe' this would represent the recipe_version.
p_status_to	IN	Varchar2	Y	The status to be changed to.
p_ignore_flag	IN	Boolean	No. Default value = FALSE	This flag defaults to FALSE. If it is changed to TRUE, then it indicates that when a recipe status is changed to either Obsolete or On Hold and if its associated Validity Rules (VR) status are not either Obsolete or On Hold, the VRs are changed to Obsolete or On Hold. If it is FALSE, then it errors out and provide a message stating that there are VRs associated for this recipe and the status cannot be updated.

A

Messages and Errors

This appendix covers the following topics:

- Handling Messages
- Interpreting Error Conditions
- Understanding Error Messages

Handling Messages

APIs put result messages into a message list. Programs calling APIs can then get the messages from this list, and process them. This can be by issuing them to the user if calling the API from an interactive process, or writing them to database tables or log files if calling the API from a batch process.

Messages are stored in an encoded format to let API callers find out message names using the standard functions provided by the message dictionary.

The structure of the message list is not public. Neither API developers nor API callers can access this list except by calling the API message utility routines.

The following utility functions are defined in the FND_MSG_PUB package, in the file AFASMSG.S.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.

Refer to the *Oracle Applications Coding Standards* guide for complete documentation of these functions and procedures for usage information.

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

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

Interpreting Error Conditions

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

- S for success
- E for error
- U for unknown or unexpected status
- Q for could not calculate the total input and output quantities

Understanding Error Messages

Error messages are output to the stored procedure message file, and can be monitored through the return `x_msg_count`. 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 the Formula API error messages:

Error Messages	Message Code
Duplicate line number for item <code>&ITEM_NO</code> in formula/version <code>&FORMULA_NO</code> &FORMULA_VERS	FM_DUPLICATE_LINE_NO
Experimental item <code>&ITEM_NO</code> in formula/version <code>&FORMULA_NO</code> &FORMULA_VERS	FM_EXPERIMENTAL_ITEM
Formula number/version <code>&FORMULA_NO</code> &FORMULA_VERS already exists	FM_FORMULA_ALREADY_EXISTS
Formula number/version <code>&FORMULA_NO</code> &FORMULA_VERS cannot be located	FM_INVALID_FORMULA
Invalid formula class for formula/version <code>&FORMULA_NO</code> &FORMULA_VERS	FM_INVALID_FORMULA_CLASS

Error Messages	Message Code
Invalid formula type for formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_FORMULA_TYPE
Invalid item &ITEM_NO in formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_ITEM
Invalid scale type for formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_SCALE_TYPE
Missing formula number	FM_MISSING_FORMULA_NO
Missing formula version	FM_MISSING_FORMULA_VERS
Ingredient/by-product lines cannot be loaded until product is established for formula/version &FORMULA_NO &FORMULA_VERS	FM_MISSING_PRODUCT
Negative quantity not permitted see item &ITEM_NO in formula/version &FORMULA_ NO &FORMULA_VERS	FM_NEGATIVE_QTY
Invalid cost allocation for item &ITEM_NO in formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_COST_ALLOC
Invalid customer &CUST_NO	FM_INVALID_CUSTOMER
Invalid end_date against effectivity for formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_END_DATE
Invalid product &ITEM_NO for formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_FORMULA_PRODUCT
Invalid formula_use for formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_FORMULA_USE
Invalid item number &ITEM_NO	FM_INVALID_ITEM_NO
Invalid line number for item &ITEM_NO in formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_LINE_NO
Invalid line type for item &ITEM_NO in formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_LINE_TYPE
Invalid phantom type for item &ITEM_NO in formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_PHANTOM_TYPE

Error Messages	Message Code
Invalid release type for item &ITEM_NO in formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_RELEASE_TYPE
Invalid rework type for item &ITEM_NO in formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_REWORK_TYPE
Invalid scrap factor for item &ITEM_NO in formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_SCRAP_FACTOR
Invalid start_date against effectivity for formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_START_DATE
Invalid UOM &ITEM_UM against effectivity for formula/version &FORMULA_NO &FORMULA_VERS	FM_INVALID_UOM
Invalid user name &USER_NAME	FM_INVALID_USER
No conversion to standard UOM for item &ITEM_NO	FM_MISSING_UOM_CONVERSION
No conversion defined from &ITEM_UM to primary UOM for item &ITEM_NO in formula/version &FORMULA_NO &FORMULA_VERS	FM_UOM_CONVERSION
Operation cannot be approved for Lab use or General use. Please attach resource(s) for the following activity(s): &ACTIVITY	GMD_ATTACH_RESOURCES
This formula is used in one or more recipes. Status of this formula cannot be changed to obsolete or on-Hold	GMD_FORMULA_INUSE
The target status &TO_STATUS is invalid for the current status.	GMD_INV_TARGET_STATUS
Field Validation: Missing &MISSING (Id = &ID, No = &NO, Version = &VERS)	GMD_MISSING
This operation is used in one or more routings or batch steps. Status of this operation cannot be changed to obsolete or on-Hold	GMD_OPERATION_INUSE

Error Messages	Message Code
<p>On changing the Recipe status &TO_STATUS, the status for Validity Rules associated with this recipe is/are also changed to &TO_STATUS. Since the user has decided to not modify the validity rules status, All changes to this Recipe/Validity rule are roll-backed.</p>	<p>GMD_RCP_VR_STATUS</p>
<p>This recipe is used in one or more batches. Status of this recipe cannot be changed to obsolete or on-Hold.</p>	<p>GMD_RECIPE_INUSE</p>
<p>This routing is used in one or more recipes. Status of this routing cannot be changed to obsolete or on-Hold</p>	<p>GMD_ROUTING_INUSE</p>
<p>The Status cannot be changed until all dependent entities are approved to the appropriate level</p>	<p>GMD_STATUS_DEPEND_NOT_APPROVED</p>
<p>Cannot change the recipe status, as the associated validity rules require approval to be changed to &STATUS status</p>	<p>GMD_VLDT_APPR_REQD</p>

How to Get Your OPM Formula APIs Running

This appendix is used in conjunction with the rest of the guide, but is not intended to replace it. The information below is supplemental material that provides additional information regarding OPM Formula APIs.

Following are the steps to use the Formula APIs:

1. Creating a wrapper file
2. Running the wrapper file

Step 1 Creating a wrapper file

Formula APIs expect the formula data to be passed using a PL/SQL table (one of the parameters for the API). In constructing the PL/SQL table, it's important to assign all required values to the columns in this table.

As a guideline, we have provided a Formula API wrapper package, GMD_CREATE_FORMULA..create_formula or GMDPCFMB.pls. This procedure builds the PL/SQL table based on the data in a .csv file or any flat file under the UTL file directory. The UTL file directory is generally specified in the init.ora parameter file OR users could run the following sql query on the environment to execute the FM API.

```
SELECT NVL ( SUBSTR (VALUE, 1, INSTR ( VALUE, ',' ) - 1 ), VALUE )
FROM v$parameter
WHERE name = 'utl_file_dir';
```

However, it's important that the GMDPCFMB.pls file gets updated to include the correct UTL file directory path. This wrapper file parses the data from a flat file or a text file, and then constructs a PL/SQL table and passes this PL/SQL table as a parameter to the Formula API. The results from the API could be debugged and

tested after the API is executed. Refer the Formula API for more details on how to understand and debug results from the API.

Example in the flat file - Formula data stored in the flat file

```
I,MyForm10,2,0,New formula
desc,,,,0,,,1,1,OPM111,10,KGM,0,0,1,1,0,0,0,OPME,700,2060,,,Y,ORAUSR,2060,2060,1
5-Apr-02,15-Apr-02,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,eof_line
I,MyForm10,2,0,New formula
desc,,,,0,,,1,1,OPM112,20,LB,0,0,1,1,0,0,0,OPME,700,2060,,,Y,ORAUSR,2060,2060,1
5-Apr-02,15-Apr-02,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,eof_line
I,MyForm10,2,0,New formula
desc,,,,0,,,1,2,OPM113,10,LB,0,0,1,1,0,0,0,OPME,700,2060,,,Y,ORAUSR,2060,2060,1
5-Apr-02,15-Apr-02,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,eof_line
```

Example - Contents from the wrapper file - GMDCFMB.pls

```
/* Line below in this file needs to be modified to parse the flat file from the
   correct path */
/*      Opens a text file
        PLEASE NOTE: my file exists in /sqlcom/log/opm115m
        This needs to be changed !!!!!
*/
form_handle :=
    UTL_FILE.FOPEN('/sqlcom/log/opm115g','FMAPI_wocomments.csv','R');
/* Line below - shows the call made to the API and the debugging of the results
*/

    GMD_FORMULA_PUB.Insert_Formula
    ( 1.0
      , FND_API.G_FALSE
      , FND_API.G_TRUE
      , 'YES'
      , l_return_status
      , x_msg_count
      , x_msg_data
      , formula_insert_table
    );

    IF (l_return_status <> 'S') THEN
        for i IN 1 .. x_msg_count LOOP
            my_text := FND_MSG_PUB.get(i,
                                     p_encoded => FND_API.G_FALSE);
```

```
        dbms_output.put_line('The text is '||my_text);  
    END LOOP;  
END IF;
```

Step 2 Running the wrapper function

The Formula APIs are PL/SQL based and run database packages, and are run by SQL*Plus or Unix, initiating a sql session first.

Through Unix

```
sqlplus apps/apps@database @GMDPCFMB.pls
```

Through Sql*Plus

Login and start the wrapper file using:

```
Exec GMD_CREATE_FORMULA.create_formula
```

Index

A

API, 1-1

B

business objects
 formula API, 2-2
by-product, 2-2
byproduct, 2-2

C

call pass/fail status, 2-5
comma-delimited text file, 2-3
common formula package, 1-5
coproduct, 2-2
create formula detail, 1-3
create formula effectivity, 1-3
create formula header, 1-3
Create_formula, 2-3
creating a new formula, 2-3

D

descriptive flexfield segments, 2-7
detail, 1-6
detail package, private formula, 1-5
DML operation, 2-6

E

effectivity, 1-6
entity, 1-6

error, 1-2, 1-6, 3

F

flexfield segments, 2-7
FM_DUPLICATE_LINE_NO, 3
FM_EXPERIMENTAL_ITEM, 3
FM_FORMULA_ALREADY_EXISTS, 3
FM_INVALID_COST_ALLOC, 4
FM_INVALID_CUSTOMER, 4
FM_INVALID_END_DATE, 4
FM_INVALID_FORMULA, 3
FM_INVALID_FORMULA_CLASS, 3
FM_INVALID_FORMULA_PRODUCT, 4
FM_INVALID_FORMULA_TYPE, 4
FM_INVALID_FORMULA_USE, 4
FM_INVALID_ITEM, 4
FM_INVALID_ITEM_NO, 4
FM_INVALID_LINE_NO, 4
FM_INVALID_LINE_TYPE, 4
FM_INVALID_PHANTOM_TYPE, 4
FM_INVALID_RELEASE_TYPE, 5
FM_INVALID_REWORK_TYPE, 5
FM_INVALID_SCALE_TYPE, 4
FM_INVALID_SCRAP_FACTOR, 5
FM_INVALID_START_DATE, 5
FM_INVALID_UOM, 5
FM_INVALID_USER, 5
FM_MISSING_FORMULA_NO, 4
FM_MISSING_FORMULA_VERS, 4
FM_MISSING_PRODUCT, 4
FM_MISSING_UOM_CONVERSION, 5
FM_NEGATIVE_QTY, 4
FM_UOM_CONVERSION, 5

- FND_API, 1-4
- FND_API.G_FALSE, 2-5
- FND_MESSAGE, 1-4
- FND_PUB_MSG, 1-4
- fnd_user, 2-13
- FND_USR, 2-3
- formula
 - creation, 2-3
 - description, 2-2
 - number, 2-2
 - status, 2-2
 - validity, 2-2
 - version, 2-2
- formula API
 - business objects, 2-2
- Formula Create, 1-3
- formula detail, 1-3
 - creation, 1-3
 - information, 1-3
- formula effectivity
 - creation, 1-3
- formula header, 2-3
 - creation, 1-3
 - modification, 1-3
- formula information, importing, 1-2
- formula package, common, 1-5

G

- general input structure, 2-5
- GMD_ATTACH_RESOURCES, 5
- GMD_FORMULA_COMMON_PKG, 1-5
- GMD_FORMULA_DETAIL_PUB, 1-4, 1-6
- GMD_FORMULA_DETAIL_PVT, 1-5
- GMD_FORMULA_DETAIL_xxx, 1-3
- GMD_FORMULA_HEADER_PUB, 2-3
- GMD_FORMULA_INUSE, 5
- GMD_FORMULA_PUB, 1-4
- GMD_FORMULA_PVT, 1-4
- GMD_FORMULA_xxx, 1-3
- GMD_INV_TARGET_STATUS, 5
- GMD_MISSING, 5
- GMD_OPERATION_INUSE, 5
- GMD_RCP_VR_STATUS, 6
- GMD_RECIPE_INUSE, 6

- GMD_ROUTING_INUSE, 6
- GMD_STATUS_DEPEND_NOT_APPROVED, 6
- GMD_STATUS_PUB, 1-5
- GMD_VLDT_APPR_REQD, 6
- GMDPFMCS.pls, 1-5
- GMDPFMDB.pls, 1-4
- GMDPFMDS.pls, 1-4
- GMDPFMHB.pls, 1-4
- GMDPFMHS.pls, 1-4
- GMDPFMVB.pls, 1-5
- GMDPFMVS.pls, 1-5
- GMDPSTSB.pls, 1-5
- GMDPSTSS.pls, 1-5
- GMDPSURB.pls, 1-5
- GMDPSURS.pls, 1-5
- GMDVFMDB.pls, 1-5
- GMDVFMDS.pls, 1-5
- GMDVFMHB.pls, 1-4
- GMDVFMHS.pls, 1-4

H

- header, 1-6, 2-2
- header data, 2-6
- header package, private formula, 1-4

I

- importing formula information, 1-2
- ingredient, 2-2
- input structure, general, 2-5
- Insert_FormulaDetail, 1-6
- inventory item, 2-6
- item-specific data, 2-6

M

- message encoding, 2-6
- message list initialization, 2-5
- message list, specifying number of messages
 - added, 2-5
- modifying formula header information, 1-3

N

new formula creation, 2-3

O

Oracle forms, in calling function, 2-3

P

p_api_version, 2-5
p_commit, 2-5
p_formula_insert_tbl_type, 2-6, 2-7
p_formula_update_tbl_type, 2-6
p_init_msg_list, 2-5
parameters, 2-5
pass/fail status, call, 2-5
PL/SQL, 1-2, 1-6, 2-3, 2-5, 2-6
PL/SQL table, 1-6
private formula detail package, 1-5
private formula header package, 1-4
processing standard message functions, 2-6
product, 2-2
public formula surrogate package, 1-5
public formula validation package, 1-5
public package, surrogates, 1-5
public packages, validation, 1-5
public procedures, 1-6

S

standard message function processing, 2-6
stored procedure, 1-6
success, 1-2, 1-6, 3
support policy, 1-3
surrogate package, public formula, 1-5

T

text file, comma-delimited, 2-3

U

unexpected, 1-2, 1-6, 3
unknown, 1-2, 1-6, 3
user name validation, 1-6

V

valid ID, 1-6
validating the user name, 1-6
validation package, public formula, 1-5
version compatibility, validation, 2-5

W

wrapper function, 1-6, 2-6

X

x_msg_count, 2-5
x_msg_data, 2-6
x_return_status, 2-5

