Oracle® Process Manufacturing

Quality Management API User's Guide

Release 11i

Part No. B10350-02

January 2005

ORACLE®

Oracle Process Manufacturing Quality Management API User’s Guide, Release 11i

Part No. B10350-02

Primary Author: Michele-Andrea Fields

Major Contributors: Sanjay Rastogi, Karen Hunt, Harminder Verdding, Nancy Cheung, Brenda Stone

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

Contents

Send Us Your Comments

Oracle Process Manufacturing Quality Management API User's Guide, Release 11*i*

Part No. B10350-02

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. 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 about this manual?

If you find any errors or have any other suggestions for improvement, please indicate 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, and telephone number below.

____

____

____

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 Process Manufacturing Quality Management API User's Guide*.

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

- The principles and customary practices of your business area.
- *Oracle Process Manufacturing (OPM)*

  If you have never used *Oracle Process Manufacturing Quality Management APIs*, Oracle suggests you attend one or more of the *Oracle Process Manufacturing* training classes available through Oracle University.
- The Oracle Applications graphical user interface.

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

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

How To Use This Guide

This guide contains the information you need to understand and use *Oracle Process Manufacturing Quality Management APIs.*

- Chapter 1 describes the Application Program Interfaces (APIs) that support external interfaces to the OPM Quality Management tables including: packages supplied, Quality Management API bill of material, wrapper function, stored procedures, and validating the user name.
- Chapter 2 provides the relationships between the Quality Management API table structure and its entities. Discusses Quality Management API business objects, the entity relationship diagram, business object interface design, creating a new formula and importing formula data structures.
- Chapter 3 provides the relationships between the Quality Management API outbound table structure and its entities. Discusses Quality Management API outbound 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.

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.

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 Quality Management APIs*.

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*. The new features section also includes information about any features that were not yet available when this guide was printed. For example, if your administrator has installed software from a mini-packs an upgrade, this document describes the new features. Online help patches are available on MetaLink.

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

Related User's Guides

Oracle Process Manufacturing Quality Management APIs shares business and setup information with other Oracle Applications products. Therefore, you may want to refer to other user's guides when you set up and use *Oracle Process Manufacturing*.

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 Quality Management APIs* (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 Quality Management APIs*. This manual details additional steps and setup considerations for implementing *Oracle Process Manufacturing Quality Management APIs* with this feature.

Multiple Organizations in Oracle Applications

This guide describes how to set up and use *Oracle Process Manufacturing Quality Management APIs* 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*.

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 Quality Management APIs* implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This manual also provides information on creating custom reports on flexfields data.

Oracle eTechnical Reference Manuals

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

Oracle Manufacturing APIs and Open Interfaces Manual

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

Oracle Order Management Suite APIs and Open Interfaces Manual

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

Oracle Applications Message Reference Manual

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

Training and Support

Training

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

You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many Education Centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep *Oracle Process Manufacturing Quality Management 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 Process Manufacturing Quality Management APIs* and this user's guide.

Oracle values your comments and feedback. At the end of this guide is a Reader's Comment Form you can use to explain what you like or dislike about *Oracle Process Manufacturing Quality Management 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

Quality Management API Introduction

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

- Introducing the Quality Management API
- Oracle Applications Packages Supplied
- Quality Management API Bill of Materials
- Wrapper Function and Stored Procedures

Introducing the Quality Management API

Quality Management APIs let you import quality information from an existing quality management system into the Oracle Process Manufacturing Quality Management tables. When you import this information, you can include all pertinent information using a friendly tool that does not have cryptic IDs and system specific information. Quality Management APIs can process virtually all information. The interface ensures that your imported information contains the same detail as those you enter manually on the OPM Quality Management window.

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 Quality Management 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 Quality Management application.

Quality Management APIs are written in PL/SQL that can be called by your own programs. To make use of these APIs, 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

Quality Management 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

Quality Management API Support Policy

Quality Management 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

Quality Management 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.

Quality Management API Bill of Materials

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

Package Name	**File Names**	**Description**
GMD_SPEC_PUB	GMDPSPCS.pls, GMDPSPCB.pls	Public level package that the user defined function calls. The business API can be used for creating, modifying, or deleting a specifications.
GMD_SPEC_GRP	GMDGSPCS.pls, GMDGSPCB.pls	Group level specifications package containing validation procedures and functions for verifying specification and spec test data.
GMD_SPEC_VRS_PUB	GMDPSVRS.pls, GMDPSVRB.pls	Public level specification validity rules package supplying APIs to create or delete specifications and associated specification validity rules.
GMD_SPEC_VRS_GRP	GMDGSVRS.pls, GMDGSVRB.pls	Group level specification validity rules package containing validation procedures and functions for verifying specification validity rules data.
GMD_SAMPLES_PUB	GMDPSMPS.pls, GMDPSMPB.pls	Public level samples package supplying APIs to create or delete samples and associated entities.

Package Name	File Names	Description
GMD_SAMPLES_GRP	GMDGSMPS.pls, GMDGSMPB.pls	Group level samples package containing validation procedures and functions for verifying samples.
GMD_RESULTS_GRP	GMDGRESS.pls, GMDGRESB.pls	Group level results package containing validation procedures and functions for verifying results.
GMD_RESULTS_PUB	GMDPRESS.pls, GMDPRESB.pls	Public level results package supplying APIs to record results and add tests to a sample.
GMD_QC_TESTS_PUB	GMDPTSTS.pls, GMDPTSTB.pls	Public level QC Tests package containing validation procedures and functions for verifying test and results data.
GMD_QC_TESTS_GRP	GMDGTSTS.pls, GMDGTSTB.pls	Group level QC Tests package containing validation procedures and functions for verifying test and results data.
GMD_OUTBOUND_APIS_PUB	GMDOAPIS.pls, GMDOAPIB.pls	Public level QM package containing APIs to export inoformation to third party products.

Understanding the General Input Structure

All stored procedure APIs 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
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.
p_init_msg_list	varchar2	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 FND_API.G_FALSE which means that the API does not initialize the message list.
p_commit	varchar2	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 FND_API.G_FALSE.
p_validation_ level	varchar2	IN	N	For future use.
x_return_status	varchar2	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
x_msg_count	number	OUT	N	Specifies number of messages added to message list.
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.

API Hints

For performance improvement, NOCOPY hints have been added to the OUT parameters of the APIs. When an API has the same type of parameter defined as IN and OUT, pass in different variables.

For example, the Create_sample API contains p_qc_samples_rec and x_qc_samples_rec:

```
Create_Sample(p_qc_samples_rec GMD_SAMPLES%ROWTYPE,
              x_qc_samples_rec GMD_SAMPLES%ROWTYPE)
```

Therefore, the call can be set up to read:

```
create_sample(p_qc_samples_rec => l_sample
              x_qc_samples_rec => l_sample)
```

In this example, p_qc_samples_rec and x_qc_samples_rec both have the variable l_sample. This gives an incorrect result because both the parameters cannot have the same variable.

You must set the call up so that p_qc_samples_rec and x_qc_samples_rec have different variables:

```
create_sample(p_qc_samples_rec => l_in_sample
              x_qc_samples_rec => l_out_sample)
```

In addition, check the return status of the API (generally returned through x_return_status parameter) before looking at other OUT variables returned by the API. If the return status is not Success, then you must not use any of the OUT parameters passed back from the API.

2

Quality Management Inbound API Business Objects

To use Quality Management 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 quality data structures and manage any errors that arise during these operations.

The following topics are discussed:

- OPM Quality Specifications APIs
- OPM Quality Specification Validity Rules APIs
- OPM Quality Test APIs
- OPM Quality Samples APIs
- OPM Quality Results APIs

OPM Quality Specifications APIs

A specification defines the tests to be carried out on an item for quality control purposes. Each specification is identified by its name and version; together these form a unique key. The specification references an item and, optionally, a grade. It also has an owner and an owning organization.

The specification owns one or more specification tests specifying the test procedures to be carried out against the item identified in the specification header.

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

- Specification Header
- Specification Detail

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

Create Specification

This procedure lets you:

- insert a specification into GMD_SPECIFICATIONS.
- insert one or more specification tests into GMD_SPEC_TESTS.

All data is validated prior to insert. If an error is encountered, then an error is returned. If the API is able to insert, then a status of success is returned.

Following is the definition of Create Specification:

Parameter	Required	In/Out	Description
p_spec	Y	IN	Record Structure of Specifications.
p_spec_tests_tbl	Y	IN	Table Structure of Specification tests.
p_user_name	Y	IN	Oracle Applications login.
x_spec	Y	OUT	Record Structure of Specifications.
x_spec_tests_tbl	Y	OUT	Table Structure of Specification tests.

Parameter - p_spec (IN)

This is a row type parameter that identifies the specification header. The following table explains the required columns of the row:

Parameter	Description
spec_id	Spec_id must be null when a new gmd_specifications row is being created. A valid spec_id must be provided when there is no requirement to create a new gmd_specifications row and the intention is to reference an existing one in order to add gmd_spec_tests.
spec_type	Valid values are I - Item specification or M - Monitoring specification.
spec_name	Spec_name and spec_vers must form a unique key.
spec_vers	Positive integer value. Spec_name and spec_vers must form a unique key.
spec_desc	Description of specification.
spec_status	Value 100 signifies NEW.
owner_orgn_code	Valid values from sy_orgn_mst.
owner_id	Valid values from sy_orgn_usr.
spec_type	Specification type. Valid values are I - item and M - monitoring.

Parameter - p_spec_tests_tbl (IN)

This is a table type parameter that identifies the specification tests. The following table explains the required columns of the table:

Parameter	Description
spec_id	Spec_id not required. It is populated with the spec_id from the owning gmd_specifications entity.
test_id	Must reference a valid entry in gmd_qc_tests.
test_method_id	Must reference a valid entry in gmd_test_methods.
seq	Sequence of this test within the specification. Positive integer value. Duplicates not allowed within a single specification.
test_priority	Valid values are 5N - Normal, 1L - Low, and 8H - High.
test_replicate	Number of test repetitions required per sample integer value greater than 0.

Parameter - x_spec (OUT)

This is a row type parameter that returns the populated specification.

Parameter - x_spec_tests_tbl (OUT)

This is a table type parameter that returns the populated specification tests.

Delete Specification

This procedure lets you delete mark a row in GMD_SPECIFICATIONS indicating that it is logically deleted.

This is only permitted where:

- The row is not already delete marked.
- The spec_status permits updates.
- The spec_id is supplied.

Following is the definition of Delete Specification:

Parameter	Required	In/Out	Description
p_spec	Y	IN	Record Structure of Specifications.
p_user_name	Y	IN	Oracle Applications login.
x_deleted_rows	Y	OUT	Number of Specification records deleted.

Parameter - p_spec (IN)

This is a row type parameter that identifies the specification being logically deleted. The following table explains the required columns of the row:

Parameter	Description
spec_id	The specification identified being logically deleted.

Delete Specification Tests

This procedure lets you physically delete a row in GMD_SPEC_TESTS.

This is only permitted where:

- The header GMD_SPECIFICATIONS row is not already delete marked.
- The header GMD_SPECIFICATIONS row has a spec_status which permits updates.

Following is the definition of Delete Specification Tests:

Parameter	Required	In/Out	Description
p_spec_tests_tbl	Y	IN	Table Structure of Specification Tests.
x_deleted_rows	Y	OUT	Number of Specification test records deleted.

Parameter - p_spec_tests_tbl (IN)

This is a table type parameter that identifies the specification tests being physically deleted. The following table explains the required columns of the table:

Parameter	Description
spec_id	Spec_id is required. It is a valid specification.
test_id	Must reference a valid entry in gmd_tests.

OPM Quality Specification Validity Rules APIs

The circumstances under which a specification is used are determined by its validity rules. Specification validity rules (SVRs) fall into the following categories that dictate the usage type:

- Inventory
- WIP
- Supplier
- Customer

A single specification can have multiple validity rules both across and within the four categories. The selection of a specification is controlled by the criteria particular to the validity rules. These can be of a general or specific nature.

Create Inventory Specification Validity Rules

This procedure lets you insert one or more rows into GMD_INVENTORY_SPEC_VRS.

Following is the definition of Create Inventory Specification Validity Rules:

Parameter	Required	In/Out	Description
p_inventory_spec_vrs_tbl	Y	IN	Input table structure for Inventory Specification Validity Rule data.
p_user_name	Y	IN	Oracle Applications username.
x_inventory_spec_vrs_tbl	Y	OUT	Table structure containing inserted Inventory Specification Validity Rules.

Parameter - p_inventory_spec_vrs_tbl (IN)

This is a table type parameter that identifies the inventory specification validity rule. The following table explains the required columns of the table:

Parameter	Description
spec_vr_id	Must be null. A unique spec_vr_id is generated by the insert processing.
spec_id	Must reference a valid, non-delete marked entry in gmd_specifications.
spec_vr_status	Must be 100 (New) in a create scenario.

Parameter	Description
start_date	Valid date.

Parameter - x_inventory_spec_vrs_tbl (OUT)

This is a table type parameter that returns the populated inventory specification validity rule.

Create WIP Specification Validity Rules

This procedure lets you insert one or more rows into GMD_WIP_SPEC_VRS.

Following is the definition of Create WIP Specification Validity Rules:

Parameter	Required	In/Out	Description
p_wip_spec_vrs_tbl	Y	IN	Input table structure for WIP Specification Validity Rule data.
p_user_name	Y	IN	Oracle Applications username.
x_wip_spec_vrs_tbl	Y	OUT	Table structure containing inserted WIP Specification Validity Rules.

Parameter - p_wip_spec_vrs_tbl (IN)

This is a table type parameter that identifies the work in progress specification validity rule. The following table explains the required columns of the table:

Parameter	Description
spec_vr_id	Must be null. A unique spec_vr_id is generated by the insert processing.
spec_id	Must reference a valid, non-delete marked entry in gmd_ specifications.
spec_vr_status	Must be 100 (New) in a create scenario.
start_date	Valid date.

Parameter - x_wip_spec_vrs_tbl (OUT)

This is a table type parameter that returns the populated work in progress specification validity rule.

Create Customer Specification Validity Rules

This procedure lets you insert one or more rows into GMD_CUSTOMER_SPEC_VRS.

Following is the definition of Create Customer Specification Validity Rules:

Parameter	Required	In/Out	Description
p_customer_spec_vrs_tbl	Y	IN	Input table structure for Customer Specification Validity Rule data.
p_user_name	Y	IN	Oracle Applications username.
x_customer_spec_vrs_tbl	Y	OUT	Table structure containing inserted Customer Specification Validity Rules.

Parameter - p_customer_spec_vrs_tbl (IN)

This is a table type parameter that identifies the customer specification validity rule. The following table explains the required columns of the table:

Parameter	Description
spec_vr_id	Must be null. A unique spec_vr_id is generated by the insert processing.
spec_id	Must reference a valid, non-delete marked entry in gmd_specifications.
cust_id	Must reference a valid cust_account_id in hz_cust_accounts_all.
spec_vr_status	Must be 100 (New) in a create scenario.
start_date	Valid date.

Parameter - x_customer_spec_vrs_tbl (OUT)

This is a table type parameter that returns the populated customer specification validity rule.

Create Supplier Specification Validity Rules

This procedure lets you insert one or more rows into GMD_SUPPLIER_SPEC_VRS.

Following is the definition of Create Supplier Specification Validity Rules:

Parameter	Required	In/Out	Description
p_supplier_spec_vrs_tbl	Y	IN	Input table structure for Supplier Specification Validity Rule data.
p_user_name	Y	IN	Oracle Applications username.
x_supplier_spec_vrs_tbl	Y	OUT	Table structure containing inserted Supplier Specification Validity Rules.

Parameter - p_supplier_spec_vrs_tbl (IN)

This is a table type parameter that identifies the supplier specification validity rule. The following table explains the required columns of the table:

Parameter	Description
spec_vr_id	Must be null. A unique spec_vr_id is generated by the insert processing.
spec_id	Must reference a valid, non-delete marked entry in gmd_specifications.
supplier_id	Must reference a valid vendor_id in po_vendors.
spec_vr_status	Must be 100 (New) in a create scenario.
start_date	Valid date.

Parameter - x_supplier_spec_vrs_tbl (OUT)

This is a table type parameter that returns the populated supplier specification validity rule.

Create Monitoring Specification Validity Rules

This procedure creates monitoring specification validity rules. It accepts a table of Monitoring SpecificationValidity Rule definitions, validates each table entry and if found valid, then it inserts a corresponding row into the gmd_monitoring_spec_vrs table.

Following is the definition of Create Monitoring Specification Validity Rules:

Parameter	Required	In/Out	Description
p_monitoring_spec_vrs_tbl	Y	IN	Input table structure for Monitoring Specification Validity Rule data.

Parameter	Required	In/Out	Description
p_user_name	Y	IN	Oracle Applications username.
x_monitoring_spec_vrs_tbl	Y	OUT	Table structure containing inserted Monitoring Specification Validity Rules.

Parameter - p_monitoring_spec_vrs_tbl (IN)

This is a table type parameter that identifies the monitoring specification validity rule. The following table explains the required columns of the table:

Parameter	Description
spec_vr_id	Must be null. A unique spec_vr_id is generated by the insert processing.
spec_id	Must reference a valid, non-delete marked entry in gmd_specifications.
spec_vr_status	Must be 100 (New) in a create scenario.
start_date	Valid date.

Parameter - x_monitoring_spec_vrs_tbl (OUT)

This is a table type parameter that returns the populated monitoring specification validity rule.

Delete Inventory Specification Validity Rules

This procedure lets you logically delete one or more rows in GMD_INVENTORY_SPEC_VRS.

This is permitted where:

- The header GMD_SPECIFICATIONS row is not already delete marked.
- The targeted row in GMD_INVENTORY_SPEC_VRS is not already delete marked.

Following is the definition of Delete Inventory Specification Validity Rules:

Parameter	Required	In/Out	Description
p_inventory_spec_vrs_tbl	Y	IN	Input table structure for Inventory Specification Validity Rule data.

Parameter	Required	In/Out	Description
p_user_name	Y	IN	Oracle Applications username.
x_deleted_rows	Y	OUT	Number of rows deleted.

Parameter - p_inventory_spec_vrs_tbl (IN)

This is a table type parameter that identifies the inventory specification validity rule. The following table explains the required columns of the table:

Parameter	Description
spec_vr_id	Cannot be NULL. This refers to a valid existing spec vr id.
spec_id	Must reference a valid, non-delete marked entry in gmd_ specifications.

Delete WIP Specification Validity Rules

This procedure lets you logically delete one or more rows in GMD_WIP_SPEC_ VRS.

This is permitted where:

- The header GMD_SPECIFICATIONS row is not already delete marked.
- The targeted row in GMD_WIP_SPEC_VRS is not already delete marked.

Following is the definition of Delete WIP Specification Validity Rules:

Parameter	Required	In/Out	Description
p_wip_spec_vrs_tbl	Y	IN	Input table structure for WIP Specification Validity Rule data.
p_user_name	Y	IN	Oracle Applications username.
x_deleted_rows	Y	OUT	Number of rows deleted.

Parameter - p_wip_spec_vrs_tbl (IN)

This is a table type parameter that identifies the work in progress specification validity rule. The following table explains the required columns of the table:

Parameter	Description
spec_vr_id	Cannot be NULL. This refers to a valid existing spec vr id.
spec_id	Must reference a valid, non-delete marked entry in gmd_ specifications.

Delete Customer Specification Validity Rules

This procedure lets you logically delete one or more rows in GMD_CUSTOMER_ SPEC_VRS.

This is permitted where:

- The header GMD_SPECIFICATIONS row is not already delete marked.
- The targeted row in GMD_CUSTOMER_SPEC_VRS is not already delete marked.

Following is the definition of Delete Customer Specification Validity Rules:

Parameter	Required	In/Out	Description
p_customer_spec_vrs_ tbl	Y	IN	Input table structure for Customer Specification Validity Rule data.
p_user_name	Y	IN	Oracle Applications username.
x_deleted_rows	Y	OUT	Number of rows deleted.

Parameter - p_customer_spec_vrs_tbl (IN)

This is a table type parameter that identifies the customer specification validity rule. The following table explains the required columns of the table:

Parameter	Description
spec_vr_id	Cannot be NULL. This refers to a valid existing spec vr id.
spec_id	Must reference a valid, non-delete marked entry in gmd_ specifications.

Delete Supplier Specification Validity Rules

This procedure lets you logically delete one or more rows in GMD_SUPPLIER_ SPEC_VRS.

This is permitted where:

- The header GMD_SPECIFICATIONS row is not already delete marked.
- The targeted row in GMD_SUPPLIER_SPEC_VRS is not already delete marked.

Following is the definition of Delete Supplier Specification Validity Rules:

Parameter	Required	In/Out	Description
p_supplier_spec_vrs_tbl	Y	IN	Input table structure for Supplier Specification Validity Rule data.
p_user_name	Y	IN	Oracle Applications username.
x_deleted_rows	Y	OUT	Number of rows deleted.

Parameter - p_supplier_spec_vrs_tbl (IN)

This is a table type parameter that identifies the supplier specification validity rule. The following table explains the required columns of the table:

Parameter	Description
spec_vr_id	Cannot be NULL. This refers to a valid existing spec vr id.
spec_id	Must reference a valid, non-delete marked entry in gmd_specifications.

Delete Monitoring Specification Validity Rules

This procedure deletes multiple monitoring specification validity rules. It accepts a table of Monitoring SpecificationValidity Rule definitions and validates each table entry to ensure the corresponding row is not already delete marked. When validation is successful, a logical delete is performed setting the delete_mark to 1.

Following is the definition of Delete Monitoring Specification Validity Rules:

Parameter	Required	In/Out	Description
p_monitoring_spec_vrs_tbl	Y	IN	Input table structure for Monitoring Specification Validity Rule data.
p_user_name	Y	IN	Oracle Applications username.
x_deleted_rows	Y	OUT	Number of rows deleted.

Parameter - p_monitoring_spec_vrs_tbl (IN)

This is a table type parameter that identifies the monitoring specification validity rule. The following table explains the required columns of the table:

Parameter	Description
spec_vr_id	Cannot be NULL. This refers to a valid existing spec vr id.
spec_id	Must reference a valid, non-delete marked entry in gmd_specifications.

OPM Quality Test APIs

A Quality Test is required when a sample is taken for a product. The test must have a description, type, and code. The type is used to determine what values can be recorded for this test. The values entered determine what values can be recorded for this test. Customer values can also be recorded.

The Create Test API lets you create a new test, test_values, and customer specific tests. It also lets you create values or customer tests for a test that already exists.

The Delete Test API also supports the deletion of either a test header record, details values, or customer specific tests. The delete APIs are:

- DELETE_TEST_HEADERS - This deletes a test header record. Specify the test_code or the unique test_id in the test_record being passed. This logically marks the record as deleted.
- DELETE_TEST_VALUES - This accepts one or more test values that must be deleted for a test. Specify the test_id in the record being passed, as well as the unique test value id. This physically deletes the records.
- DELETE_CUSTOMER_TESTS- This accepts one or more customer test values that must be deleted for a test. You must specify the test_id in the record being passed, as well as the unique customer id. This physically deletes the records.

You must first delete the test header record before you can delete the test values or customer tests.

Create Tests

This procedure is used to create:

- a test header
- test values
- customer test records

It can also be used to create test values or customer tests for a test that already exists. This routine determines what entities are required for creation based on the input parameters passed.

Following is the definition of Create Tests:

Parameter	Required	In/Out	Description
p_qc_tests_rec	Y	IN	Record structure of Test Headers.

Parameter	Required	In/Out	Description
p_qc_test_values_tbl	Y	IN	Table structure of Test Values.
p_qc_cust_tests_tbl	Y	IN	Table structure of Customer Tests.
p_user_name	Y	IN	Oracle Applications login.
x_qc_tests_rec	Y	OUT	Record structure of Test Headers.
x_qc_test_values_tbl	Y	OUT	Table structure of Test Values.
x_qc_cust_tests_tbl	Y	OUT	Table structure of Customer Tests.

Parameter - p_qc_tests_rec (IN)

This is a row type parameter that identifies the tests. The following table explains the required columns of the row:

Parameter	Description
test_id	Must be null if the requirement is to create a new row in gmd_qc_tests. A unique test_id is generated by the insert processing. Or, if the child records are being added to an exisiting gmd_qc_tests row, then the test_id must reference a valid, non delete marked row.
test_code	A code to uniquely identify the test.
test_desc	Free format text description.
test_method_id	Test method ID.
test_type	Test type. Note that there are additional fields required depending on the test type.

Parameter - p_qc_test_values_tbl (IN)

This is a table type parameter that identifies the test values. The following table explains the required columns of the table:

Parameter	Description
test_value_id	Must be null. A unique test_value_id is generated by the insert processing.
test_id	Must identify the parent gmd_qc_tests where this already exists. If the parent gmd_qc_tests is being created in the same API call, then this must be null.

Parameter - p_qc_cust_tests_tbl (IN)

This is a table type parameter that identifies the customer test values. The following table explains the required columns of the table:

Parameter	Description
test_id	Must identify the parent gmd_qc_tests where this already exists. If the parent gmd_qc_tests is being created in the same API call, then this must be null.
cust_id	Must identify a valid cust_account_id in hz_cust_accounts_all.

Parameter - x_qc_tests_rec (OUT)

This is a row type parameter that returns the populated test.

Parameter - x_qc_test_values_tbl (OUT)

This is a table type parameter that returns the populated test values.

Parameter - x_qc_cust_tests_tbl (OUT)

This is a table type parameter that returns the populated customer test values.

Delete Test Headers

This procedure is used to logically delete (set delete_mark=1) the entity test header - GMD_QC_TESTS.

Following is the definition of Delete Test Headers:

Parameter	Required	In/Out	Description
p_qc_tests_rec	Y	IN	Record structure of Test Headers.
p_user_name	Y	IN	Oracle Applications login.

Parameter - p_qc_tests_rec (IN)

This is a row type parameter that identifies the tests. The following table explains the required columns of the row:

Parameter	Description
test_id	Required if test_code is NULL.
test_code	Required if test_id is NULL.

Delete Test Values

This procedure lets you physically delete one or more rows in GMD_QC_TEST_VALUES.

This is permitted where:

- The header GMD_QC_TESTS row is not delete marked.
- All targeted rows in GMD_QC_TEST_VALUES belong to the same header (GMD_QC_TESTS).

Following is the definition of Delete Test Values:

Parameter	Required	In/Out	Description
p_qc_test_values_tbl	Y	IN	Table structure of Test Values.
x_deleted_rows	Y	OUT	Number of Test Value records Deleted.

Parameter - p_qc_test_values_tbl (IN)

This is a table type parameter that identifies the test values. The following table explains the required columns of the table:

Parameter	Description
test_id	Test ID of the test value being deleted.
test_value_id	Identifier of the test value.

Delete Customer Tests

This procedure lets you physically delete one or more rows in GMD_CUSTOMER_TESTS.

This is permitted where:

- The header GMD_QC_TESTS row is not delete marked.
- All targeted rows in GMD_CUSTOMER_TESTS belong to the same header (GMD_QC_TESTS).

Following is the definition of Delete Customer Tests:

Parameter	Required	In/Out	Description
p_qc_cust_tests_tbl	Y	IN	Table structure of Test Customer Display.

Parameter	Required	In/Out	Description
x_deleted_rows	Y	OUT	Number of Test Customer Display records deleted.

Parameter - p_qc_cust_tests_tbl (IN)

This is a table type parameter that identifies the customer test values. The following table explains the required columns of the table:

Parameter	Description
test_id	Must identify the parent gmd_qc_tests where this already exists.
cust_id	Must identify a valid cust_account_id in hz_cust_accounts_all.

OPM Quality Samples APIs

A sample is a representative quantity of material that undergoes analysis to record quality characteristics. It can be drawn in response to the creation of a sampling event or simply on a standalone basis.

A sampling event documents the need for a sample to be taken and can arise in one of the following categories:

- Inventory
- WIP
- Supplier
- Customer
- Resource
- Location

Every sample belongs to a sampling event.

A specification is a group of tests that must be performed on samples associated to it. Each specification has at least one specification validity rule used to dictate when a sampling event (business event) is generated. Validity Rules can exist in the following areas:

- Inventory
- WIP
- Supplier
- Customer
- Resource
- Location

For example, the specification New Recipe has a WIP validity rule which is set up for any batch using recipe R1. If a batch is created using this recipe, then any samples taken from it can be attached to the New Recipe specification. Then, each test associated with the specification is attached to the sample.

A sample can be taken that is not associated to a specification. The sampling event which owns that sample, and any other samples created with the same source data, also does not have a specification associated to it.

The Samples API can be used to:

- create a sample, including create or update associated sampling event, attach result records, or set up disposition records.
- delete a sample and update associated sampling event.

The following public APIs are provided in package GMD_SAMPLES_PUB to support the creation and deletion of samples:

- CREATE_SAMPLES - provides a mechanism to create the samples
- DELETE_SAMPLES - supports the logical deletion of GMD_SAMPLES

Create Samples

This procedures lets you:

- insert a row into GMD_SAMPLES.
- associate the sample_row GMD_SAMPLES with an existing sampling event GMD_SAMPLING_EVENT, or insert a new GMD_SAMPLING_EVENT. Every sample belongs to a sampling event. Where a sampling_event_id is supplied, it is assumed that the sample is being associated with this. Where no sampling_event_id is supplied, a new row is created. Therefore, parameter p_find_matching_spec is examined. This determines whether the sampling event should be free-standing or associated with a specification. If the parameter is set to Y, then the specification validity rules are examined to find the specification with the closest match to the supplied input data. They can be input as follows:
- insert a row into GMD_SAMPLE_SPEC_DISP. This is necessary to record the disposition of the current instance of the sampling plan. It indicates whether all the requirements for the sampling plan have been met. For example, if the sampling plan requires three lot samples, then the disposition is changed to COMPLETE when all three samples are created.
- insert a row into GMD_EVENT_SPEC_DISP. This is necessary to record the disposition of the current sample in terms of the results for all tests prescribed by the specification. Test results may be considered in specification or out of specification.
- insert a result set, which is one or more rows in GMD_RESULTS. One result row is inserted for each replicate of each test in the associated specification. For example, if a test has a replicate number of two, then two result rows are inserted into the table. Each row ID acts as a skeleton entry put in place for recording actual test results.

- insert a row into GMD_SPEC_RESULTS. This is a skeleton entry put in place for recording the results of testing the sample. When a sample is new, the relationship between gmd_results and gmd_spec_results is one-to-one.

All data is validated prior to insert.

Following is the definition of Create Samples:

Parameter	Required	In/Out	Description
p_qc_samples_rec	Y	IN	Record structure for samples input.
p_user_name	Y	IN	Oracle Applications login.
p_find_matching_ spec	N	IN	Flag to ensure that sampling event has specification associated.
p_grade	N	IN	Grade
x_qc_samples_rec	Y	OUT	Record structure for samples output.
x_sampling_events_ rec	Y	OUT	Record structure for sampling events.
x_sample_spec_disp	Y	OUT	Record structure for Sample Specification Disposition.
x_event_spec_disp_ rec	Y	OUT	Record structure for Event Specification Disposition.
x_results_tab	Y	OUT	Table structure for results.
x_spec_results_tab	Y	OUT	Table structure for Specification results.

Parameter - p_qc_samples_rec (IN)

This is a table type parameter that identifies the sample records. The following table explains the required columns of the table:

Parameter	Description
sampling_event_id	Where supplied, this must reference a valid, non delete marked entry in gmd_sampling_events. Where a null value is supplied, it assumes that the gmd_sampling_events row must be created.
sample_id	Must be null. A unique sample_id is generated by the insert processing.

Parameter	Description
sample_no	Unique identifier for the sample. Automatically generated where document numbering is in place for document type SMPL. Otherwise, the unique identifier must be supplied.
sample_organization	The organization code of the organization creating the sample.
source	Identifies the source of the sample. Valid values are I - Inventory, W - Wip, C - Customer, S - Supplier.
sample_type	Sample type.
sampler_id	Valid user_id from fnd_user.

Parameter - x_qc_samples_rec (OUT)

This is a row type parameter that returns the populated samples record.

Parameter - x_sampling_events_rec (OUT)

This is a row type parameter that returns the populated sampling events record.

Parameter - x_event_disp_rec (OUT)

This is a row type parameter that returns the populated event disposition record.

Parameter - x_sample_spec_disp (OUT)

This is a row type parameter that returns the populated sample specification disposition.

Parameter - x_results_tab (OUT)

This is a table type parameter that returns the populated results for the sample.

Parameter - x_spec_results_tab (OUT)

This is a table type parameter that returns the populated specification results for the sample.

Delete Sample

This procedure lets you logically delete one or more rows in GMD_SAMPLES. This is permitted where the targeted row in GMD_SAMPLES is not already delete marked.

Following is the definition of Delete Sample:

Parameter	Required	In/Out	Description
p_qc_samples_rec	Y	IN	Record structure for samples.
p_user_name	Y	IN	Oracle Applications login.

Parameter - p_qc_samples_rec (IN)

This is a table type parameter that identifies the sample records. The following table explains the required columns of the table:

Parameter	Description
sample_id	If specified, then orgn_code and sample_no can be NULL.
orgn_code	If specified with sample_no, then sample_id can be NULL.
sample_no	If specified with orgn_code, then sample_id can be NULL.

OPM Quality Results APIs

Prior to documenting results, create a specification, a validity rule, and a sample. The sample has a set of tests against which results must be recorded. Therefore, when the sample is created, a result set is also established.

The Record Results API lets actual results be recorded for a particular test. The result row already exists since it is created with the sample, but the detail of the test result is now put in place.

The Add Tests to Samples API lets the result set for a sample be extended. New tests or replications of tests can be added.

Record Results

This procedure can be used to record actual test results. It modifies the appropriate row in GMD_RESULTS which is put in place when the result set is created for a particular sample. The input result is analyzed to determine if it is considered in specification or out of specification and this information is used to update GMD_SPEC_RESULTS.in_spec_ind.

Following is the definition of Record Results:

Parameter	Required	In/Out	Description
p_results_rec	Y	IN	Record structure for gmd_results table.
p_user_name	Y	IN	Oracle Applications username.
x_results_rec	Y	OUT	Record structure for gmd_results table.
x_spec_results_rec	Y	OUT	Record structure for gmd_spec_results table.

Parameter - p_results_rec (IN)

This is a row type parameter that identifies the results being recorded. The following table explains the required columns of the row:

Parameter	Description
result_value	This is the value for the test result. This is a required value.
result_id	Identifies a specific result in a specific sample. This is required if no other values except for result_value are passed.

Parameter	Description
sample_id	The ID for the sample of the result. This is required, if sample_no and orgn_code are not defined. Otherwise, the system checks sample_id first. You must use the sample_id or sample_no and orgn_code combination with the test_id or test_code and test_replicate_cnt combination.
sample_no	In conjunction with orgn_code, must reference a valid sample in gmd_samples. If specified, then sample_id is not required.
orgn_code	In conjunction with sample_no, must reference a valid sample in gmd_samples.
test_id	The ID for the test associated to a result. This is required, if test_code is not defined. Otherwise, the system checks test_id first.
test_code	Test code is used whenever test_id is not passed.
test_replicate_cnt	Used in conjunction with sample and test identifiers to determine the required row in gmd_results.
result_date	You can pass a date to this field. If no date is passed, then it defaults to SYSDATE.

Example:

The following combinations are valid:

- result_value and result_id
- result_value with sample_id and test_id and test_replicate_cnt
- result_value with sample id and test_code with test_replicate_cnt
- result value with sample_no and orgn_code with test_id and test_replicate_cnt
- result_value with orgn_code, sample_no, test_code, test_replicate_cnt

Parameter - x_results_rec (OUT)

This is a row type parameter that returns the results.

Parameter - x_spec_results_rec (OUT)

This is a row type parameter that returns the specification results.

Add Tests to Samples

This procedure lets tests be added to an existing sample. For each test referenced, rows are created in GMD_RESULTS and GMD_SPEC_RESULTS which is done in preparation for recording the outcome of the testing.

Each results row references a test. There can be multiple replications of a test. After the results rows have been added, dispositions are set to In Progress on the following entities:

- GMD_SAMPLES
- GMD_SAMPLING_EVENTS
- GMD_SAMPLE_SPEC_DISP
- GMD_EVENT_SPEC_DISP

Following is the definition of Add Tests to Samples:

Parameter	Required	In/Out	Description
p_sample_rec	Y	IN	Record structure for gmd_samples table.
p_user_name	Y	IN	Oracle Applications username.
p_test_id_tab	Y	IN	Table structure for Table IDs GMD_API_Pub.number_tab
p_event_spec_disp_id	Y	IN	Event specification disposition ID.
x_results_tab	Y	OUT	Table structure for GMD_API_Pub.gmd_results_tab.
x_spec_results_tab	Y	OUT	Table structure for GMD_API_Pub.gmd_spec_results_tab.
p_result_date	Y	IN	You can pass a date to this field. If no date is passed, then it defaults to SYSDATE.

Parameter - p_sample_rec (IN)

This is a row type parameter that identifies the sample being recorded. The following table explains the required columns of the row:

Parameter	Description
sample_id	The ID for the sample of the result. This is required, if sample_no and orgn_code are not defined. Otherwise, the system checks sample_id first. You must use the sample_id or sample_no and orgn_code combination with the test_id or test_code combination.
sample_no	In conjunction with orgn_code, must reference a valid sample in gmd_samples.
orgn_code	In conjunction with sample_no, must reference a valid sample in gmd_samples.

Parameter - p_test_id_tab (IN)

This is a table type parameter that identifies the test ID being recorded. The following table explains the required columns of the row:

Parameter	Description
test_id	The ID for the test associated to a result. This is required, if test_code is not defined. Otherwise, the system checks test_id first.

Parameter - x_results_tab (OUT)

This is a table type parameter that returns the results.

Parameter - x_spec_results_tab (OUT)

This is a table type parameter that returns the specification results.

3

Quality Management Outbound API Business Objects

OPM provides outbound APIs to export Quality Management data to third party systems. This download of Quality information facilitates further analysis and reporting in an external system. The outbound APIs include:

- Fetch Test Methods
- Fetch Tests
- Fetch Specifications and Validity Rules
- Fetch Samples
- Fetch Sample Groups
- Fetch Results
- Fetch Composite Results

Fetch Test Methods

This procedure retrieves test method data from the table GMD_TEST_METHODS. If an error is encountered, then an error status is returned.

If the API is able to select one or more test methods without error, then a status of success is returned and the rows are returned in the output table x_test_methods_tbl.

Following is the definition of Fetch Test Methods:

Parameter	Required	In/Out	Description
p_from_test_method_code	N	IN	If specified, defines the start value for a range-based retrieval.
p_to_test_method_code	N	IN	If specified, defines the end value for a range-based retrieval.
p_test_method_id	N	IN	If specified, restricts the retrieval to the single test_method_id.
p_test_kit_item_no	N	IN	If specified, restricts the retrieval to the single item_no.
p_test_kit_item_id	N	IN	If specified, restricts the retrieval to the single item_id.
p_resource	N	IN	If specified, restricts the retrieval to the single resource.
p_delete_mark	N	IN	If specified, retrieval is restricted to deleted rows (if specified as 1) or undeleted rows (if specified as 0).
p_from_last_update_date	N	IN	If specified, defines the start date for a range-based retrieval.
p_to_last_update_date	N	IN	If specified, defines the end data for a range-based retrieval.
p_user_name	Y	IN	Oracle Applications user name.
x_test_methods_table	Y	OUT	Record type gmd_test_methods_tab_type.

Parameter - x_test_methods_table (OUT)

This is a table type parameter that returns the fetched test method information.

Parameter	Description
Test_method_id	Test method identifier.
Test_method_code	Test method code.
Test_method_desc	In the language of the session, actual text comes from gmd_test_methods_tl via gmd_test_methods_vl view. If the descriptions do not exist in the language of the session, then the retrieval fails and test method data is not returned.
Test_qty	Test quantity.
Test_uom	Test unit of measure.
Delete_mark	Logical delete record indicator.
Display_precision	Display precision.
Test_duration	Test duration.
Days	Test duration in number of days.
Hours	Test duration in number of hours.
Minutes	Test duration in number of minutes.
Seconds	Test duration in number of seconds.
Test_replicate	Test replicate.
Resources	Resources.
Test_kit_item_id	Test kit item.
Test_kit_item_no	Test kit item number.
Text_code	Text code.
attribute1-30	Descriptive Flexfield Segments.
attribute_category	Descriptive Flexfield Segment Category.
creation_date	Standard Who Column.
created_by_id	Standard Who Column.
created_by_user	Standard Who Column.
last_updated_by_id	Standard Who Column.
Last_updated_by_user	Standard Who Column.
last_update_date	Standard Who Column.

Parameter	Description
Last_update_login	Standard Who Column.

Fetch Tests

The Fetch Tests API retrieves tests from GMD_TEST_QC_TESTS, GMD_QC_TEST_VALUES, and GMD_CUSTOMER_TESTS.

Following is the definition of Fetch Tests.

Parameter	Required	In/Out	Description
p_from_test_code	N	IN	Starting test code.
p_to_test_code	N	IN	Ending test code.
p_from_test_method_code	N	IN	Starting test method code.
p_to_test_method_code	N	IN	Ending test method code.
p_test_id	N	IN	Testing identifier.
p_test_method_id	N	IN	Test method identifier.
p_test_class	N	IN	Test class.
p_test_type	N	IN	Test type.
p_priority	N	IN	Priority
p_delete_mark	N	IN	Logical delete record indicator.
p_from_last_update_date	N	IN	Starting last update date.
p_to_last_update_date	N	IN	Ending last update date.
p_user_name	Y	IN	Oracle Applications user name.
x_tests_table	Y	OUT	Record type gmd_qc_tests_tab_type.

Parameter - x_tests_table (OUT)

This is a table type parameter that returns the fetched header information for the test.

Field/Column	Description
Row_id	Row ID.
Test_id	Test identifier.

Field/Column	Description
Test_Code	Test code.
Test_desc	Test description.
Test_method_id	Test method ID.
Test_oprn_line_id	Test operation line ID.
Test_provider_code	Test provider code.
Test_class	Test classes.
Test_type	Test type.
Test_unit	Test unit.
Min_value_num	Minimum value.
Max_value_num	Maximum value.
Exp_error_type	Expression error type.
Below_spec_min	Below specification minimum.
Above_spec_min	Above specification minimum.
Below_spec_max	Below specification maximum.
Above_spec_max	Above specification maximum.
Below_min_action_code	Below minimum action code.
Above_min_action_code	Above minimum action code.
Below_max_action_code	Below maximum action code.
Above_max_action_code	Above maximum action code.
Expression	Expression
Display_precision	Display precision.
Report_precision	Report precision.
Priority	Priority.
Test_oprn_id	Test operation ID.
Delete_mark	Logical delete record indicator.
attribute1-30	Descriptive Flexfield Segments.
attribute_category	Descriptive Flexfield Segment Category.
creation_date	Standard Who Column

Field/Column	Description
created_by	Standard Who Column
last_updated_by	Standard Who Column
last_update_date	Standard Who Column
Last_update_login	Standard Who Column
x_qc_test_values_tab	Record type qc_tests_values_tab.
x_qc_customer_tests_tab	Record type qc_customer_test_tab.

Parameter - x_qc_test_values_tab

This is a table type parameter that returns the fetched quality test values.

Field/Column	Description
Row_id	Row identifier.
Test_value_id	Test value identifier.
Test_id	Test unique identifier.
Min_num	Numeric range from value recorded in quality units specified in test_unit.
Max_num	Numeric range to value recorded in quality units specified in test_unit.
Display_label_numeric_range	Display the label numeric range.
Test_value_desc	Test value description.
Value_char	Varchar value for list of values or text range.
Text_range_seq	Test range sequence.
Expression_ref_test_id	ID of referenced test for expression test type.
attribute1-30	Descriptive Flexfield Segments
attribute_category	Descriptive Flexfield Segment Category
creation_date	Standard Who Column.
created_by	Standard Who Column.
last_updated_by	Standard Who Column.
last_update_date	Standard Who Column.

Field/Column	Description
Last_update_login	Standard Who Column.

Parameter - x_qc_customer_tests_tab

This is a table type parameter that returns the fetched quality customer test information.

Field/Column	Description
Row_id	Row identifier.
Test_id	Test identifier.
Cust_id	Customer identifier.
Report_precision	Report precision.
Cust_test_display	Customer test display.
Text_code	Text code.
creation_date	Standard Who Column.
created_by	Standard Who Column.
last_updated_by	Standard Who Column.
last_update_date	Standard Who Column.
Last_update_login	Standard Who Column.

Fetch Specification Validity Rules

The Fetch Specification Validity Rules API retrieves specifications, specification tests, and validity rules from GMD_SPECIFICATIONS, GMD_SPEC_TESTS, GMD_WIP_SPEC_VRS, GMD_CUSTOMER_SPEC_VRS, GMD_INVENTORY_SPEC_VRS, and GMD_SUPPLIER_SPEC_VRS.

Following is the definition of Fetch Specification Validity Rules:

Parameter	Required	In/Out	Description
p_from_spec_name	N	IN	Starting specification name.
p_to_spec_name	N	IN	Ending specification name.
p_spec_id	N	IN	Specification identifier.
p_spec_version	N	IN	Specification version.
p_from_grade	N	IN	Starting grade.
p_to_grade	N	IN	Ending grade.
p_from_item_code	N	IN	Starting item code.
p_to_item_code	N	IN	Ending item code.
p_from_last_update_date	N	IN	Starting specification last update date.
p_to_last_update_date	N	IN	Ending specification last update date.
p_spec_type	N	IN	Specification type.
p_user_orgn_code	N	IN	User organization code.
p_user_name	Y	IN	Oracle Applications user name.
p_item_id	N	IN	Item identifier.
p_spec_status	N	IN	Specification status.
p_delete_mark	N	IN	Logical delete record indicator.
p_test_code	N	IN	Test code.
p_test_id	N	IN	Test identifier.
p_test_method_code	N	IN	Test method code.
p_test_method_id	N	IN	Test method identifier.

Parameter	Required	In/Out	Description
p_test_uom	N	IN	Test unit of measure.
p_test_priority	N	IN	Test priority.
p_from_test_last_update_date	N	IN	Starting last test update date.
p_to_test_last_update_date	N	IN	Ending last test update date.
p_spec_vr_status	N	IN	Specification validity rule status.
p_vr_orgn	N	IN	Validity rule organization.
p_vr_start_date	N	IN	Validity rule start date.
p_vr_end_date	N	IN	Validity rule end date.
p_vr_coa_type	N	IN	Validity rule certificate of analysis type.
p_vr_sampling_plan	N	IN	Validity rule sampling plan.
p_vr_sampling_plan_id	N	IN	Validity rule sample plan identifier.
p_vr_delete_mark	N	IN	Logical delete record indicator.
p_vr_source	N	IN	Validity rule source.
p_vr_lot_no	N	IN	Validity rule lot number.
p_vr_lot_id	N	IN	Validity rule lot identifier.
p_vr_sublot_no	N	IN	Validity rule sublot number.
p_vr_warehouse	N	IN	Validity rule warehouse.
p_vr_location	N	IN	Validity rule location.
p_vr_orgn_batch_no	N	IN	Validity rule organization batch number.
p_vr_batch_id	N	IN	Validity rule batch identifier.
p_vr_recipe_no	N	IN	Validity rule recipe number.
x_specifications_tbl		OUT	Record type specifications_tbl_type.

Parameter - x_specifications_tbl (OUT)

This is a table type parameter that returns the fetched header information for the specification.

Field/Column	Description
Row_id	Row identifier.
Spec_id	Specification identification.
Spec_name	Specification name.
Spec_vers	Specification version.
Spec_desc	Specification description.
Spec_type	Specification type.
Overlay_ind	Overlay indicator.
Base_spec_id	Base specification indicator.
Item_id	Item ID.
Grade	Grade.
Spec_status	Specification status.
Owner_orgn_code	Owner organization code.
Owner_id	Owner ID.
Sample_inv_trans_ind	Transact inventory for sample.
Delete_mark	Logical delete record indicator.
Text_code	Text code.
attribute1-30	Descriptive Flexfield Segments.
attribute_category	Descriptive Flexfield Segment Category.
creation_date	Standard Who Column.
created_by	Standard Who Column.
last_updated_by	Standard Who Column.
last_update_date	Standard Who Column.
Last_update_login	Standard Who Column.
x_spec_tests	Record type spec_tests.
x_cust_spec_vrs	Record type cust_spec_vrs.
x_wip_spec_vrs	Record type wip_spec_vrs.
x_supl_spec_vrs	Record type supl_spec_vrs.

Field/Column	Description
x_inv_spec_vrs	Record type inv_spec_vrs.
x_mon_spec_vrs	Record type mon_spec_vrs.

Parameter - x_spec_tests

This is a table type parameter that returns the header information for the fetched specification tests.

Parameter	Description
test_id	Test ID.
test_code	Test code.
test_method_id	Test method ID.
test_method_code	Test method code.
seq	Sequences.
test_qty	Test quantity.
test_uom	Test unit of measure.
min_value_num	Minimum target value for numeric range test type.
target_value_num	Test target value.
max_value_num	Maximum target value for numeric range test type.
min_value_char	Minimum target value for text range test type.
target_value_char	Target value for test when the test type is a list of values or a text range.
max_value_char	Maximum target value for text range test type.
test_replicate	The number of times a test is repeated for the same sample instance. Default value is 1.
check_result_interval	Frequency result values are checked for out-of-spec condition.
print_on_coa_ind	Print on certificate of analysis.
use_to_control_step	Use to control step.
out_of_spec_action	Out of specification action.
exp_error_type	NUM = Number, PCT = Percent.
below_spec_min	Below specification minimum.

Parameter	Description
above_spec_min	Above specification minimum.
below_spec_max	Below specification maximum.
above_spec_max	Above specification maximum.
below_min_action_code	Action code for results in the test error region specification minimum minus test error.
above_min_action_code	Action code for results in the test error region specification minimum plus test error.
optional_ind	Indicates if a test is optional. Valid values are Y - Yes, Null - No.
display_precision	Decimal precision of saved value.
report_precision	Decimal precision used to display value.
test_priority	Test priority. Valid values are H - high, N - normal, and L - low.
retest_lot_expiry_ind	Indicates if the specification is used to retest for lot expiration.
print_spec_ind	Print specification indicator.
print_result_ind	Print result indicator.
below_max_action_code	Action code for results in the test error region specification maximum minus test error.
above_max_action_code	Action code for results in the test error region specification maximum plus test error.
test_display	Test display.
days	Days.
hours	Hours.
minutes	Minutes.
seconds	Seconds.
from_base_ind	From base indicator.
exclude_ind	Exclude indicator.
modified_ind	Modified indicator.
calc_uom_conv_ind	Calculation of unit of measure conversion indicator.
to_uom	To unit of measure.
text_code	Text code.

Parameter	Description
attribute_category	Descriptive Flexfield Segment Category.
attribute1-30	Descriptive Flexfield Segments.
creation_date	Standard Who Column.
created_by_id	Standard Who Column.
created_by_name	Standard Who Column.
last_update_date	Standard Who Column.
last_updated_by_id	Standard Who Column.
last_updated_by_name	Standard Who Column.
last_update_login	Standard Who Column.

Parameter - x_cust_spec_vrs

This is a table type parameter that returns the fetched customer specification validity rule information.

Parameter	Description
spec_vr_id	Specification validity rule identifier.
orgn_code	Organization code.
cust_id	Customer identification.
cust_no	Customer number.
order_id	Order ID.
order_number	Order number.
order_line	Order line.
order_line_id	Order line ID.
ship_to_site_id	Ship-to site ID.
ship_to_site_name	Ship-to site name.
org_id	Organization ID.
org_name	Organization name.
spec_vr_status	Specification validity rule status.

Parameter	Description
spec_vr_status_ description	Specification validity rule status description.
start_date	Start date.
end_date	End date.
sampling_plan_id	Sampling plan ID.
sampling_plan_name	Sampling plan name.
sample_inv_trans_ind	Indicates an inventory transaction is created for the sample amount. Valid values are Y - Yes, Null - No.
lot_optional_on_sample	Lot optional on sample.
coa_type	Certificate of Analysis type.
coa_at_ship_ind	Certificate of Analysis shipment indicator.
coa_at_invoice_ind	Certificate of Analysis invoice indicator.
coa_req_from_supl_ind	Certificate of Analysis from supplier.
delete_mark	Logical delete record indicator.
text_code	Text code.
attribute_category	Descriptive Flexfield Segment Category.
attribute1-30	Descriptive Flexfield Segments.
creation_date	Standard Who Column.
created_by_id	Standard Who Column.
created_by_name	Standard Who Column.
last_update_date	Standard Who Column.
last_updated_by_id	Standard Who Column.
last_updated_by_name	Standard Who Column.
last_update_login	Standard Who Column.

Parameter - x_wip_spec_vrs

This is a table type parameter that returns the fetched work in process specification validity rule information.

Parameter	Description
spec_vr_id	Specification validity rule identifier.
orgn_code	Organization code.
batch_id	Batch ID.
batch_no	Batch number.
recipe_id	Recipe ID.
recipe_no	Recipe number.
recipe_version	Recipe version.
formula_id	Formula ID.
formula_no	Formula number.
formula_vers	Formula version.
routing_id	Routing ID.
routing_no	Routing number.
routing_vers	Routing version.
step_id	Step ID.
step_no	Step number.
oprn_id	Operation ID.
oprn_no	Operation number.
oprn_vers	Operation version.
charge	Charges.
spec_vr_status	Specification validity rule status.
spec_vr_status_ description	Specification validity rule status description.
start_date	Start date.
end_date	End date.
sampling_plan_id	Sampling plan ID.
sampling_plan_name	Sampling plan name.
sample_inv_trans_ind	Indicates an inventory transaction is created for the sample amount. Valid values are Y - Yes, Null - No.

Parameter	Description
lot_optional_on_sample	Lot optional on sample.
delayed_lot_entry	Delayed lot entry.
auto_sample_ind	Auto sample indicator.
control_lot_attrib_ind)	Valid values are A - Auto, W - WorkFlow, Null - No
out_of_spec_lot_status	Outside specification lot status.
in_spec_lot_status	In specification lot status.
coa_type	Certificate of Analysis type.
control_batch_step_ind	Control batch step indicator.
coa_at_ship_ind	Certificate of Analysis shipment indicator.
coa_at_invoice_ind	Certificate of Analysis invoice indicator.
coa_req_from_supl_ind	Certificate of Analysis from supplier.
delete_mark	Logical delete record indicator.
text_code	Text code.
attribute_category	Descriptive Flexfield Segment Category.
attribute1-30	Descriptive Flexfield Segments.
creation_date	Standard Who Column.
created_by_id	Standard Who Column.
created_by_name	Standard Who Column.
last_update_date	Standard Who Column.
last_updated_by_id	Standard Who Column.
last_updated_by_name	Standard Who Column.
last_update_login	Standard Who Column.

Parameter - x_supl_spec_vrs

This is a table type parameter that returns the fetched supplier specification validity rule information.

Parameter	Description
po_header_id	Purchase order header identification.

Parameter	Description
po_number	Purchase order number.
spec_vr_id	Specification validity rule identifier.
orgn_code	Organization code.
supplier_id	Supplier ID.
supplier_name	Supplier name.
supplier_site_id	Supplier site ID.
supplier_site_name	Supplier site name.
po_line_id	Purchase order ID.
po_line_number	Purchase order line number.
spec_vr_status	Specification validity rule status.
spec_vr_status_ description	Specification validity rule status description.
start_date	Start date.
end_date	End date.
sampling_plan_id	Sampling plan ID.
sampling_plan_name	Sampling plan name.
sample_inv_trans_ind	Indicates an inventory transaction is created for the sample amount. Valid values are Y - Yes, Null - No.
lot_optional_on_sample	Lot optional on sample.
delayed_lot_entry	Delayed lot entry.
auto_sample_ind	Auto sample indicator.
coa_type	Certificate of Analysis type.
coa_at_ship_ind	Certificate of Analysis shipment indicator.
coa_at_invoice_ind	Certificate of Analysis invoice indicator.
coa_req_from_supl_ind	Certificate of Analysis supplier indicator.
delete_mark	Logical delete record indicator.
text_code	Text code.
attribute_category	Descriptive Flexfield Segment Category.
attribute1-30	Descriptive Flexfield Segments.

Parameter	Description
creation_date	Standard Who Column.
created_by_id	Standard Who Column.
created_by_name	Standard Who Column.
last_update_date	Standard Who Column.
last_updated_by_id	Standard Who Column.
last_updated_by_name	Standard Who Column.
last_update_login	Standard Who Column.

Parameter - x_inv_spec_vrs

This is a table type parameter that returns the fetched inventory specification validity rule information.

Parameter	Description
spec_vr_id	Specification validity rule ID.
orgn_code	Organization code.
lot_id	Lot ID.
lot_number	Lot number.
sublot_number	Sublot number
whse_code	Warehouse code.
location	Location
spec_vr_status	Specification validity rule status.
spec_vr_status_ description	Specification validity rule status description.
start_date	Start date.
end_date	End date.
sampling_plan_id	Sampling plan ID.
sampling_plan_name	Sampling plan name.
sample_inv_trans_ind	Indicates an inventory transaction is created for the sample amount. Valid values are Y - Yes, Null - No.
control_lot_attrib_ind	Valid values are A - Auto, W - WorkFlow, Null - No.

Parameter	Description
lot_optional_on_sample	Lot optional on sample.
delay_lot_entry	Delayed lot entry.
auto_sample_ind	Auto sample indicator.
in_spec_lot_status	In specification lot status.
out_of_spec_lot_status	Outside specification lot status.
control_batch_step_ind	Control batch step indicator.
coa_type	Certificate of Analysis type.
coa_at_ship_ind	Certificate of Analysis shipment indicator.
coa_at_invoice_ind	Certificate of Analysis invoice indicator.
coa_req_from_supl_ind	Certificate of Analysis suppler indicator.
delete_mark	Logical delete record indicator.
text_code	Text code.
attribute_category	Descriptive Flexfield Segment Category.
attribute1-30	Descriptive Flexfield Segments.
creation_date	Standard Who Column.
created_by_id	Standard Who Column.
created_by_name	Standard Who Column.
last_update_date	Standard Who Column.
last_updated_by_id	Standard Who Column.
last_updated_by_name	Standard Who Column.
last_update_login	Standard Who Column.

Parameter - x_mon_spec_vrs

This is a table type parameter that returns the fetched inventory specification validity rule information.

Parameter	Description
spec_vr_id	Specification validity rule indicator.
spec_id	Specification ID.

Parameter	Description
rule_type	Rule type.
loct_orgn_code	Location organization code.
whse_code	Warehouse code.
location	Location.
resources	Resources.
resource_orgn_code	Resource organization code.
resource_instance_id	Resource instance ID.
spec_vr_status	Specification validity rule status.
spec_vr_status_ description	Specification validity rule status description.
start_date	Start date.
end_date	End date.
sampling_plan_id	Sampling plan ID.
sampling_plan_name	Sampling plan name.
delete_mark	Logical delete record indicator.
text_code	Text code.
attribute_category	Descriptive Flexfield Segment Category.
attribute1-30	Descriptive Flexfield Segments.
creation_date	Standard Who Column.
created_by	Standard Who Column.
last_updated_by	Standard Who Column.
last_update_date	Standard Who Column.
last_update_login	Standard Who Column.

Fetch Sample Groups

The Fetch Sample Groups API retrieves sample groups from the GMD_SAMPLING_EVENTS and GMD_EVENT_SPEC_DISP tables.

Following is the definition of Fetch Sample Groups:

Parameter	Required	In/Out	Description
p_user_name	Y	IN	Oracle Applications user name.
p_orgn_code	N	IN	Organization code.
p_from_sample_no	N	IN	Starting sample number.
p_to_sample_no	N	IN	Ending sample number.
p_sample_event_id	N	IN	Sample event identifier.
p_from_item_no	N	IN	Starting item number.
p_to_item_no	N	IN	Ending item number.
p_item_id	N	IN	Item identifier.
p_from_lot_no	N	IN	Starting lot number.
p_to_lot_no	N	IN	Ending lot number.
p_from_sublot_no	N	IN	Starting sublot number.
p_to_sublot_no	N	IN	Ending sublot number.
p_lot_id	N	IN	Lot identifier.
p_disposition	N	IN	Disposition.
p_whse_code	N	IN	Warehouse code.
p_location	N	IN	Location.
p_location_id	N	IN	Location identifier.
p_wip_orgn_code	N	IN	Work in process organization code.
p_wip_batch_no	N	IN	Work in process batch number.
p_wip_batch_id	N	IN	Work in process batch identifier.
p_wip_recipe_no	N	IN	Work in process recipe number.
p_wip_recipe_vers	N	IN	Work in process recipe version.
p_wip_recipe_id	N	IN	Work in process recipe identifier.

Parameter	Required	In/Out	Description
p_wip_formula_no	N	IN	Work in process formula number.
p_wip_formula_vers	N	IN	Work in process formula version.
p_wip_formula_id	N	IN	Work in process formula identifier.
p_wip_formulaline_no	N	IN	Work in process formula line number.
p_wip_formulaline_id	N	IN	Work in process formula line identifier.
p_wip_formulaline_type	N	IN	Work in process formula line type.
p_wip_routing_no	N	IN	Work in process routing number.
p_wip_routing_vers	N	IN	Work in process routing version.
p_wip_routing_id	N	IN	Work in process routing identifier.
p_wip_step_no	N	IN	Work in process step number.
p_wip_step_id	N	IN	Work in process step identifier.
p_wip_oprn_no	N	IN	Work in process operation number.
p_wip_oprn_vers	N	IN	Work in process operation version.
p_wip_oprn_id	N	IN	Work in process operation identifier.
p_customer	N	IN	Customer.
p_customer_id	N	IN	Customer identifier.
p_customer_org_id	N	IN	Customer organization identifier.
p_customer_ship_to_location	N	IN	Customer ship to location.
p_customer_ship_to_location_id	N	IN	Customer ship to location identifier.
p_customer_order_number	N	IN	Customer order number.
p_customer_order_id	N	IN	Customer order identifier.
p_customer_order_type	N	IN	Customer order type.

Parameter	Required	In/Out	Description
p_customer_order_line	N	IN	Customer order line.
p_customer_order_line_id	N	IN	Customer order line identifier.
p_supplier	N	IN	Supplier.
p_supplier_id	N	IN	Supplier identifier.
p_supplier_site	N	IN	Supplier site.
p_supplier_po_number	N	IN	Supplier purchase order number.
p_supplier_po_id	N	IN	Supplier purchase order identifier.
p_supplier_po_line	N	IN	Supplier purchase order line.
p_supplier_po_line_id	N	IN	Supplier purchase order line identifier.
p_delete_mark	N	IN	Logical delete record indicator.
p_from_last_update_date	N	IN	Starting last update date.
p_to_last_update_date	N	IN	Ending last update date.
p_sg_orgn_code	N	IN	Organization code.
p_resources	N	IN	Resources.
p_instance_id	N	IN	Instance identifier.
p_ss_id	N	IN	Stability study identifier.
p_ss_orgn_code	N	IN	Stability study organization code.
p_ss_no	N	IN	Stability study number.
p_variant_id	N	IN	Variant identifier.
p_variant_no	N	IN	Variant number.
p_time_point_id	N	IN	Time point identifier.
x_sample_groups_table	Y	OUT	Record type gmd_sampling_events_tab_type.

Parameter - x_sample_groups_table (OUT)

This is a row type parameter that returns the header information for the fetched sample groups.

Parameter	Description
receipt_id	Receipt Identification.
receipt_number	Receipt number.
po_header_id	Purchase order header ID.
po_number	Purchase order number.
sampling_event_id	Sampling event ID.
original_spec_vr_id	Original specification validity rule ID.
disposition	Final result value.
sample_req_cnt	Required number of samples to be taken.
sample_taken_cnt	Current number of samples taken.
sampling_plan_id	Sampling plan ID.
event_type_code	Identifies the event triggering a sample to be taken. For example, Batch or Lot Expiration.
event_id	Identifier of event triggering a sample to be taken.
item_id	Item ID.
item_no	Item number.
lot_id	Lot ID.
lot_no	Lot number.
sublot_no	Sublot number.
whse_code	Warehouse code.
location	Location.
batch_id	Batch ID.
batch_no	Batch number.
recipe_id	Recipe ID.
recipe_number	Recipe number.
recipe_vers	Recipe version.

Parameter	Description
formula_id	Formula ID.
formula_no	Formula number.
formula_vers	Formula version.
formulaline_id	Formula line ID.
formulaline_no	Formula line number.
routing_id	Routing ID.
oprn_id	Operation ID.
charge	Charge.
cust_id	Customer ID.
cust_name	Customer name.
order_id	Order ID.
order_number	Order number.
order_line_id	Order line ID.
order_line_no	Order line number.
org_id	Organization ID.
organization_code	Organization code.
supplier_id	Supplier ID.
supplier_name	Supplier name.
po_line_id	Purchase order line ID.
po_line_number	Purchase order line number.
receipt_line_id	Receipt line ID.
receipt_line_number	Receipt line number.
supplier_lot_no	Supplier lot number.
complete_ind	Indicates if all samples for this specification are complete.
sample_id_to_evaluate	Sample identifier to evaluate.
composite_id_to_evaluate	Composite ID to evaluate.
text_code	Text code.

Parameter	Description
creation_date	Standard Who Column.
created_by_id	Standard Who Column.
created_by_name	Standard Who Column.
last_updated_by_id	Standard Who Column.
last_updated_by_name	Standard Who Column.
last_update_date	Standard Who Column.
last_update_login	Standard Who Column.
supplier_site_id	Supplier site ID.
supplier_site_name	Supplier site name.
ship_to_site_id	Ship-to site ID.
ship_to_site_name	Ship-to site name.
step_id	Step ID.
step_no	Step number.
lot_retest_ind	Lot retest indicator.
recomposite_ind	Recomposite idicator.
sample_active_cnt	Number of active samples taken, which are samples not marked for purge and not canceled.
orgn_code	Organization code.
resources	Resources.
instance_id	Instance ID.
time_point_id	Time point ID.
variant_id	Variant ID.
archived_taken	Archive taken.
reserved_taken	Reserved taken.
x_sample_event_disps	Sample event disposition.

Parameter - x_sample_events_disp

This is a row type parameter that returns the event disposition information for a fetched sample.

Parameter	Description
event_spec_disp_id	Event specification disposition ID.
spec_id	Specification ID.
spec_name	Specification name.
spec_vers	Specification version.
spec_vr_id	Specification validity rule ID.
disposition	Disposition.
spec_used_for_lot_ attrib_ind	Specification used for lot attribute indicator.
delete_mark	Logical delete record indicator.
creation_datedate	Standard Who Column.
created_by_id	Standard Who Column.
created_by_name	Standard Who Column.
last_update_date	Standard Who Column.
last_updated_by_id	Standard Who Column.
last_updated_by_name	Standard Who Column.
last_update_login	Standard Who Column.

Fetch Samples

The Fetch Samples API retrieves sample groups from the GMD_SAMPLES and GMD_SAMPLE_SPEC_DISP tables.

Following is the definition of Fetch Samples:

Parameter	Required	In/Out	Description
p_user_name	Y	IN	Oracle Applications user name.
p_orgn_code	N	IN	Organization code.
p_from_sample_no	N	IN	Starting sample number.
p_to_sample_no	N	IN	Ending sample number.
p_sample_id	N	IN	Sample identifier.
p_sampling_event_id	N	IN	Sampling event identifier.
p_from_item_no	N	IN	Starting item number.
p_to_item_no	N	IN	Ending item number.
p_item_id	N	IN	Item identifier.
p_from_lot_no	N	IN	Starting lot number.
p_to_lot_no	N	IN	Ending lot number.
p_sublot_no	N	IN	Sublot number.
p_lot_id	N	IN	Lot identifier.
p_priority	N	IN	Priority.
p_spec_name	N	IN	Specification name.
p_spec_vers	N	IN	Specification version.
p_spec_id	N	IN	Specification identifier.
p_source	N	IN	Source.
p_from_date_drawn	N	IN	Starting date drawn.
p_to_date_drawn	N	IN	Ending date drawn.
p_from_expiration_date	N	IN	Starting expiration date.

Parameter	Required	In/Out	Description
p_to_expiration_datedate	N	IN	Ending expiration date.
p_source_whse	N	IN	Source warehouse.
p_source_location	N	IN	Source location.
p_grade	N	IN	Grade.
p_sample_disposition	N	IN	Sample disposition.
p_storage_whse	N	IN	Storage warehouse.
p_storage_location	N	IN	Storage location.
p_qc_lab_orgn_code	N	IN	Quality lab organization code.
p_external_id	N	IN	External identifier.
p_sample	N	IN	Sample.
p_lot_retest_ind	N	IN	Lot retest indicator.
p_whse_code	N	IN	Warehouse code.
p_location	N	IN	Location.
p_location_id	N	IN	Location identifier.
p_wip_plant_code	N	IN	Work in process plan code.
p_wip_batch_no	N	IN	Work in process batch number.
p_wip_batch_id	N	IN	Work in process batch identifier.
p_wip_recipe_no	N	IN	Work in process recipe number.
p_wip_recipe_version	N	IN	Work in process recipe version.
p_wip_recipe_id	N	IN	Work in process recipe identifier.
p_wip_formula_no	N	IN	Work in process formula number.
p_wip_formula_version	N	IN	Work in process formula version.
p_wip_formula_id	N	IN	Work in process formula identifier.
p_wip_formulaline	N	IN	Work in process formula line.

Parameter	Required	In/Out	Description
p_wip_ formulaline_id	N	IN	Work in process formula line identifier.
p_wip_line_type	N	IN	Work in process line type.
p_wip_routing_no	N	IN	Work in process routing number.
p_wip_routing_ vers	N	IN	Work in process routing version.
p_wip_routing_id	N	IN	Work in process routing identifier.
p_wip_batchstep_ no	N	IN	Work in process batch step number.
p_wip_batchstep_ id	N	IN	Work in process batch step identifier.
p_wip_oprn_no	N	IN	Work in process operation number.
p_wip_oprn_vers	N	IN	Work in process operation version.
p_wip_oprn_id	N	IN	Work in process operation identifier.
p_cust_name	N	IN	Customer name.
p_cust_id	N	IN	Customer identifier.
p_cust_org_id	N	IN	Customer organization identifier.
p_cust_ship_to_ site_id	N	IN	Customer ship to site identifier.
p_cust_order	N	IN	Customer order.
p_cust_order_id	N	IN	Customer order identifier.
p_cust_order_type	N	IN	Customer order type.
p_cust_order_line	N	IN	Customer order line.
p_cust_order_line_ id	N	IN	Customer order line identifier.
p_supplier	N	IN	Supplier.
p_supplier_id	N	IN	Supplier identifier.
p_supplier_site_id	N	IN	Supplier site identifier.
p_supplier_po	N	IN	Supplier purchase order.
p_supplier_po_id	N	IN	Supplier purchase order identifier.

Parameter	Required	In/Out	Description
p_supplier_po_line	N	IN	Supplier purchase order line.
p_supplier_po_line_id	N	IN	Supplier purchase order line identifier.
p_from_date_received	N	IN	Starting date received.
p_to_date_received	N	IN	Ending date received.
p_from_date_required	N	IN	Starting date required.
p_to_date_required	N	IN	Ending date required.
p_resources	N	IN	Resources.
p_instance_id	N	IN	Instance identifier.
p_from_retrieval_date	N	IN	Starting retrieval date.
p_to_retrieval_date	N	IN	Ending retrieval date.
p_sample_type	N	IN	Sample type.
p_ss_id	N	IN	Stability study identifier.
p_ss_orgn_code	N	IN	Stability study organization code.
p_ss_no	N	IN	Stability study number.
p_variant_id	N	IN	Variant identifier.
p_variant_no	N	IN	Variant number.
p_time_point_id	N	IN	Time point identifier.
p_from_last_update_date	N	IN	Starting last update date.
p_to_last_update_date	N	IN	Ending last update date.
p_retain_as	N	IN	Retain as.
p_delete_mark	N	IN	Logical delete record indicator.
x_samples_table	Y	OUT	Record type samples_table.

Parameter - x_samples_table (OUT)

This is a table type parameter that returns the header information for the fetched samples.

Parameter	Description
sampling_event_id	Sampling event ID.
step_no	Step number.
step_id	Step ID.
sample_id	Sample unique identifier.
sample_no	Sampling number.
sample_desc	Sampling description.
spec_id	Specification ID.
spec_name	Specification name.
spec_vers	Specification version.
qc_lab_orgn_code	Quality lab organization.
item_id	Item ID.
item_no	Item number.
location	Warehouse location when a sample is taken from a warehouse.
expiration_date	Samples expiration date.
lot_id	Lot ID.
lot_no	Items lot number of sample.
sublot_no	Sublot number.
batch_id	Batch ID.
plant_code	Plant code.
batch_no	Batch number.
recipe_id	Recipe ID.
recipe_no	Recipe number.
recipe_version	Recipe version.
formula_id	Formula ID.
formula_no	Formula number.

Parameter	Description
formula_vers	Formula version.
formulaline_id	Formula line ID.
formulaline_no	Formula line number.
line_type	Line type.
routing_id	Routing ID.
routing_no	Routing number.
routing_vers	Routing version.
oprn_id	Operation ID.
oprn_no	Operation number.
oprn_vers	Operation version.
batchstep_id	Batch step ID.
batchstep_no	Batch step number.
charge	Charge.
cust_id	Customer ID.
cust_name	Customer name.
order_id	Order ID.
order_type	Order type.
order_number	Order number.
order_line_id	Order line ID.
order_line_number	Order line number.
org_id	Organization ID.
org_name	Organization name.
supplier_id	Supplier ID.
supplier_no	Supplier number.
sample_qty	Sample quantity.
sample_uom	Sample unit of measure.
source	Source.
sampler_id	Sampler ID.

Parameter	Description
sampler	Sampler.
date_drawn	Date the sample was drawn.
source_comment	Source comments.
storage_whse	Storage warehouse.
storage_location	Storage location.
external_id	External ID.
sample_approver_id	Sample approver ID.
inv_approver_id	Inventory approver ID.
priority	Priority.
sample_inv_trans_ind	Sample inventory transaction indicator.
supplier_site_id	Supplier site ID.
supplier_site	Supplier site.
whse_code	Warehouse code.
orgn_code	Organization code.
po_header_id	Purchase order header identification.
po_number	Purchase order number.
po_line_id	Purchase order line identification.
po_line_number	Purchase order line number.
receipt_id	Receipt ID.
receipt_no	Receipt number.
receipt_line_id	Receipt line ID.
receipt_line	Receipt line.
sample_disposition	Sample disposition
ship_to_site_id	Ship-to site ID.
ship_to_location	Ship-to location.
supplier_lot_no	Supplier lot ID.
lot_retest_ind	Lot retest indicator.
sample_instance	Current instance of the sample created.

Parameter	Description
source_whse	Warehouse where the sample quantity must be stored.
source_location	Location where the sample quantity must be stored.
date_received	The date the same was received.
date_required	The date the same was required.
resources	Resources.
instance_id	Instance ID.
retrieval_date	Retrieval date for the sample.
sample_type	Sample type.
time_point_id	Time point ID.
variant_id	Variant ID.
delete_mark	Logical delete record indicator.
text_code	Text code.
attribute_category	Descriptive Flexfield Segment Category
attribute1-30	Descriptive Flexfield Segments
creation_date	Standard Who Column
created_by	Standard Who Column
created_by_name	Standard Who Column
last_update_date	Standard Who Column
last_updated_by	Standard Who Column
last_updated_by_name	Standard Who Column
last_update_login	Standard Who Column
retain_as	Retain as.
remaining_qty	Remaining quantity of the sample.

Fetch Results

The Fetch Results API retrieves sample groups from the GMD_RESULTS table.

Following is the definition of Fetch Results:

Field	Required	In/Out	Description
p_user_name	Y	IN	Oracle Applications user name.
p_orgn_code	N	IN	Organization code.
p_from_sample_no	N	IN	Starting sample number.
p_to_sample_no	N	IN	Ending sample number.
p_sample_id	N	IN	Sample identifier.
p_from_result_date	N	IN	Starting result date.
p_to_result_date	N	IN	Ending result date.
p_sample_disposition	N	IN	Sample disposition.
p_in_spec_ind	N	IN	In specification indicator.
p_qc_lab_orgn_code	N	IN	Lab organization code.
p_evaluation_ind	N	IN	Evaluation indicator.
p_tester	N	IN	Tester.
p_tester_id	N	IN	Test identifier.
p_test_provider_id	N	IN	Test provider identifier.
p_delete_mark	N	IN	Logical delete record indicator.
p_from_last_update_date	N	IN	Starting last update date.
p_to_last_update_date	N	IN	Ending last update date.
p_planned_resource	N	IN	Planned resource.
p_planned_resource_instance	N	IN	Planned resource instance.
p_actual_resource	N	IN	Actual resource.
p_actual_resource_instance	N	IN	Actual resource instance.

Field	Required	In/Out	Description
p_from_planned_ result_date	N	IN	Starting planned result date.
p_to_planned_ result_date	N	IN	Ending planned result date.
p_from_test_by_ date	N	IN	Starting test by date.
p_to_test_by_date	N	IN	Ending test by date.
p_reserve_sample_ id	N	IN	Reserve sample identifier.
x_results_table	Y	OUT	Record type results table.

Parameter - x_results_table (OUT)

This is a table type parameter that returns the fetched sample results.

Parameter	Description
update_instance_id	Used to store e-signature instance identification.
result_id	Results unique identifier.
sample_id	Identifies the sample the test results is recorded.
sample_no	Sample number.
test_id	Test performed the results are recorded.
test_code	Test code.
test_replicate_cnt	Number of times test was performed on the same sample.
qc_lab_orgn_code	Quality lab organization code where the test was performed.
result_value_num	Numeric result value.
result_date	Date result occurred.
test_kit_item_id	Item ID of solvent used in the testing procedure.
test_kit_item_no	Item number of solvent used in the testing procedure.
test_kit_lot_no	Solvent lot number.
test_kit_sublot_no	Solvent sublot number.
tester	User ID of person who performed the test.

Parameter	Description
tester_id	Tester ID.
test_provider_id	Test provider ID.
ad_hoc_print_on_coa_ind	Ad hoc printing on Certificate of Analysis indicator.
seq	Sequence.
result_value_char	Character result value.
test_provider_code	Test provider code.
assay_retest	Assay retest.
in_spec_ind	In specification indicator.
disposition	Disposition.
evaluation_ind	Evaluation indicator.
planned_resource	Planned resource.
planned_resource_instance	Planned resource instance.
actual_resource	Actual resource.
actual_resource_instance	Actual resource instance.
planned_result_date	Planned result date.
test_by_date	Test by date.
delete_mark	Logical delete record indicator.
text_code	Text code.
attribute_category	Descriptive Flexfield Segment Category
attribute1-30	Descriptive Flexfield Segments
creation_date	Standard Who Column
created_by	Standard Who Column
created_by_name	Standard Who Column
last_updated_by	Standard Who Column
last_updated_by_name	Standard Who Column
last_update_date	Standard Who Column
last_update_login	Standard Who Column

Parameter	Description
test_qty	Test quantity.
test_uom	Test unit of measure.
reserve_sample_id	Reserve sample indicator.
consumed_qty	Consumed quantity.
parent_result_id	Parent result ID.
test_method_id	Test method ID.

Fetch Composite Results

The Fetch Composite Results API retrieves composite results from the GMD_COMPOSITE_RESULTS, GMD_COMPOSITE_SPEC_DISP, and GMD_COMPOSITE_RESULT_ASSOC tables.

Following is the definition of Fetch Composite Results:

Parameter	Required	In/Out	Description
p_user_name	Y	IN	Oracle Applications user name.
p_sampling_event_id	N	IN	Sampling event identifier.
p_composite_result_disposition	N	IN	Composite result disposition.
p_from_item_no	N	IN	Starting item number.
p_to_item_no	N	IN	Ending item number.
p_from_lot_id	N	IN	Starting lot identifier.
p_to_lot_id	N	IN	Ending lot identifier.
p_from_last_update_date	N	IN	Starting last update date.
p_to_last_update_date	N	IN	Ending last update date.
p_delete_mark	N	IN	Logical delete record indicator.
x_composite_results_table	Y	OUT	Record type composite_results_table.

Parameter - x_composite_results_tbl (OUT)

This is a row type parameter fetched from gmd_composite_results, gmd_composite_spec_disp, and gmd_composite_result_assoc.

Parameter	Description
composite_result_id	Composite results.
test_id	Test Identifier.
mean	Statistical mean.
mode_num	Mode number.
mode_char	Mode character.

Parameter	Description
low_num	Lower value.
high_num	Higher numeric range.
range	Statistical range.
standard_deviation	Statistics standard deviation.
sample_total	Sample total.
sample_cnt_used	Sample count used.
non_validated_result	Non-validated result.
high_char	Highest character value.
low_char	Lowest character value.
median_char	Median character value.
median_num	Median numeric value.
composite_spec_disp_id	Composite specification disposition identification.
in_spec_ind	Composite specification indicator.
wf_response	Workflow response.
value_in_report_ precision	Report precision.
disposition	Disposition.
item_id	Item ID.
item_no	Item number.
lot_id	Lot ID.
lot_no	Lot number.
sublot_no	Sublot number.
delete_mark	Logical delete record indicator.
text_code	Text code.
attribute_category	Descriptive Flexfield Segment Category.
attribute1-30	Descriptive Flexfield Segments.
creation_date	Standard Who Column.
created_by	Standard Who Column.

Parameter	Description
created_by_name	Standard Who Column.
last_update_date	Standard Who Column.
last_updated_by	Standard Who Column.
last_updated_by_name	Standard Who Column.
last_update_login	Standard Who Column.
parent_composite_ result_id	Parent composite result ID.

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:

- 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 the following 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 Quality Management API error messages:

Error Messages	Message Code
Supplier validity rule exists	GMD_SUPP_VR_EXIST
Error &ERROR occurred in Package &PACKAGE at &POSITION	GMD_API_ERROR
API failure while processing record for table &l_table_name identified by column & column_name of value &l_key_value	GMD_API_RECORD_IDENTIFIER
Batch not found	GMD_BATCH_NOT_FOUND
Batch step not found	GMD_BATCH_STEP_NOT_FOUND
Customer validity rule exists	GMD_CUST_VR_EXIST'
Customer not found	GMD_CUSTOMER_NOT_FOUND

Error Messages	Message Code
Customer Required	GMD_CUSTOMER_REQD
Event Spec disp id is Required.	GMD_EVENT_SPEC_DISP_NULL
Event Spec Record not found for event_spec_disp_id &event_disp	GMD_EVENT_SPEC_NOTFOUND
Failure to delete record in table %l_table_name identified by column %l_column_name of value %l_key_value	GMD_FAILED_TO_DELETE_ROW
Failure to fetch record in table %l_table_name identified by column %l_column_name of value %l_key_value	GMD_FAILED_TO_FETCH_ROW
Failure to insert record in table %l_table_name identified by column %l_column_name of value %l_key_value	GMD_FAILED_TO_INSERT_ROW
Formula line not found	GMD_FORMULA_LINE_NOT_FOUND
Formula not found	GMD_FORMULA_NOT_FOUND
GMD_INVALID_OPERATION	GMD_INVALID_OPERATION
Absolute experimental Error Value cannot be greater than &MAX_VAL	GMD_INVALID_SPEC_VAL_NUM
INVALID test type &test_type for test values record	GMD_INVALID_TEST_TYPE
Invalid user	GMD_INVALID_USER_NAME
Item lot not found	GMD_ITEM_LOT_NOT_FOUND
Item sublot not found	GMD_ITEM_SUBLOT_NOT_FOUND
Failure to lock record in table %l_table_name identified by column %l_column_name of value %l_key_value	GMD_LOCKING_FAILURE
Location not found	GMD_LOCT_NOT_FOUND
No data has been found while retrieving the row from the table &TABLE_NAME.	GMD_NO_DATA_FOUND
No keys have been specified to identify the row in the table &TABLE_NAME.	GMD_NO_KEYS

Error Messages	Message Code
Event specification record does not exist for spec_id &spec_id and sampling_event &samp_event	GMD_NO_SPEC_EVENT_FOUND
For WIP sample, at least Batch No or Recipe ID is required	GMD_NO_WIP_PARAM
Order line not found	GMD_ORDER_LINE_NOT_FOUND
Order not found	GMD_ORDER_NOT_FOUND
Organization not found	GMD_ORG_NOT_FOUND'
Orgn Code not found	GMD_ORGN_CODE_NOT_FOUND
Purchase Order Line not found	GMD_PO_LINE_NOT_FOUND
PO not found	GMD_PO_NOT_FOUND'
QC Lab orgn code not found	GMD_QC_LAB_ORGN_CODE_NOT_FOUND
QC Lab orgn code required	GMD_QC_LAB_ORGN_CODE_REQD
Recipe not found	GMD_RECIPE_NOT_FOUND
Record is already delete marked - see table %l_table_name identified by column %l_column_name of value %l_key_value	GMD_RECORD_DELETE_MARKED
Record locked	GMD_RECORD_LOCKED
Report precision cannot be greater then Display Precision	GMD_REP_GRTR_DIS_PRCSN
Could not find matching result char value.	GMD_RESULT_CHAR_NOTFOUND
Result record not found for sample_id &sample_id, test_id &test_id and test_replicate_cnt value &test_rep_cnt	GMD_RESULT_NOT_FOUND
Result value char required for Results	GMD_RESULT_VAL_CHAR_REQD
Result value num required for Results	GMD_RESULT_VAL_NUM_REQD
Routing not found	GMD_ROUTING_NOT_FOUND
Routing Step not found	GMD_ROUTING_STEP_NOT_FOUND
Invalid sample disposition	GMD_SAMPLE_DISPOSITION_INVALID
Sample already exists	GMD_SAMPLE_EXIST

Error Messages	Message Code
Sample record does not exists for sample_no &sample_no and orgn_code &orgn_code	GMD_SAMPLE_NOT_FOUND
Sample number required	GMD_SAMPLE_NUMBER_REQD
Sample requires a Orgn Code to be defined	GMD_SAMPLE_ORGN_CODE_REQD
Sample quantity required	GMD_SAMPLE_QTY_REQD
Invalid source &source defined for sample	GMD_SAMPLE_SOURCE_INVALID
Sample UOM required	GMD_SAMPLE_UOM_REQD
Sampling event missing	GMD_SAMPLING_EVENT_MISSING
Sampling event not found	GMD_SAMPLING_EVENT_NOT_FOUND
Ship to not found	GMD_SHIP_TO_NOT_FOUND
Reference test(s) used in the expression test is missing in the specification	GMD_SOME_REF_TESTS_MISSING
Cannot define Specification error region.	GMD_SPEC_ERROR_REG_NOT_APPL
SPEC_ID must be supplied	GMD_SPEC_ID_REQUIRED
Specification Item not found.	GMD_SPEC_ITEM_NOT_FOUND
Specification Item required.	GMD_SPEC_ITEM_REQD
Specification Name required.	GMD_SPEC_NAME_REQD
Could not find matching Specification Record	GMD_SPEC_NOT_FOUND
Specification Owner Organization &ORGN not found.	GMD_SPEC_ORGN_NOT_FOUND
Specification Owner Organization Code required.	GMD_SPEC_ORGN_REQD
Specification Owner not found.	GMD_SPEC_OWNER_NOT_FOUND
Specification Owner required.	GMD_SPEC_OWNER_REQD
Spec status prevents updates	GMD_SPEC_STATUS_BLOCKS_UPDATE
Specification status &STATUS not found.	GMD_SPEC_STATUS_NOT_FOUND
Specification Status required.	GMD_SPEC_STATUS_REQD
Spec Test ID required	GMD_SPEC_TEST_ID_REQUIRED

Error Messages	Message Code
Test Sequence &SEQ already exists on the current specification.	GMD_SPEC_TEST_SEQ_EXIST
Specification &SPEC Version &VERS exists.	GMD_SPEC_VERS_EXIST
Specification Version must be a whole positive number.	GMD_SPEC_VERS_INVALID
Specification Version required	GMD_SPEC_VERS_REQD
There is an error in effectivity date of spec validity rule	GMD_SPEC_VR_EFF_DATE_ERROR
SPEC_VR_ID must be supplied	GMD_SPEC_VR_ID_REQUIRED
Effectivity start date is required for Validity rule.	GMD_SPEC_VR_START_DATE_REQD
Specification Validity is higher.	GMD_SPEC_VR_STATUS_HIGHER
Specification warehouse not found.	GMD_SPEC_WHSE_NOT_FOUND
Supplier not found	GMD_SUPPLIER_NOT_FOUND
Supplier required	GMD_SUPPLIER_REQD
Parameter : Test_id Table is empty.	GMD_TEST_ID_TABLE_EMPTY
Test not found	GMD_TEST_NOT_FOUND
User &OWNER does not have access to Organization &ORGN.	GMD_USER_ORGN_NO_ACCESS
Warehouse not found.	GMD_WHSE_NOT_FOUND
Insufficient data for WIP Validity Rule	GMD_WIP_VR_ALL_NULL
WIP validity rule exists	GMD_WIP_VR_EXIST

Index