Oracle Process Manufacturing (OPM) Migration Reference Guide Release 11*i* to Release 12

Table of Contents

This reference guide contains the following information:

Change Record

Date	Version - Description of Change
July 5, 2005	Version 1 - Created Document
September 29, 2005	Version 2 - Second Draft
November 1, 2005	Version 3 - Final Document
August 11, 2006	Version 4 - Edits made based on Internal Review
December 27, 2006	Version 5 - Addition of Section for Lot Status Migration scenarios

Oracle Corporation

Author and Date
Michele-Andrea Fields, July 2005, September 2005

Revisions and Date
Karen Theel, November 2005, August 2006, December 2006

Copyright Information

Disclaimer
This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Trademark Information

Introduction

The Oracle Process Manufacturing (OPM) application was developed to meet the needs of process manufacturers in industries such as pharmaceuticals, chemicals, food and beverage, metals, pulp and paper, and textiles. Prior to Release 12, this application had its own inventory management system, which maintained a central item master, tracked inventory in two simultaneous units of measure and sequenced unique lot numbers. Beginning with Release 12, these features have been added to Oracle Inventory and the Process Manufacturing application now relies on this core inventory system. As part of the installation of Release 12 and above, you migrate data from Oracle Process Inventory into Oracle Inventory.

About This Document

This document provides information on how Oracle Process Manufacturing data maps with the converged inventory model and how the migration runs. This document discusses:

- Performing the Data Migration and Release 12 Installation
- Using the Migration User Interface
- Understanding Data Migration and Transformation from OPM Inventory to Oracle Inventory
- Data Mapping

Business Needs

The primary focus of migration is to minimize system downtime, while streamlining the implementation of Oracle Inventory for process manufacturers. To this extent, the migration:

- **Lets you retain or change your organization structure.** Inventory convergence lets you fully embrace the Oracle Inventory organization structure. Prior to Release 12, Process Manufacturing does not support the concept of subinventories. Although a subinventory is created as part of Warehouse setup, inventory transactions are recorded at the warehouse level only. Subinventories are integral to the Oracle Inventory organization structure. All transactions are performed at this level. The migration process allows you to optionally map your warehouses as subinventories. You can keep the original warehouse as an active inventory organization, or to make it inactive. Alternatively you can keep warehouses as inventory organizations and define new subinventories associated with them.

- **Updates references to the new item master and organization master.** Internal IDs are used to reference the item number and inventory organization for a record. These new IDs are added to existing records to let you continue to query and update existing data.
- **Transfers inventory balances.** All onhand inventory balances are transferred to existing or new organizations, based on your pre-migration setup. You can review the balances in Oracle Inventory and review historical transactional data in OPM Inventory. Period end balances for the most recent inventory period are also migrated.
- **Transfers lot and sublot information.** Lot and sublot information is transferred from Process Manufacturing to Oracle Inventory lot tables.
- **Transfers other master data for lot status, quality grades, and costing.**
- **Transfer open production batches**

The Migration Process

The migration of data for convergence takes place in three phases – pre-migration, inline migration (during the Release 12 installation), and post-migration.
The pre-migration activities are done through a form built specifically for the migration. This lets you set up the data for the actual migration. A detailed explanation of this form is given in later sections of this document. The inline migration is the installation of Release 12. During this period, the actual migration of the data is completed. The post-migration section is where the data is transformed into the new data model, and final scripts are run to complete the movement of all data.

Pre-Migration

Pre-Migration Window

The pre-migration for Inventory Convergence consists of the validation and cleaning of data to ensure accuracy after the installation of Release 12. A patch must be applied to work with the new Convergence Migration window, which consists of tabs where you view your data and make changes either manually or through the system. Details for these windows are located in the following sections of this document.

Early Migration of Items, Orgs, Warehouses, and Addresses

If you are on Release 11.5.6 or higher, you can complete pre-migration steps to migrate Items, Organizations, Warehouses, and Addresses. After the data is ensured for accuracy in the migration window, you can run this migration prior to the blackout period. This saves time during the blackout, but the data is not completely transformed into Release 12 at this point. During the installation of Release 12, the final data transformation takes place to complete the process.

If you are on Release 11.5.5 or lower, you cannot complete this early migration. The Items, Orgs, Warehouses, and Addresses are migrated and transformed during the installation of Release 12.

Process Execution Batches

For Process Execution Batches, it is highly recommended that you review the detailed batch information and clean up as much erroneous data as possible BEFORE batches are processed for migration. If you are on on release 11.5.8 and below you must run the pre-migration validation script and correct any reported errors BEFORE batches are processed for migration. Refer to the Process Execution Migration document for details.

For all release levels, you must do the following before moving to Release 12:

- All completed batches must be closed.
- If you want to have pending, WIP, and FPO batches automatically recreated in Release 12, then you must complete the Process Batches for Migration snapshot procedure from the Batches tab of the Common Migration Setup form. Once this procedure has been run, you can no longer manipulate batches until after the migration to Release 12.
- You must run Subledger Update, GL Export, and Journal Import prior to migration. This ensures that all transactions related to closed batches have been properly booked. You cannot run these programs on batches closed in 11.5 after the migration to Release 12 has been completed.

Inventory Validation

There is an OPM Inventory validation script that must be run to ensure accuracy in your data. This validation script does the following:

- Displays all items in OPM that do not have the auto lot numbering set up in the validation report.
- Verifies the descriptive flexfield setup in both OPM Inventory and Oracle Inventory item master for duplicate attributes. In 11*i*, ic_item_mst had 30 descriptive flexfields and mtl_system_items had 15. OPM migrates flexfields from OPM Inventory to Oracle Inventory. In Release 12, there are a total of 30 flexfields on mtl_system_items. There are two scenarios that can happen. The first scenario is in OPM you used segments 1, 2, 3 and in Oracle Inventory you used 10, 11, 12. When the migration runs, the 10, 11, 12 in Oracle Inventory remain and 1, 2, 3 in OPM migrate to 1, 2, 3 in Oracle Inventory. The other scenario is when you have 1, 2, 3 in OPM Inventory and 1, 2, 3 in Oracle Inventory. This pre-validation script logs this as an error. Then you must go into one of the tables and either remove the duplicate or relocate one of them to another segment.

Quality Validation

If you are on on release 11.5.8 and below you must run the pre-migration validation script (5102439) and correct any reported errors BEFORE quality transactions are processed for migration. Refer to the Patch Read Me section for details.

In the Convergence Migration form, on the Quality tab, you must set the stability studies default organization code to a valid value. Before the blackout, you must run a validation script that verifies that this is set properly. If it is not, then the validation script displays an error, and you must go back into the form and complete this set up.

Cost Management and Manufacturing Accounting Controller (MAC) Validation

For Release 12, OPM Companies are migrated to Legal Entities. If multiple OPM Companies in Release 11i that are all pointing to the same Legal Entity you can, as part of the pre-migration configuration, choose to migrate these companies to their own legal entities. If you do not access the form before the migration, then all OPM Companies are migrated to the Legal Entities they currently point to.

Also, there is a Cost Management validation script you must run to ensure the accuracy of your data. This validation script does the following:

- Determines if there are any unposted inventory transactions or unposted resource transactions or unposted closed batches. If any are found, then a warning is raised. You must then verify and post the transactions by running the subledger update process. Though running the subledger is a mandatory step, this script validates that it was done. However, the script cannot automatically correct the situation.
- If actual costing is used for inventory valuation, then it looks for active transactions in the current open period in which you are migrating. If they are found, then a warning is raised that the actual costs may not be accurate for the current month once they migrate to Release 12 and run the actual cost at month end for the current open period. You must correct this situation prior to the migration.

Regulatory Management Validation

If you are using OPM Regulatory Management, then you must run this validation script. The script checks for that:

- the value of the profile GR: Default Warehouse is not NULL, as this value is used as the owner organization for migrated standalone Regulatory formulas.
- the value of the profile FM: Yield Type is not NULL, as this value is used as the unit of measure code for migrated standalone Regulatory formulas.
- each Regulatory Item has an associated CAS number, as it is a required field when the hazardous_material flag is set in mtl_system_items.
- all ingredients in the item concentrations table that already exist in Oracle Inventory can be converted to the product’s unit of measure.

The script returns a log file that displays any failure information. If anything in the script fails, then you must resolve the issue in Regulatory Management, and rerun the validation script until it comes back with no errors.

Order Fulfillment to Order Management

While the Release 12 install contains the migration of Order Fulfillment to Order Management, it is necessary to complete a series of pre-migration steps to ready the data for migration. These steps can be found in the Oracle Process Manufacturing Order Fulfillment to Order Management Migration documentation available on Oracle*Metalink*. These documents are supported by release, so you must select the appropriate version for your current implementation. Additional details are provided in this document in the Order Management Migration topic.

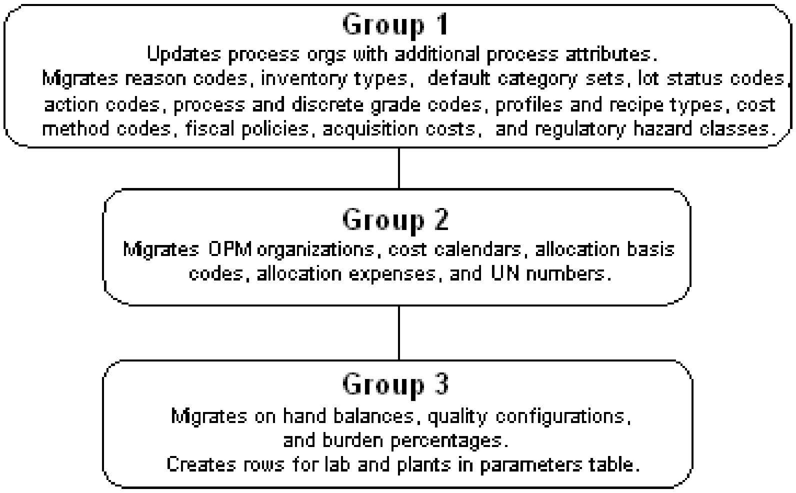
Process Forecast to Discrete Forecast

In R12, Forecasts will be defined in a common forecast table. Migration to this table will occur only if the OPM forecasts have a forecast set name. The Forecast Set Name field was added in 11.5.10.

If below 11.5.10, you will need to apply patch 4268525 and then enter the Forecast Set Name for all the forecasts that you want to migrate.

Release 12 Inline Migration

The migration is phased so that data moves in a logical order, and the migration scripts support this phasing. In the event that a migration does not migrate or transform the data properly, the previous scripts must be run again. The following flow details the phase in which the scripts run and the order within that phase:



Group 1

Updates process orgs with additional process attributes.
Migrates reason codes, inventory types, default category sets, lot status codes, action codes, process and discrete grade codes, profiles and recipe types, cost method codes, fiscal policies, acquisition costs, and regulatory hazard classes.

Group 2

Migrates OPM organizations, cost calendars, allocation basis codes, allocation expenses, and UN numbers.

Group 3

Migrates on hand balances, quality configurations, and burden percentages.
Creates rows for lab and plants in parameters table.

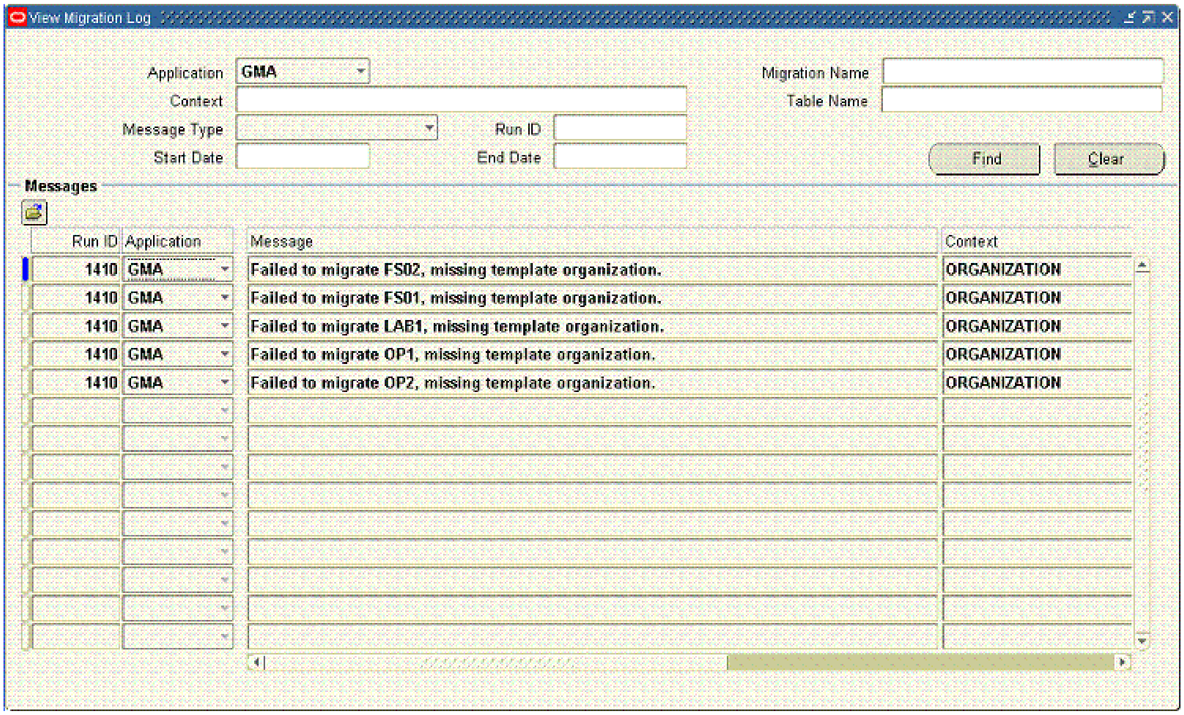
Post-Migration

- After Release 12 is installed, all data must be checked to ensure accurate migration. This can be done by reviewing the log files found in the System Administration Responsibility.

The “View Migration Log” menu option is available from OPM System Administration Responsibility to view the messages and errors logged by migration. The window will have the following options to filter the retrieval of messages / errors.

- Application - Source application that has logged the message / error.
- Migration Name – Name of the migration.
- Table Name – Table being migrated or upgraded.
- Context – The context of the migration in which the message / error was logged.
- Message Type – Informational or Error or Database Error or Progress
- Start Date – Migration start date
- End Date – Migration End Date

The logging window is available to all users, so for the migration you want to use a wildcard in the Context field, such as OPM_Convergence%, to find the information.



Application GMA

Run ID	Application	Message	Context
1410	GMA	Failed to migrate FS02, missing template organization.	ORGANIZATION
1410	GMA	Failed to migrate FS01, missing template organization.	ORGANIZATION
1410	GMA	Failed to migrate LAB1, missing template organization.	ORGANIZATION
1410	GMA	Failed to migrate OP1, missing template organization.	ORGANIZATION
1410	GMA	Failed to migrate OP2, missing template organization.	ORGANIZATION

If you find that data did not migrate correctly, or if a migration did not complete, you can run the migration script again. This finds any unmoved data and completes the process without disturbing the data that did move successfully.

Common Transformation

After all the data has been successfully migrated, you must run the common transformation patch. This takes the migrated data and manipulates it to work properly in the new Oracle Inventory organization and structure. This script performs data transformation on the following data:

- Updates Organization_Id, Inventory_Item_Id, Lot_Number, and Unit of Measure columns in the Product Development, Process Execution, and Process Planning tables.
- Migrates specifications, samples, and stability studies.
- Migrates Process Execution profiles and WIP entities.
- Migrates item categories and item lot conversions.
- Migrates move orders for Order Management.
- Updates PO shipment line.
- Updates Receiving tables.

Cost Management Post-Migration

This post-migration script completes the data transformation for the Cost Management module. This script performs the following data migration and transformation:

- Migration of resource details, cost component details, standard cost ledgers, actual cost ledgers, adjustment details, burden details, item costs, lot costs, lot costing item setup, lot cost adjustments, lot costing material transactions, lot costing burdens.
- If multiple companies are migrated to the same Legal Entity, then the scripts copy over the existing costs appropriately.
- Migration of account mappings and various data intensive GMF entities to SLA.

Regulatory Management Post-Migration

Regulatory has a post-migration script that creates records for regulatory item migration and updates dispatch history records for organization and item, migrates the regulatory items to Discrete, and migrates regulatory standalone formulas to Product Development.

Batch Recreation

You must run this concurrent program to recreate the WIP, Pending, and FPO Batches that were unreleased and canceled during the snapshot taken during pre-migration. This step renames the old version of the batch appended with a -M, and creates the new batches with the same batch number.

Planning Post-Migration

Define attributes for ASCP planning on the MRP/MPS Planning Tab and Lead-times Tab of your migrated Organizational Items.

Purchasing Post-Migration

Post migration script updates, secondary ship quantity and secondary cancelled quantitiy for PO shipments are based on the respective primary transactions quantities.

Sublots are modeled as Parent/Child lots in the converged model. When a sublot is migrated to Oracle Inventory, one record is created with the Parent Lot field representing the lot number and the lot field containing a concatenation of the lot and sublot fields. For a sublot-controlled item with a blank or NULL sublot field, the Parent lot field is left blank and the lot field contains the lot number. Receiving lot and lot supply transactions are updated for the lot number field representing the parent/child lot combination in the common inventory model. All of the error messages are logged in gml_po_mig_errors.

Edit Text Migration

Users can submit the concurrent program "GMA: Edit Text Migration" from OPM System Administration responsibility to migrate the text associated with the following entities as attachments.

- Grades
- Items
- Lots
- Organizations
- Reason Codes

Costing and Company Migration

OPM Companies are migrated to Legal Entities. All references to the OPM Company in Costing are transformed to the Legal Entity it relates to in the existing setup. In the existing setup, the OPM Company is tied to an Operating Unit. An Operating Unit is tied to a Legal Entity. After the migration, the Fiscal Policy is set up for the Legal Entity.

If you have multiple OPM companies pointing to the same Operating Unit, then these companies are no longer available after migration. If each company has a different costing method, then you lose that flexibility, and the Legal Entity is set up with only one costing type as the method used for inventory valuation.

In this case, upon migration, the Legal Entity has only one fiscal policy record with a single cost type. In order to keep existing valuations synchronized with amounts already existing in GL, the costs from the various cost methods of the companies are copied to the cost type that is selected for the Legal Entity.

However, a choice is provided in the company configuration tab to move one or more of these companies to new legal entities (created through Legal Entity pre-migration configuration) to represent the various OPM Companies. If you choose to move to new legal entities, then the inventory organizations created for the plants and warehouses that belonged to that company now point to this new legal entity. The new legal entities have the same Ledger as their primary ledgers.

The default option migrates all companies under the one legal entity that is associated to the operating unit.

Legal Entities

In Oracle Applications 11*i*, the Legal Entity (LE) is used for legal reporting purposes even though it is not a separately identified entity. The legal entity is established as a classification or context for an organization and attached to an Operating Unit. In General Ledger (GL), it is often implemented as one or more balancing segment values (BSV). Currently, there is a one-to-many relationship between the Operating Unit and Legal Entity.

In Oracle Applications 12, the Legal Entity can be created as a separately identifiable entity with its own reporting, jurisdiction, and tax rules. A legal entity can be associated to a primary ledger and any number of secondary ledgers. An inventory organization is attached to only one legal entity. ***The link between the Operating Unit and Legal Entity is broken*** and the Operating Unit only has a default legal entity context associated to it for use in transactions where a legal entity context cannot be established. The default legal entity is a required field in a shared Ledger Configuration. In a shared ledger configuration, more than one legal entity shares the same ledger. If only a single LE is associated to a Ledger, then it becomes an exclusive configuration.

LE is the reporting or legal entity that owns the transactions. The OPM Company serves both as a reporting structure (LE) as well as securing entity (like Operating Unit) for accounting in OPM.

Upon migration to Release 12, the Legal Entity linked to the OU remains as the default LE context and the inventory organizations associated to the OU also belong to this LE.

The Legal Entity product identifies all the existing legal entities that can be migrated and created as legal entities in the new Legal Entity model in Release 12. As part of the pre-migration setup, you have an option to create new legal entities. The entire configuration is used in the live migration for creation of Legal Entities in the new model.

Oracle Financials provides a migration patch, as well as pre-migration configuration for Legal Entities.

Company Migration

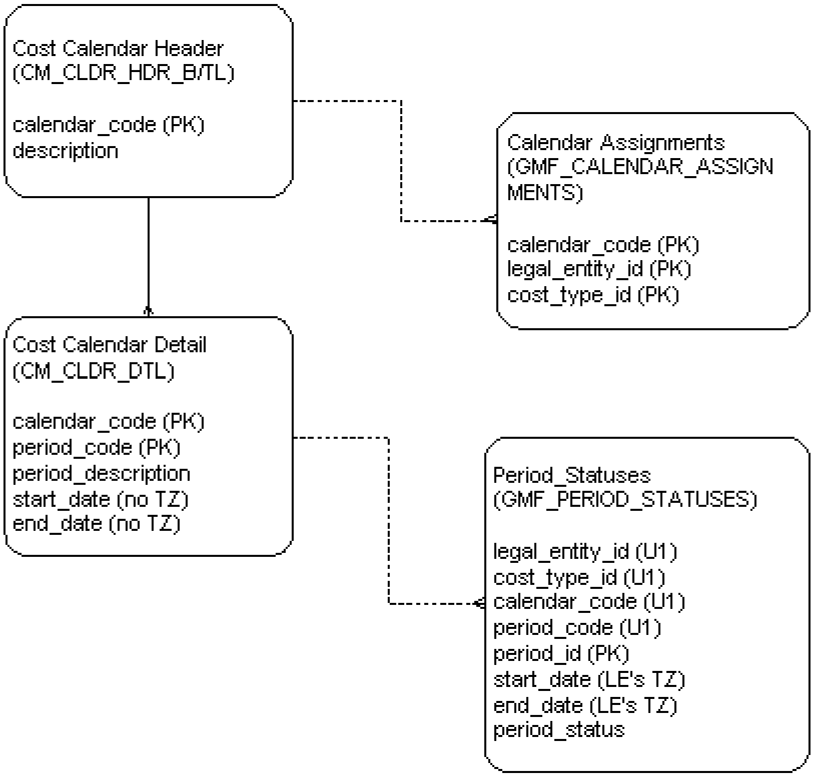
Consider the following setup before the migration:

Company	Cost Type	Operating Unit	Legal Entity
PRU	STND	PR-Operations	PR-LE
PRF	ACT	PR-Operations	PR-LE
PRD	STND	PR-Operations	PR-LE
VIS	STND	Vision Operations	Vision LE
VISF	STND	Vision Operations	Vision LE
HQ	ACT	HQ	HQ-LE
CA	STND	CA	HQ-LE

The first three companies by default are migrated to PR-LE legal entity and the valuation cost type is set to STND because a majority of the companies tied to PR-Operations Operating Unit have STND as the cost type. You have the option to migrate the PRF company to a new legal entity that is set up using the Legal Entity pre-migration configuration. For HQ and CA companies, the default cost type is chosen for the first occurring row because there is no single cost type that is used by the majority of the companies. For VIS and VISF Companies, no such problem exists.

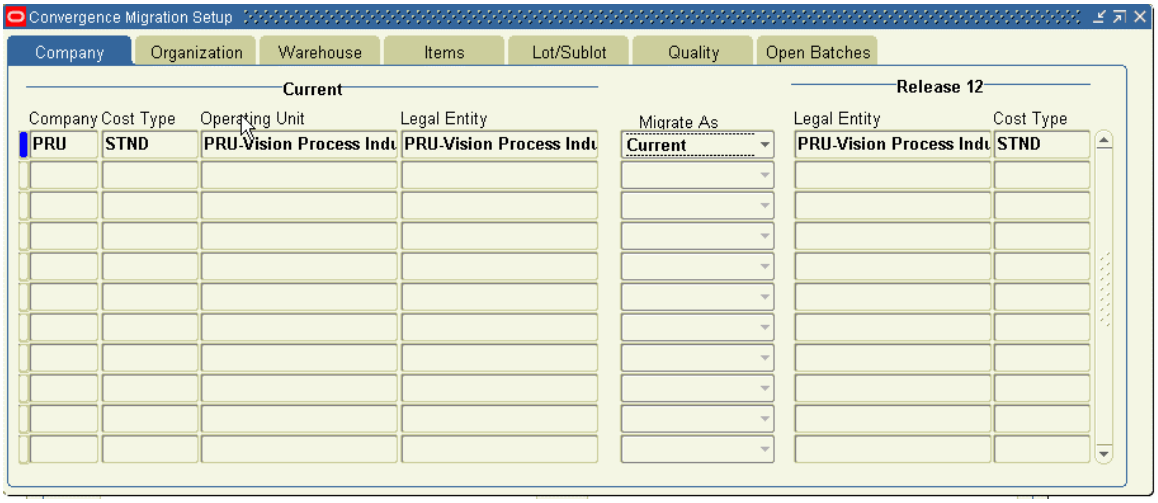
Cost Calendar

The Cost Calendars table is undergoing a three-way split in Release 12. The main cost calendar header and detail tables continue to hold the calendar and period information, but are not specific to a Legal Entity and Cost Type. Two new tables are being introduced. GMF_CALENDAR_ASSIGNMENTS stores the assignment of the calendar to a particular Legal Entity and Cost Type and GMF_PERIOD_STATUSES stores the period status by Legal Entity and Cost Type combination. The following diagram depicts the logical data model for cost calendars in Release 12. Refer to Table Mapping for details on new columns and changed tables.



Company Tab

This tab provides the company migration options. When the window is opened initially, the default options display but can be overridden.



Current					Release 12	
Company	Cost Type	Operating Unit	Legal Entity	Migrate As	Legal Entity	Cost Type
PRU	STND	PRU-Vision Process Indu	PRU-Vision Process Indu	Current	PRU-Vision Process Indu	STND

The following validations are done:

- When the window is opened, all companies from the fiscal policy display. The records are first ordered by Operating Unit and then by Company.
- By default, all companies that are linked to the same operating unit have the same legal entity attached to the Operating Unit as the effective legal entity in Release 12.
- You can move to a new legal entity for any one or more of the companies that are linked to the same operating unit.
- The new legal entities selected must have the same Ledger mapped to the Operating Unit as their primary ledger, same chart of accounts, currency, etc.
- If only one company is linked to an Operating Unit, then you cannot move to a new legal entity.

Manufacturing Accounting Controller (MAC)

OPM Costing will use the Oracle Subledger Accounting (SLA) module for accounting of material and resource transactions to the OPM Subledger. The Release 11.5 Mfg. Accounting Controller (MAC) will be made obsolete and the necessary accounting seeded data definitions will be done with the Subledger Accounting equivalent setups and rules. The SLA offers a superset of functionality that was available in MAC.

Some of the important features include the use of enhanced Account Derivation Rules to create the account mapping definitions where several hundred attributes are available to set up the account mapping rules as compared to the fixed set of attributes available in MAC, generating accounting entries using additional cost types for simulation purposes.

The existing definitions in MAC will be migrated to the equivalent SLA seed data definitions and the user need only do some minimal set ups in Release 12 before running their accounting processes. For more information refer to the Release 12 OPM Migration Reference Guide, Release 12 OPM Accounting White Paper, Subledger Accounting User Guide and Subledger Accounting Implementation Guide.

Organization Migration

Organization migration adopts different routes based on the existence of the migration setup.

Migration based on the setup

All the OPM organizations are migrated appropriately based on the setup information, as discussed in the following section about the Organization tab in the pre-migration form. Errors are logged for cases where the migration encounters issues. For example, this happens if the inventory organization entered on the setup screen was unique at the time of data setup but later the same organization was created manually before the execution of the migration.

Migration in the absence of setup

If migration setup is not available for an organization, then Manufacturing Plants and Laboratories are migrated as inventory organizations with the process-enabled indicator checked. The Non-Manufacturing organizations that are not defined as a company are migrated as a Non-Classified organization.

For organizations migrated as inventory organizations, the attributes are copied as follows:

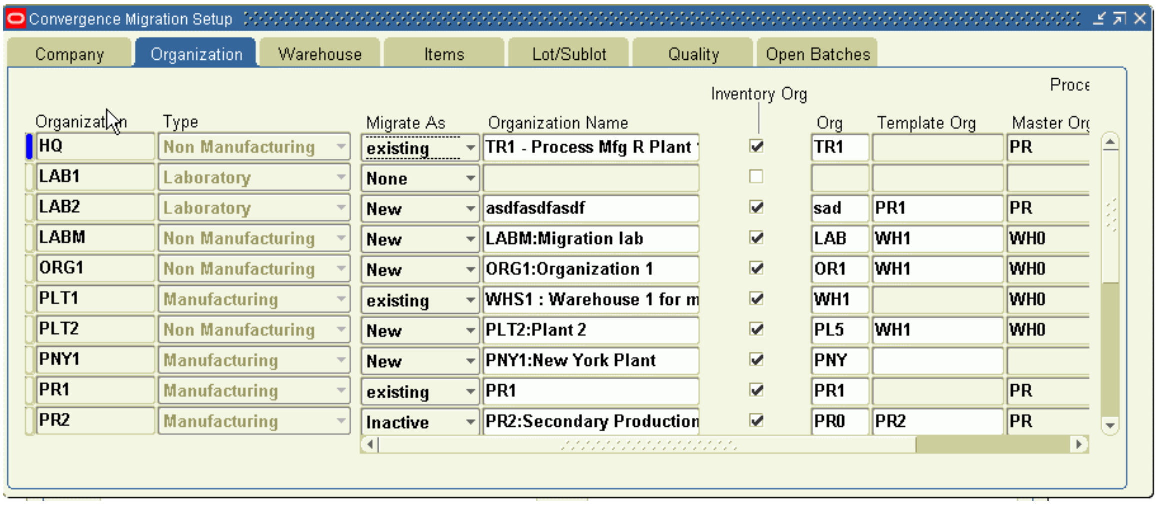
- The first three characters of the OPM Organization are used for the inventory organization. If an organization exists for the code, then the system tries building a new organization code with the first two characters of the organization and appending or incrementing a numeral for the third character. A concatenation of Organization Code and Organization Name (<Orgn_Code>: <Orgn_Name>) is used for the name while creating the record.
- If a resource warehouse code exists for the organization, then the attributes of the resource warehouse code are used to default the settings of the new organization.
- In the absence of resource warehouse code, the attributes default from an existing warehouse defined for the organization. If there are multiple warehouses, then the system chooses the first warehouse based on the warehouse code.

In the absence of a warehouse, an error is logged and the current organization is not migrated.

The new organization ID is written to the OPM organization table for further reference and to assist in subsequent data transformation.

Organization Tab

The Organization tab lets you map the attributes of the migrating organizations. The template organization is used to default the attribute values when a new inventory organization is created in Oracle Inventory.



				Inventory Org			Proce
Organization	Type	Migrate As	Organization Name		Org	Template Org	Master Org
HQ	Non Manufacturing	existing	TR1 - Process Mfg R Plant	☑	TR1		PR
LAB1	Laboratory	None		☐			
LAB2	Laboratory	New	asdfasdfasdf	☑	sad	PR1	PR
LABM	Non Manufacturing	New	LABM:Migration lab	☑	LAB	WH1	WH0
ORG1	Non Manufacturing	New	ORG1:Organization 1	☑	OR1	WH1	WH0
PLT1	Manufacturing	existing	WHS1 : Warehouse 1 for m	☑	WH1		WH0
PLT2	Non Manufacturing	New	PLT2:Plant 2	☑	PL5	WH1	WH0
PNY1	Manufacturing	New	PNY1:New York Plant	☑	PNY		
PR1	Manufacturing	existing	PR1	☑	PR1		PR
PR2	Manufacturing	Inactive	PR2:Secondary Production	☑	PR0	PR2	PR

All the organizations defined in OPM are populated by default on the form. The Migrate As field, as well as other fields, are disabled for organizations that have been migrated with the setup information. In those cases, this serves as a visual indicator for determining the migrated organizations.

Organization

This is populated by default from the orgn_code in the sy_orgn_mst table.

Type

This is populated by default from the plant_code in the sy_orgn_mst table.

Migrate As

Valid values for Migrate As are:

- Existing – If there is a resources warehouse code defined for the OPM Organization
- Inactive - If the organization is marked for deletion
- New - All other organizations

When the Migrate As list value is changed, then the following validations are performed.

New

A warning message displays if there is a resource warehouse code associated with the organization. The Organization Name and Inventory Org checkbox are enabled, and the Organization Name defaults as <Orgn Code> : <Orgn Name>.

If the organization is of type Plant or Lab, then the Inventory Org checkbox is checked and the following fields are populated:

- **Org** with the first three characters of the OPM Organization. If an organization exists for the code, then the system builds a new organization code with the first two characters of the OPM organization and appends a numeral for the third character.
- **Template Org** with resource warehouse code if one exists, or with the existing warehouse code of the organization.
- **Master Org** with the master organization of the template organization.

Existing

The Organization Name field is enabled and the Inventory Org, Org, Template Org, and Master Org fields are disabled.

The Organization Name and Org fields are populated with the resource warehouse code organization name and resource warehouse code if one exists for the OPM Organization.

Inactive

Inactive has the same validations as New. In addition, a warning message displays if the organization has dependents.

None

All the dependent fields are disabled. Appropriate errors are raised if the Organization was used elsewhere in OPM.

Organization Name

If Migrate As is Existing, then the Organization name defaults to Resource Warehouse Organization Name. If Migrate As is New or Inactive, then the Organization name defaults to Organization Code:Organization Name.

The validations on Organization Name are based on the value of the Migrate As indicator:

- If Migrate As is Existing, then only valid HR Organizations are allowed. Only one to one mapping is allowed for the organizations. The organization entered cannot be associated as a mapping with any other OPM organizations. The organization entered must have the same set of books ID as that of the OPM organization, based on the company. The Inventory Org and Org fields are populated based on the organization entered.

- If Migrate As is New or Inactive, then a unique HR organization name must be entered in this field.

Inventory Org

Inventory Org is enabled by default for manufacturing plants and laboratories. An error displays if there are dependents on the organization and the checkbox is unchecked. When checked, the Org, Template Org, and Master Org fields are enabled.

The Org field is populated with the first three characters of the OPM Organization. If an organization exists for the code, then the system builds a new organization code with the first two characters of the OPM organization and appends a numeral for the third character.

When checked, the Template Org is populated with the resource warehouse code if one exists. Otherwise, it is populated with an existing migrated warehouse that belongs to the organization. The Master Org is populated based on the Template Org.

Org

A unique inventory organization must be entered. If Migrate As is Existing, then the organization is populated with the resource warehouse code.

If Migrate As is New or if Inventory Org is enabled, then the first three characters of the OPM Organization are used for the inventory organization. If an organization exists for the code, then the system builds a new organization code with the first two characters of the OPM organization and appending a numeral for the third character. This field is open for updates if the code has to be changed.

Template Org

If Migrate As is New or if Inventory Org is enabled, then this is defaulted from an existing warehouses defined for the organization. If there are multiple warehouses, then the system selects the first warehouse based on the warehouse code.

The template organization entered must have the same set of books ID as that of the OPM organization, derived based on the company.

Master

If Migrate As is New or if Inventory Org is enabled, then this is defaulted with the master organization of the template organization. The master organization entered must have the same set of books ID as that of the OPM organization, derived based on the company.

Process Enabled

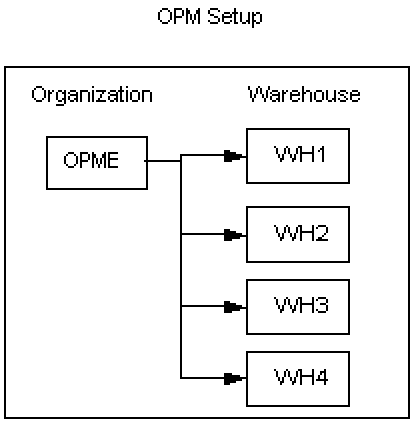
If Migrate As is New or if Inventory Org is enabled, then the default is enabled. An error is raised if there are dependents on the organization and the checkbox is not enabled.

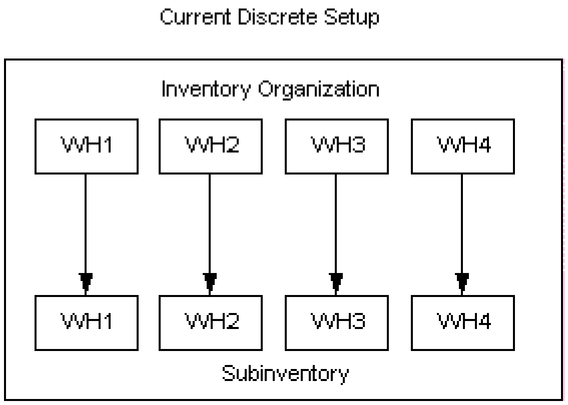
Regulatory Org

If Migrate As is set to a value other than none, the check box is enabled.

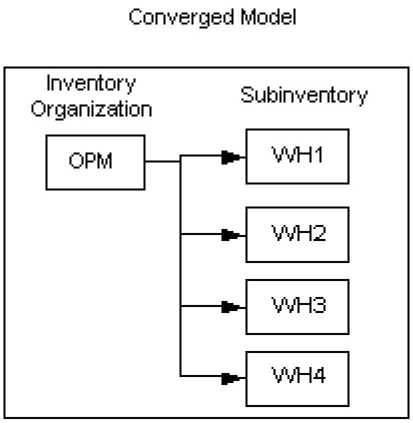
Warehouse Migration

Currently, OPM Warehouses already exist in Oracle Inventory as inventory organizations. As part of migration, you can convert these warehouses to subinventories under existing inventory organizations or retain them as inventory organizations. Also, note that OPM Warehouse Locations are the same as Discrete Stock Locators.
For example, consider the following setup:





You can map WH1, WH2, WH3, and WH4 as subinventories under the new inventory organization OPM created for the organization OPME.

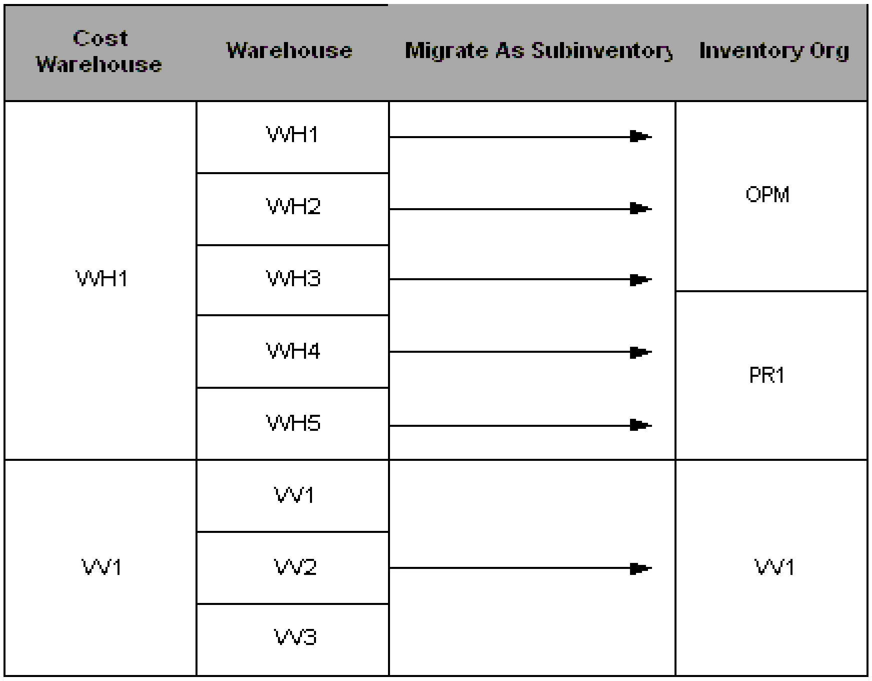


The following must be considered before converting an existing warehouse to a subinventory.

- Existing on-hand balances for the warehouses move under the organization where the warehouse is being created as a subinventory. Therefore, balances from WH1 are found in subinventory WH1 within Inventory Organization OPM.
- The source inventory organization is inactivated, which invalidates, or obsoletes, the items and any parameters associated with that inventory organization. This is based on the setting of the Disable Warehouse Organization flag in the Migration form. This is optional.
- Any open documents (purchase order, sales order) against the source organization are not accessible once the organization is inactivated. This is based on the setting of the Disable Warehouse Organization flag in the Migration form. The impact of inactivating the source inventory organization must be carefully considered. If you are migrating from 11.5.5 or higher, then it is possible to migrate open purchase orders. If the organization on the shipment line (into which the purchased material is received) is inactivated, then you cannot receive against these purchase orders. This is because you are not able to query for receipt lines based on the inactive inventory organization.

Following are the restrictions for converting an existing warehouse to a subinventory:

- All the warehouses that are being mapped as subinventories under a single organization must belong to the same cost warehouse.



Cost Warehouse	Warehouse	Migrate As Subinventory	Inventory Org
WH1	WH1	→	OPM
	WH2	→	
	WH3	→	
	WH4	→	PR1
	WH5	→	
W1	W1		W1
	W2	→	
	W3		

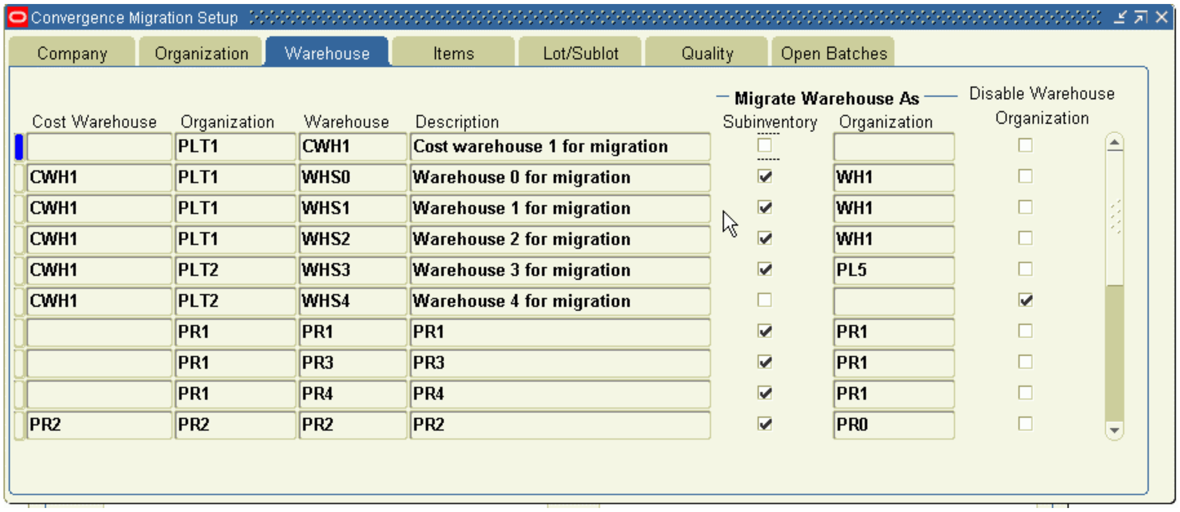
- The Locator is unique within an Inventory Organization in Oracle Inventory. In OPM, two different warehouses can have the same location name. When these warehouses are moved as a subinventory under a single organization, the two different locations with the same name cannot coexist. Therefore, the warehouse code and locator are concatenated to form a new stock locator.

- The Locator Control of the warehouse must match the Locator Control of the organization under which it is being moved as a subinventory.
- The warehouse and OPM organization that is referenced by quality samples must maintain a similar hierarchy to a subinventory under the same inventory organization. If the warehouse is converted to a subinventory under a different inventory organization, the sample source and storage subinventory may be invalid for the sample creation inventory organization noted on the sample record.

Converting warehouses to subinventories is an optional step. If you choose to migrate without creating the subinventories, then the migration follows the default path and migrates the inventory balances from the warehouses to the existing discrete organization for the warehouse.

Warehouse Tab

The following tab lets you specify the conversion of existing warehouses to subinventories.



				Migrate Warehouse As		Disable Warehouse
Cost Warehouse	Organization	Warehouse	Description	Subinventory	Organization	Organization
	PLT1	CWH1	Cost warehouse 1 for migration	☐		☐
CWH1	PLT1	WHS0	Warehouse 0 for migration	☑	WH1	☐
CWH1	PLT1	WHS1	Warehouse 1 for migration	☑	WH1	☐
CWH1	PLT1	WHS2	Warehouse 2 for migration	☑	WH1	☐
CWH1	PLT2	WHS3	Warehouse 3 for migration	☑	PL5	☐
CWH1	PLT2	WHS4	Warehouse 4 for migration	☐		☑
	PR1	PR1	PR1	☑	PR1	☐
	PR1	PR3	PR3	☑	PR1	☐
	PR1	PR4	PR4	☑	PR1	☐
PR2	PR2	PR2	PR2	☑	PR0	☐

Cost Warehouse

Populated by default from the Cost Warehouse Association table for the warehouse code in the OPM Inventory Warehouse Master table.

Organization

Populated by default from the Organization Code of the warehouse.

Warehouse

Populated by default from Warehouse Code in the OPM Inventory Warehouse Master table.

Description

Populated by default from Warehouse Name in the OPM Inventory Warehouse Master table.

Subinventory

The Organization field is enabled when the subinventory checkbox is checked.

Organization (Migrate As)

Enabled when the Subinventory flag is checked.

The following validations are performed on the organization entry:

- If the organization is an existing organization, then an error is raised if a subinventory already exists under the organization with the same name as the warehouse code.
- The warehouse company and the organization company must be the same.
- The cost warehouse for the current warehouse matches the cost warehouse of the warehouses associated as subinventory under the organization.
- The Locator Control of the warehouse matches the Locator Control of the organization.
- The existing locations under the warehouse must not exist with the same name under any of the warehouses that have been associated to the organization as subinventories.

Inventory, Lot, and Sublot Migration

Each area of OPM Inventory is migrated in a specific way. This section describes how each part of Inventory is migrated, where the data is transferred, and what the options are when it is not migrated.

Items

The Oracle Inventory item master was enhanced for Release 12 to include OPM item master attributes. Since items are already defined in Oracle Inventory, the existing item definition in the Oracle Inventory item master is updated to reflect the OPM item attributes. There is no impact to an item's definition in a discrete organization. For example, if an OPM item is dual UOM in OPM and exists in the discrete organization as a non-dual item, it continues to work as a non-dual item in the discrete organization.
If an item has never been transacted in a particular warehouse, it is not migrated to the corresponding inventory organization. If the item is required in that organization in the future, then you assign the items to that organization. This also ensures that the item is visible in an organization only if it is used there.
A separate, and optional, migration script is provided to migrate all OPM items to a specific inventory organization, regardless of whether it was used in that organization. This migration does not run automatically during the installation.
There are many OPM item master attributes that become obsolete in the common inventory model. A separate migration script is provided that lets you migrate the

obsolete columns to descriptive flexfields. This migration does not run automatically during the installation, but must be run manually after the convergence code installation.

Note:

For an item under shelf life (or expiration date) control in Oracle Inventory, a number of days, greater than zero must be entered on the item master. This value is used to calculate the expiration date of a lot. In OPM Inventory, it is possible to enter 0 (zero) for shelf life on the item master. Since this is not supported in Oracle Inventory, an item defined with shelf life days as 0 (zero) in OPM is migrated as an item which is not under lot expiration control.

Inventory Type

The Inventory types in OPM are migrated to the Item Types in Oracle Inventory. The item types are defined as the Lookup Values in Oracle Inventory.

OPM Category Sets/Classes Assignments

In the common inventory model, the OPM Item Classes and Category Sets relationships are seeded in Oracle Inventory. The migration script updates the category set assignment to the Oracle Inventory functional areas based upon the OPM definition. Note that only the following OPM Item Classes are seeded in Discrete:

- Allocation Class
- Sequence Dependency Class
- Substandard Item Class
- Technical Class
- Process GL Class
- GL Business Class
- Process Cost Class
- Process Product Line

Lot/Sublot

Lot / Sublots are global in OPM whereas they are organization specific in discrete. Depending upon the usage of the lots in OPM, they may need to be migrated to multiple organizations. The lot is migrated to Oracle Inventory if:

- An OPM inventory transaction or inventory balance exists for a given lot/sublot and warehouse (corresponding to the inventory organization in Oracle Inventory) combination. The check is performed for both tables because the OPM purge can potentially remove data from one of these tables.

- Any quality data exists for the lot. In that case you must determine the inventory organization where the Process Quality expects the lot to exist and the discrete lots are created for these organizations.
- There are any saved simulations for the lot in the NPD spreadsheets (lm_sprd_dtl). If simulations exist, then the lot is migrated for the correct inventory organization.
- A lot cost, adjustment, or burden exists for the Lot/ Sublot and warehouse combination.

In order to ensure that the lots are created in the correct inventory organizations, the migration of the lot is run as part of the upgrade script for lot_id column. A routine is provided to return the discrete lot number for an OPM lot / warehouse or inventory organization combination. The routine creates the discrete lot if it does not exist.
If a lot exists in multiple locations within a warehouse and has a different lot status in those locations, direct mapping to the Discrete Lot Status is not possible.. A pre-migration validation window is provided to show the list of lots and lets you correct these prior to the convergence migration. Refer to Lot Status Migration Recommendations for a discussion of migration options.

Because the lot number is a key field that is stored as a reference in other application tables, it is important that you complete the pre-migration, or change the lot number as needed.

Sublots are modeled as Parent/Child lots in the common inventory model. When a sublot is migrated to Oracle Inventory, one record is created with the Parent Lot field representing the lot number and the lot field containing a concatenation of the lot and sublot fields. For a sublot-controlled item with a blank or NULL sublot field, the Parent lot field is left blank and the lot field contains the lot number.

The following table indicates how the OPM lots and sublots are migrated:

OPM Item	OPM Lot	OPM Sublot	Discrete Parent Lot	Discrete Lot (Child)
OPM100 (No Sublot Control)	L100	NULL	NULL	L100
OPM110 (Sublot Controlled)	L110	S1	L110	L110S1
	L110	S2	L110	L110S2
OPM111 (Sublot Controlled)	L111	NULL	NULL	L111
	L111	S1	L111	L111S1
	L111	23	L111	L11123

Note the Discrete Lot L11123 is created using the migration. It is difficult to determine the child lot component in this lot number. Using the pre-migration window, you can specify a delimiter for lot/sublot concatenation and selectively change the discrete lot numbers prior to migration.

Lot/Sublot UOM Conversions

Items and Lots are organization specific, so the OPM lot UOM conversion is migrated to multiple organizations. This depends on the organization to which the OPM lots have been migrated. Lot conversion migration is run toward the end so that it creates lot conversions for all organizations in which the lots have been created.

Action Codes

Expiration Action codes are maintained in a new form and table in Oracle Inventory. All existing expiration action codes are migrated from OPM to the new table. The Item Master contains a new column to specify a default expiration action code for an item, while the lot master includes a column to maintain the value for a specific lot. The migration process populates these values where appropriate.

Grade

Grade codes are maintained in a new form and table in Oracle Inventory. All existing grade codes are migrated from OPM to the new table. The Item Master contains new columns to designate an item as grade controlled and to specify a default grade for an item. The lot master includes a column to maintain the value for a specific lot. The migration process populates these values where appropriate.

Inventory Calendar

Since inventory transactions are not migrating to discrete inventory, there is no need to ensure that the accounting periods exist for the old data. The migration script checks if the inventory periods are open. If they are not open, then the script opens them.

Item Categories

The item categories for all process-enabled organizations are migrated from OPM Item Categories assignment to Oracle Inventory.

Item UOM Conversion

OPM Inventory customers currently define units of measure and related setup through the Oracle Inventory forms. Migration of this data is not necessary.

Inventory Transactions

Inventory transactions are not migrated. The old responsibility and menu options remain available as read-only. This provides visibility of historical transactions.

Inventory Balances

On-hand inventory balances at the warehouse-location level from the OPM application are transferred to Oracle Inventory. Note that the inventory transactions are not migrated. You can view the historical inventory data in OPM using the Process Inventory responsibility.

The migration results in the creation of the on-hand balances in Oracle Inventory tables and sets the balance in the OPM tables to zero. The inventory balances are migrated to the inventory organization/subinventory that is mapped to the OPM warehouse. If the inventory period for the organization is not open, then the migration opens the period prior to migrating the balances.

In OPM the material status is associated to the on-hand inventory, where as in the discrete common inventory model, the material status can be associated to lot, subinventory, locator, etc. The on-hand inventory is not assigned any specific material status.

The migration approach for the on-hand inventory lot status is to associate the corresponding material status to the migrated lots. This can result in an issue, where it is not possible to determine a single material status for the migrated lot in an organization because the corresponding lot in OPM exists in multiple warehouse/locations and has a different lot status in each location. The following example shows some of these cases:

Organization	Migrate As	Organization Code	Comments
PLT1	Existing	WH1	
PLT2	New	PL2	
ORG1	New	OR1	
LAB	New	LAB	

Warehouse Setup

Organization	Cost Warehouse	Warehouse	Location Controlled	Subinventory	Inventory Organization
PLT1	CWH1	WHS0	Y	Y	WH1
PLT1	CWH1	WHS1	Y	Y	WH1
PLT1	CWH1	WHS2	N	Y	WH1
PLT1	CWH1	CWH1	N	Y	WH1
PLT2	CWH1	WHS3	Y (Dynamic)	N	WH3
PLT2	CWH1	WHS4	Y	N	WH4

Inventory Balances Setup

Item	Lot	Sublot	Warehouse	Location	Lot Status	On Hand	On hand2	Parent Lot	Lot
MIG100	L1000	S1	WHS0	RECV	INSP	700	1500	L1000	L1000#S1-INSP
MIG100	L1000	S1	WHS1	STORE	HOLD	100	200	L1000	L1000#S1
MIG100	L1000	S1	WHS1	ROW1	HOLD	100	200	L1000	L1000#S1
MIG100	L1000	S1	WHS1	RECV	HOLD	800	1600	L1000	L1000#S1
MIG100	L1000	S2	WHS0	RECV	INSP	400	800	L1000	L1000#S2-INSP
MIG100	L1000	S2	WHS0	STORE	GOOD	1000	2100	L1000	L1000#S2
MIG100	L1000	S2	WHS0	ROW1	BAD	250	500	L1000	L1000#S2-BAD
MIG100	L1001	S1	WHS0	STORE	GOOD	1000	2400	L1001	L1001#S1-GOOD
MIG100	L1001	S1	WHS2	-	PROD	1500	3500	L1001	L1001#S1
MIG105	L1050	S001	WHS3	TEMP1	GOOD	500	1000	L1050	L1050#S001
MIG105	L1050	S001	WHS3	TEMP2	PROD	500	1000	L1050	L1050#S001-PROD
MIG105	L1050	S001	WHS3	TEMP3	BAD	500	1000	L1050	L1050#S001-BAD
MIG105	L1050	S001	WHS3	TEMP4	SHIP	500	1000	L1050	L1050#S001-SHIP

Lot Genealogy

Since transactions are not being migrated, the Oracle Inventory genealogy inquiry shows information for transactions performed on lots after migration only. Visibility of historical data is through the OPM Genealogy Inquiry. To ease data interrogation, a drill down to the OPM Inquiry is available from the Oracle Inventory Inquiry form.

Period Balances

In Release 12, the period ending inventory quantity balances for process organizations are stored in a new table owned by the Process Manufacturing Financials application. The inventory period ending balances prior to Release 12 are migrated to this new table. However, since only the prior inventory period balances as of the migration date are relevant, the script migrates the two most recent inventory period balances for each warehouse.

Lot Status

Lot status control is achieved using Material Status in Release 12. Material status is available to Oracle Inventory, as it no longer requires a WMS license. OPM Lot Status codes are migrated to the new model through the definition of Material Status, and are defined as lot level status.

Lot Status Migration Logic

Lot statuses in OPM are migrated to the material status in Oracle Inventory. In the common inventory mode, the material status is accessible and functional in the core inventory responsibility, even if WMS is not installed.

The Material Status definition window is accessible through WMS and Oracle Inventory. Therefore, the lot status from OPM are migrated and displayed through this window with the Lot usage flag checked. The other usage levels of subinventory, locator, and serial can be selected post migration.

Once material is associated with a material status, you cannot change the definition. Therefore, if we migrate the lot status, then migrate the lots and (associate them with a status), it is not easy to update the status definition.

In Release 12, material has to be shippable to be pick confirmed. You must create a new material status if you want to prevent pick confirm. Otherwise, material which cannot be pick confirmed now (as it is non-shippable) is able to be pick confirmed because the migrated material status definition differs slightly to the lot status definition.

The concern with material status versus lot status is really with the slight difference in definition and how it is considered by the system. In OPM, we check for a shippable status at pick confirm. If the material is not shippable, then it cannot be pick confirmed. In Oracle Inventory, we are migrating a Non Orderable lot status to a material status which disallows pick confirm only and a non shippable lot status to a material status which disallows ship confirm only. Therefore, material with a lot status that is orderable but non-shippable in OPM cannot be pick confirmed. However, the applicable material status will ALLOW pick confirm, but DISALLOW ship confirm.

For a Lot Status with the rejected indicator set, document level transactions must be prevented (i.e. use in Sales Orders, Production). Setting the rejected indicator in OPM sets all other flags off.

Both order_proc_ind and rejected_ind in OPM are migrated to Disallow the following transactions types linked to the Sales Order or Internal Order source for the material status migrated to Oracle Inventory.

Where ORDER_PROC_IND is turned off, the material status definition disallows these transactions. When defining a material status you explicitly disallow a transaction.

Transaction Type	Description	Source Type
Sales Order Pick	Staging transfer on a Sales order	Sales Order
Internal Order Pick	Staging transfer on an Internal order	Internal Order

Both prod_ind and rejected_ind are migrated to Disallow the following transactions types linked to the Job or Schedule source for the material status migrated to Oracle Inventory. Where PROD_IND is turned off, the material status definition disallows these transactions.

Transaction Type	Description	Source Type
WIP Issue	WIP Issue	Job or Schedule
WIP Completion Return	WIP Completion Return	Job or Schedule
WIP Assy Completion	Complete assemblies from WIP to stores	Job or Schedule
WIP Component Return	WIP component return transaction	Job or Schedule
WIP By-product Completion	WIP By-product Completion	Job or Schedule
WIP By-product Return	WIP By-product Return	Job or Schedule

Because Process Execution does not use the following transaction types, the following transaction types are allowed.

Transaction Type	Description	Source Type
WIP estimated scrap	WIP estimated scrap transaction	Job or Schedule
WIP Lot Quantity Update	WIP Lot Quantity Update	Job or Schedule
WIP Lot Bonus	WIP Lot Bonus	Job or Schedule
WIP Lot Merge	Lot Merge	Job or Schedule
WIP Lot Split	Lot Split	Job or Schedule
WIP return from scrap	Return assemblies scrapped to WIP	Job or Schedule
WIP Negative Return	WIP Negative Return	Job or Schedule
WIP Negative Issue	WIP Negative Issue	Job or Schedule
WIP assembly scrap	Scrap assemblies from WIP	Job or Schedule
WIP cost update	Update cost of a WIP item	Job or Schedule

The shipping_ind and rejected_ind are migrated to Disallow the following transactions types linked to the Sales Order or Internal Order source for the material status migrated to Oracle Inventory. Where SHIPPING_IND is turned off, the material status definition disallows these transactions.

Transaction Type	Description	Source Type
Sales order issue	Ship Confirm external Sales Order	Sales Order
Internal order issue	Ship Confirm Internal Order: Issue	Internal Order
Int Order Intr Ship	Ship to intransit sourced by Internal order	Internal Order

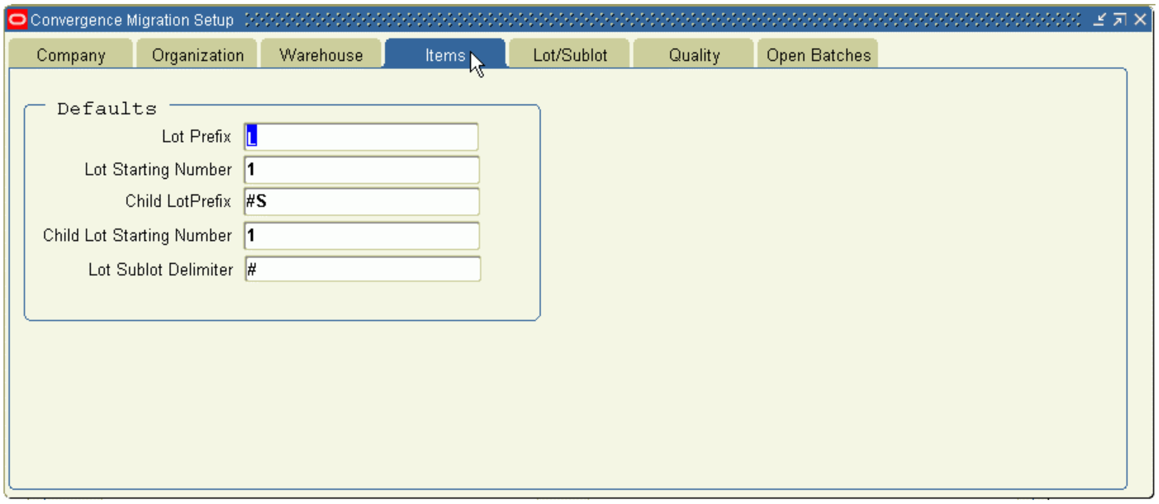
The following transaction types are set to always "Allowed" during migration of OPM lot status, they are not explicitly disallowed by a material status definition.

Transaction Type	Description	Source Type
Logical Sales Order Issue	Logical Sales Order Issue	Sales Order
Int Order Direct Ship	Direct transfer between two organizations on a internal order	Internal Order
Internal Order Xfer	Subinventory transfer sourced by Internal order	Internal Order

The rejected_ind of 1 is migrated as a discrete status which is not nettable, does not allow reservation, and is not ATP. This also prevents the document level transactions, sales order pick, sales order issue, and production transactions.

Items Tab

In the OPM Item Master, the auto lot numbering setup is optional. You can choose not to enable the OPM items for auto lot numbering. In Oracle Inventory, controls for automatic lot numbering must be defined, either at the item or organization level. However, lot numbers can be entered manually. The Items tab lets you specify the default values that the migration scripts can use for those OPM Items that are enabled for auto lot numbering.



Defaults

Lot Prefix	L
Lot Starting Number	1
Child LotPrefix	#S
Child Lot Starting Number	1
Lot Sublot Delimiter	#

Lot Prefix

Enter the default Lot Prefix for the OPM item migration. If an item in OPM is not auto lot enabled, then the migration script uses this value as Lot Prefix for the migrated discrete item.

Lot Starting Number

Enter the default Lot Starting Number for the OPM item migration. If an item in OPM is not auto lot enabled, then the migration script uses this value as Lot Starting Number for the migrated discrete item.

Child Lot Prefix

Enter the default Child Lot Prefix for the OPM item migration. If an item in OPM is not auto lot enabled, then the migration script uses this value as Child Lot Prefix for the migrated discrete item.

Child Lot Starting Number

Enter the default Child Lot Starting Number for the OPM item migration. If an item in OPM is not auto lot enabled, then the migration script uses this value as Child Lot Starting Number for the migrated discrete item.

Lot Sublot Delimiter

The OPM lots and sublots are concatenated and migrated to the discrete lot. You can specify if the migration script uses a delimiter while concatenating the OPM lots and sublot.

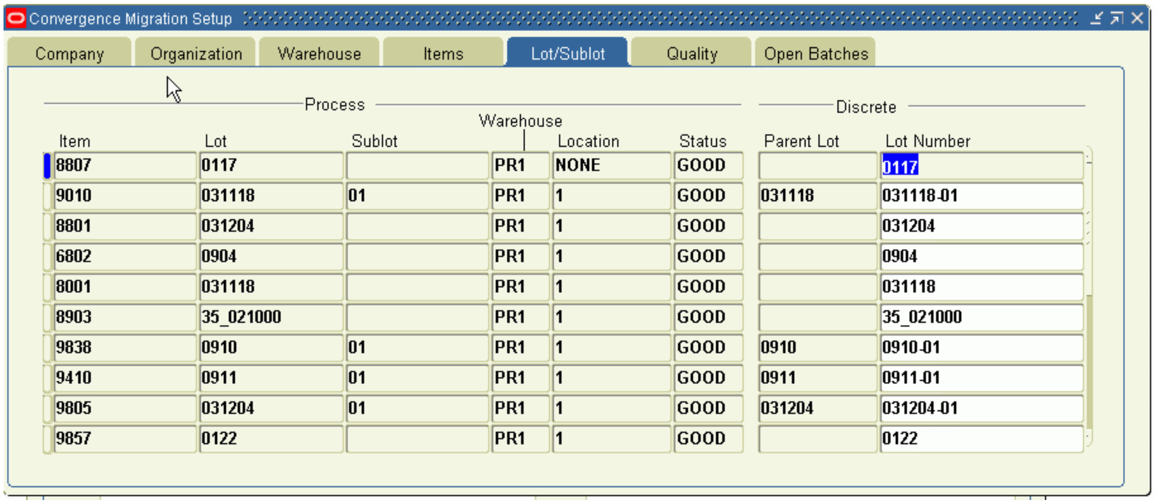
If there are any records in ic_lots_mst_mig (a table created for the migration process) with migrated_ind = 1, then this field is protected from update.

If the value for this field is changed, you are prompted 'Do you want to regenerate the Discrete Lot numbers'. If you select Yes, then the routine regenerates all the discrete lot numbers in this window using the new delimiter.

Lot/Sublot Tab

In OPM, lot status is a characteristic of inventory in a given location. A lot can have a different status per storage location. Material status designates the transactions which can be performed on inventory. A Material status can be associated with a subinventory, locator, lot or serial. However, a lot has only one material status associated with it in an inventory organization, even though it can be stored in multiple locations.

A material status is associated with each migrated lot. To counter the potential problem where a lot in OPM has a different status per storage location, data must be entered in the form below. This provides the option to correct these errors by either changing the status of the lot to the same status in all warehouse locations or assign a different lot number for Inventory with a different status.



Process						Discrete	
Item	Lot	Sublot	Warehouse	Location	Status	Parent Lot	Lot Number
8807	0117		PR1	NONE	GOOD		0117
9010	031118	01	PR1	1	GOOD	031118	031118-01
8801	031204		PR1	1	GOOD		031204
6802	0904		PR1	1	GOOD		0904
8001	031118		PR1	1	GOOD		031118
8903	35_021000		PR1	1	GOOD		35_021000
9838	0910	01	PR1	1	GOOD	0910	0910-01
9410	0911	01	PR1	1	GOOD	0911	0911-01
9805	031204	01	PR1	1	GOOD	031204	031204-01
9857	0122		PR1	1	GOOD		0122

Item

This is a display only field and displays the item for the lot.

Lot

This is a display only field and displays the OPM lot that has a different lot status in different locations in a warehouse.

Sublot

This is a display only field and displays the OPM Sublot for sublot controlled items that has a different lot status in different locations in a warehouse.

Warehouse

This is a display only field and displays the OPM warehouse for the lot where the lot has a different lot status in different locations.

Location

This is a display only field and displays the OPM warehouse locations where the lot has a different lot status in a warehouse.

Status

This is a display only field and displays the inventory lot status for the OPM Lot in a warehouse location.

Parent Lot

This is a display only field and displays the parent lot that the OPM Lot migration creates for this OPM lot.

Lot (Discrete)

This field displays the Discrete Lot that the migration script creates for the OPM Lot.

You can modify this lot to be a different value. If you modify this value, then the following validations are executed:

1. Verify that there is not another lot with the same discrete lot number on this window for this item and warehouse.
2. Verify that for the rows not displayed in this window (those which do not have multiple lot statues in different locations) there is not another lot with the same discrete lot number for this item and warehouse.

OPM Quality Management Migration

Prior to migration, the stability study did not require association to an Inventory organization. As part of this migration, you must assign these studies to an organization.

In addition, the migration of warehouses to subinventories under the same inventory organization preserves the existing data structure for inventory specification validity rules, sample numbers, and inventory and WIP sample source information. Otherwise, transformation of this data cannot occur.

Lab organizations that are migrated can be used by Process Product Development (GMD) and Quality. There is a new distinction between research and development labs and quality labs as configured through a system parameter on the respective product menus.

There are no restrictions on the status of specifications, specification validity rules, and stability studies or the disposition of samples prior to migration.

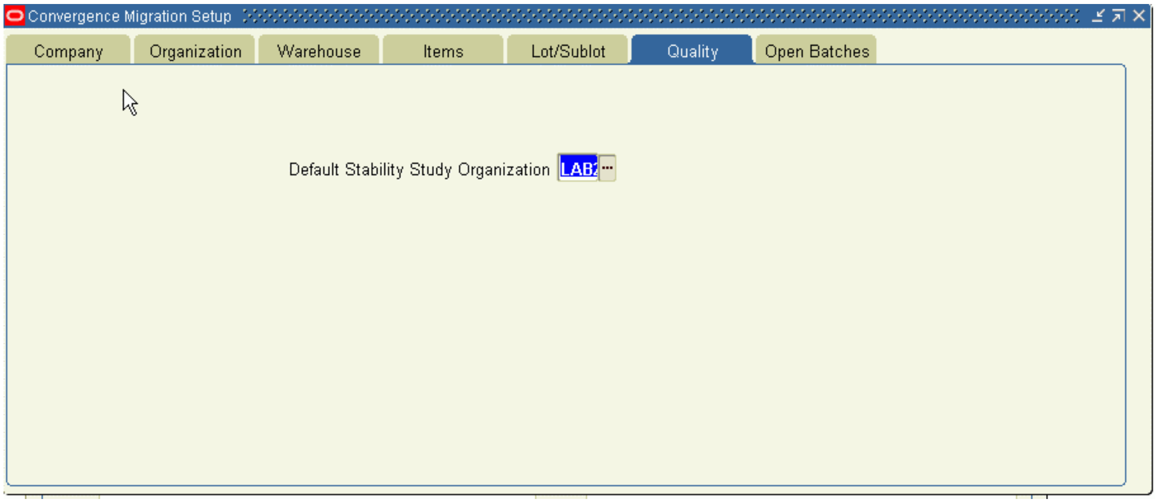
Profile options for specification version control, exact specification match, and include optional tests in sample disposition are mapped to system parameters. Site-level profile option values are migrated to organization parameters.

Grades and action codes are migrated to Oracle Inventory tables.

Document numbering for samples and stability studies can be configured through Process Quality Parameters.

Quality Tab

After convergence, stability studies must be associated to an organization. The Quality tab lets you select the organization you want to associate the stability studies to, if they currently do not have one. While this is not a required field, it must be populated prior to migration. If another organization is entered, then it must be a valid process-enabled organization and a Lab organization.



Default Stability Study Organization LAB2

OPM Batch Migration

All WIP and Pending batches and firmed planned orders must be closed or canceled prior to the installation of Release 12 so you cannot perform any action on these objects in the common inventory model. You can manipulate batches and firmed planned orders created in the common inventory model.

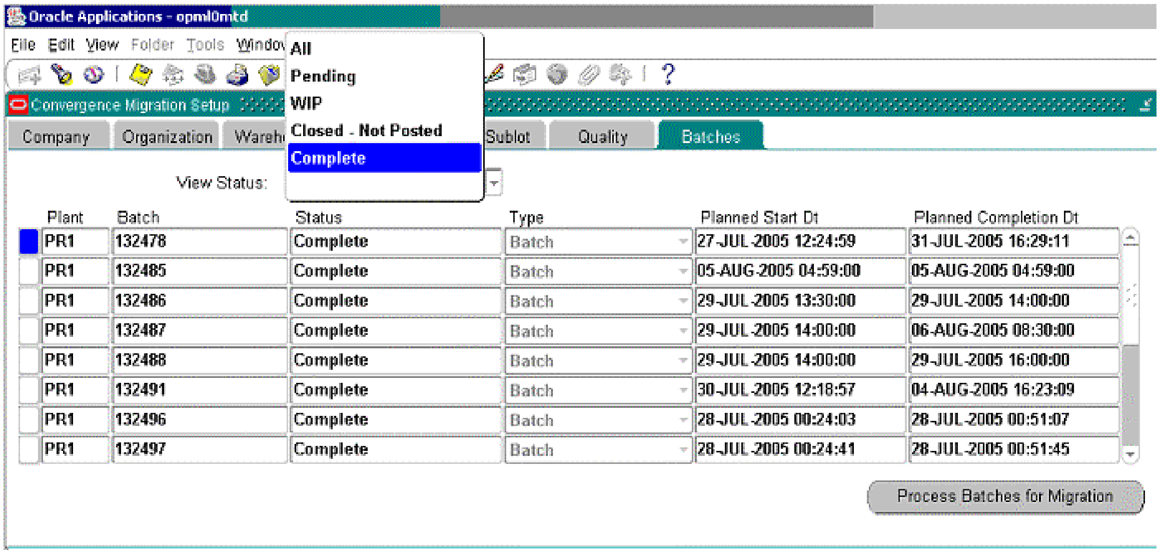
You are given the option to recreate Pending batches, WIP batches and FPOs after migration, with batch numbers being retained in Release 12. This option is available to everyone regardless of the 11*i* release. This option is available for all of the aforementioned batches, with the exception of batches that are parents of phantom batches with a different batch status than that of the parent which must be dealt with manually.

Just prior to migration, a snapshot is taken of the data pertaining to all Pending batches, WIP batches and FPOs, including associated POC and transaction data. After the snapshot is taken, the WIP batches are unreleased, and then all of the batches included in the snapshot are canceled. Once this procedure has been run, you can no longer manipulate these batches until after the migration to Release 12.

After the migration to Release 12 has been completed and the Common Data Transformation has been run, you must run a concurrent request to recreate the batches and FPOs from the snapshot data that was gathered during the migration process. All applicable batch data, including POC data material transactions, product lots and reservations (allow you to guarantee ingredient availability for the batch), are generated. All attachments are carried forward to the recreated batches. The old batch is renamed with a –M added to the batch number so that the recreated batch can use the original batch number. The applicable attachments, GMD Lab, and OPM Quality data are copied to the recreated batches. A mapping table is generated to link the old plant code, batch ID, and batch number to the new organization ID, batch ID, and batch number.

Batches Tab

The Batches tab lets you see what your current batches look like. This tab displays all the open Batches (Pending, WIP, Completed) and Open FPOs (Pending). The information on this window is display only.


Oracle Applications - opml0mtd
File Edit View Folder Tools Window
Convergence Migration Setup
Company
Organization
Wareh
Sublot
Quality
Batches
All
Pending
WIP
Closed - Not Posted
Complete
Process Batches for Migration

View Status: Complete

Plant	Batch	Status	Type	Planned Start Dt	Planned Completion Dt
PR1	132478	Complete	Batch	27-JUL-2005 12:24:59	31-JUL-2005 16:29:11
PR1	132485	Complete	Batch	05-AUG-2005 04:59:00	05-AUG-2005 04:59:00
PR1	132486	Complete	Batch	29-JUL-2005 13:30:00	29-JUL-2005 14:00:00
PR1	132487	Complete	Batch	29-JUL-2005 14:00:00	06-AUG-2005 08:30:00
PR1	132488	Complete	Batch	29-JUL-2005 14:00:00	29-JUL-2005 16:00:00
PR1	132491	Complete	Batch	30-JUL-2005 12:18:57	04-AUG-2005 16:23:09
PR1	132496	Complete	Batch	28-JUL-2005 00:24:03	28-JUL-2005 00:51:07
PR1	132497	Complete	Batch	28-JUL-2005 00:24:41	28-JUL-2005 00:51:45

It is suggested that you review open batches and FPOs and clean up whatever data that is no longer valid BEFORE the migration is run. This can be done over time to avoid the migration blackout period. To help facilitate this activity, the Batches tab lets you view batches by a selected status, then call the Batch window for the selected batch by clicking the drill down indicator. The appropriate window is called based on the version of 11*i* that is being run.

You must complete all closed batches prior to migrating to Release 12 so that any Financial associations can be completed. All WIP batches are unreleased and canceled, and all Pending and FPOs are canceled before the migration to Release 12. Once the migration is complete, you must run a concurrent request that recreates all WIP, Pending, and FPOs that were captured as part of the pre-migration snapshot.

When batches are migrated, any completed transactions against WIP batches are recreated as Material Transactions. The pending transactions against ingredients are created as Material Reservations and the pending transactions against products or byproducts are created as Pending Product Lots.

Process Batches for Migration can be clicked to create the snapshot of the appropriate batches in the system at any time. But, once it is clicked, those batches are no longer available for use in the system. Because this occurs in a Production environment, you must not complete the snapshot until you are sure you do not need to access these batches. It is suggested that you take the snapshot when you are in the blackout period. You can click Process Batches for Migration multiple times. Each time it is clicked, the current open batches are appended to the last snapshot.

If the snapshot is not taken, then pending batches and FPOs are not changed by the system while migrating data. You then have to manually close or cancel these batches. Migrated batches and firm planned orders display in query only mode. You cannot perform any additional material transactions or resource usage transactions on the migrated batches. No actions can be performed on the migrated batches except as described below.

Batch/FPO Status	**Actions Allowed**
PENDING	Users will be allowed to cancel the document.
WIP	Users will be allowed to terminate the document; Users cannot update the any other information.
COMPLETED	Users will be allowed to close this document. No other actions are allowed on this document. This document cannot be reopened once closed. Costing variances may not be accurate.

Note:

If the GL Update was run prior to migration, the inventory transactions and material transactions have already been accounted for. If the GL Update was NOT run prior to migration, the financial impact of the material transactions related to these batches will be lost, but the resource transactions will be accounted for once the GL Update is run.

You can select Mass Batch Close from the Action menu of the Batches tab to close all batches with a status of Completed. After the Mass Batch update has been run, you must re-query this form for any Completed batches. If any still appear, then you must address the issues that prevented the batches from being closed, and manually close the batch.

Note: ERES signatures are not gathered through the Mass Batch Close functionality. If it is a requirement that you must gather signatures, then you must close each individual batch manually from the Batches window.

Impact on Other Products

Process Quality

Any Samples and Sampling Events and WIP specification validity rules are updated with the batch ID, step Id and material detail line Id from the recreated batch. Any valid validity rules that are related to a recreated batch are copied and created for the recreated batch.

New Product Development

Any Lab Technical Data or Spreadsheet data related to batches that were recreated must be manually copied to the batch ID for the recreated batch.

Migrated batches can be queried in the simulator, but not updated.

Process Planning

A new data collection is triggered once the batches have been recreated in Release 12. This ensures that the data is retrieved from the new version of the batch.

E-Records and E-Signatures

Any batch related ERES information gathered in release 11*i* are associated with the old (renamed) batch, and are not copied to the recreated batch.

Attachments

New attachments records must be inserted for the recreated batches.

Order Management Migration

Order Fulfillment is not supported in R12, therefore an upgrade to Oracle Order Management, 11.5.10 is required before upgrading.

Order Fulfillment to Release 12, Order Management

Users using Order Fulfillment should first run the Order Fulfillment to Order Management migration. There are a series of pre-migration steps to setup for the migration. These steps can be found in the Order Fulfillment to Order Management Migration documentation available on Oracle Metalink. These documents are supported by release, so you must select the version appropriate for your current system. Open orders are migrated from Order Fulfillment to Order Management.

- Oracle Process Manufacturing Order Fulfillment to Order Management Migration (11.5.10) (October 2005 PDF 82 kb)
- Oracle Process Manufacturing Order Fulfillment to Order Management Migration (11.5.9) (May 2004 PDF 46 kb)
- Oracle Process Manufacturing Order Fulfillment to Order Management Migration (11.5.8) (May 2004 PDF 57 kb)
- Oracle Process Manufacturing Order Fulfillment to Order Management Migration (11.5.7 and lower) (August 2003 PDF 55 kb)

Once the migration from Order Fulfillment to OPM/OM is complete, the migration steps from OPM/OM to R12 must be completed.

OPM/OM to Release 12, Order Management

For OPM/OM users, migration will involve review of staged/picked order lines to either ship or backorder prior to OPM/OM in-line migration. Allocations made in OPM inventory will become Oracle Inventory Reservations with the completion of the migration process. Users should also make sure that the Interface Trip Stop concurrent process is run prior to the migration to ensure there are no pending interfaces

In Order Management, move orders in the process move orders tables are migrated into the Oracle Inventory tables. Pending transactions for non-pick confirmed order lines are migrated as reservations.

The following table includes the Business Process Stages in Sales Order Processing Cycle that are covered in the current design:

Booked	Scheduled	Allocated/ Picked	Pick Released	Pick Confirmed	Shipped	Invoiced	Migrated
X							Yes
	X						Yes
	X	X					Yes
X	X	X					Yes
X	X	X	X				Yes
X	X	X	X	X			No
X	X	X	X	X	X		No
X	X	X	X	X	X	X	No

Material Reservation Record Mapping

The following is the mapping of Record ic_tran_pnd to mtl_reservations:

RESERVATION_ID	NULL
REQUIREMENT_DATE	OE_ORDER_LINES_ALL.REQUEST_DATE
ORGANIZATION_ID	OE_ORDER_LINES_ALL.SHIP_FROM_ORG_ID
INVENTORY_ITEM_ID	OE_ORDER_LINES_ALL.INVENTORY_ITEM_ID
DEMAND_SOURCE_TYPE_ID	OE_ORDER_LINES_ALL.DEMAND_SOURCE_TYPE_ID
DEMAND_SOURCE_NAME	NULL
DEMAND_SOURCE_LINE_ID	OE_ORDER_LINES_ALL.LINE_ID
DEMAND_SOURCE_DELIVERY	NULL
PRIMARY_UOM_CODE	OE_ORDER_LINES_ALL.ORDER_QUANTITY_UOM
PRIMARY_UOM_ID	NULL
PRIMARY_RESERVATION_QUANTITY	OE_ORDER_LINES_ALL..ORDERED_QUANTITY
RESERVATION_UOM_CODE	OE_ORDER_LINES_ALL.ORDER_QUANTITY_UOM
RESERVATION_UOM_ID	NULL
RESERVATION_QUANTITY	OE_ORDER_LINES_ALL.ORDERED_QUANTITY
LOT_NUMBER	LOT_NO || SUBLOT_NO FROM IC_TRAN_PND

OPM Planning Migration

Forecasts

ONLY the forecasts that have a forecast set name entered for them will be migrated. Forecast set name field was added in 11.5.10 and there is a patch (4268525) for the releases prior to 11.5.10.

Replenishment Rules

OPM Replenishment Rules are replaced by attributes that defined on the MRP/MPS Planning Tab and Lead-times Tab of Organizational Items. There is no migration of this data because the meaning of the fields differs from MRP to ASCP unconstrained.

OPM System Administration Migration

Organizations

OPM Organizations are replaced by the organization structure in Oracle Applications. Information in this window, including Manufacturing Calendar and Resource Warehouse Code, can be found within different applications.

Addresses

OPM Addresses are now part of the Oracle Human Resources functionality, called HR Locations.

Geography Codes

After convergence, all address information must be entered in the HR Locations window. The HR Location model uses territory and other reference tables to find states, provinces, and countries for use with addresses.

HR Organizations

This is a menu option in OPM System Administration that provided a link to the existing HR Organizations window. After convergence, this menu item is removed, and the form is navigated to from Oracle Inventory, as well as several other applications.

HR Locations

This is a menu option in OPM System Administration that provided a link to the existing HR Locations window. After convergence, this menu item is removed, and the form is available from Oracle Inventory, as well as several other applications.

Document Ordering

After convergence, inventory transactions currently created and managed in OPM are replaced by transactions in Oracle Inventory. However, the Process Execution and OPM Quality Management applications continue to require Document Ordering. Document ordering for these two applications is available as setup under their respective system parameter menus and is maintained for an inventory organization.

Document Types

Many of the pre-seeded Document Types are not required for Inventory Convergence. Only three document types remain for Process Execution and samples and stability studies for OPM Quality Management.

User Definable Text

OPM provides an integrated approach to entering and viewing text that makes it quicker and easier for Formulators and Process Engineers to create instructions and standard operating procedures (SOPs).

For Release 12, OPM provides a Text Submenu as part of each standard application menu for the setup of Paragraph Codes and Text Tokens. Paragraph Codes are used to logically categorize text. They are table specific, and any table in OPM can have Paragraphs and Text associated to them. Text Tokens are abbreviations of longer text strings to make entering text easier.

Text can then be entered for records in Quality Management, Product Development, and Process Execution. In addition, the list of tables that can be associated to text is restricted to the product from which the window is accessed. Therefore, if the window is accessed from a Process Execution menu, only the tables for Process Execution are available.

For all other Oracle Applications, attachments are used to associate text or another object (picture, document, etc.) to a database record.

Reason Codes and Reason Code Security

Reason Codes and Reason Code Security are supported under Inventory Transactions Reasons in Oracle Inventory. Existing codes are moved to Inventory tables. The use of a Reason Code is dependent on the Document Types and Responsibilities it is associated with, and the Reason Code list of values in each window is filtered based on these associations. Similar to the Process Manufacturing functionality, you have the option to assign **All** Document Types and **All** Responsibilities when no security is required for a reason code, or the option to assign no responsibilities if no restriction is required for responsibilities for the specific document type.

Session Parameters

Session Parameters is replaced by the Change Organization function. This function is available in any Oracle application that requires the selection of an inventory organization for a transaction.

Units Of Measure

Units of measure are maintained under Oracle Inventory.

User Planning Classes

User Planning Classes are no longer supported for Process Manufacturers.

Workflow Setup

Workflow setup is included under OPM applications that support workflows.
The OPM Inventory Lot Expiry and Retest Workflow is replaced by an Oracle Inventory workflow. Setup for this workflow is done in Oracle Inventory and the Approvals Management Engine (AME).

Purge And Archive

The OPM Purge and Archive functionality continue in the common inventory model. Some of the seeded purges become obsolete and others are modified to query and archive the new Oracle Inventory tables. There is currently no archive feature in Oracle Inventory.

User Organizations

User Organizations are replaced by Organization Security functionality under Oracle Inventory. Formula Security is still supported under Process Product Development.

Table Mapping

This section contains the table mapping from OPM to Oracle Inventory. In addition, obsolete columns are denoted as such.

Inventory Tables

IC_ITEM_MST_B

The following table contains data mapping from OPM Item Master to Oracle Inventory Item Master table. Because the current Item Synchronization already creates the items in Oracle Inventory, the columns that need migration are highlighted in gray. However, the update script updates all process attributes to ensure the correct value for the process enabled organizations.

Source Column (in IC_ITEM_MST_B)	Data Type (*=Not Null)	Destination Column (in MTL_SYSTEM_ITEMS_B)	Date Type	Comments
ITEM_ID	*NUMBER(10)	INVENTORY_ITEM_ID	NUMBER	System generated id. Note that the id has the same value, if the item is assigned to multiple organizations.
		ORGANIZATION_ID	NUMBER	Only the process enabled organizations is affected by this migration (with the exception of the a new discrete master organization).
ITEM_NO	*VARCHAR2(32)	SEGMENT1	VARCHAR2(40)	OPM items are migrated to segment1 only.
ITEM_DESC1	VARCHAR2(70)	DESCRIPTION	VARCHAR2(240)	
		RECIPE_ENABLED_FLAG PROCESS_EXECUTION_ENABLED_FLAG PROCESS_COSTING_ENABLED_FLAG PROCESS_QUALITY_ENABLED_FLAG	VARCHAR2(1)	These columns do not exist in OPM. For the process-enabled organizations, they are updated to Y during migration.
ITEM_DESC2	VARCHAR2(70)	LONG_DESCRIPTION	VARCHAR2(4000)	We currently don’t synchronize this column. Note that this migration occurs if the LONG_DESCRIPTION column is null in discrete.
ALT_ITEMA	VARCHAR2(32)			Obsolete.
ALT_ITEMB	VARCHAR2(32)			Obsolete.

ITEM_UM	*VARCHAR2(4)	PRIMARY_UNIT_OF_MEASURE
		PRIMARY_UOM_CODE
		UNIT_OF_ISSUE
			VARCHAR2(25)
			VARCHAR2(3)
			VARCHAR2(25)
				Note that the migrated values are derived from the sy_uoms_mst view.

DUALUM_IND	*NUMBER(5)	TRACKING_QUANTITY_IND SECONDARY_DEFAULT_IND	VARCHAR2(30) VARCHAR2(30)	Dualum_ind Tracking_quantity_ind Secondary_default_ind 0 P NULL 1 PS F 2 PS D 3 PS N
ITEM_UM2	VARCHAR2(4)	SECONDARY_UOM_CODE	VARCHAR2(3)	Note that the migrated values are derived from the sy_uoms_mst view.
DEVIATION_LO	*NUMBER	DUAL_UOM_DEVIATION_LOW	NUMBER	= DEVIATION_LO * 100
DEVIATION_HI	*NUMBER	DUAL_UOM_DEVIATION_HIGH	NUMBER	= DEVIATION_HI * 100

ONT_PRICING_QTY_SOURCE	NUMBER(5)	ONT_PRICING_QTY_SOURCE	VARCHAR2(30)	= P, if ONT_PRICING_QTY_SOURCE = 0 = S, if ONT_PRICING_QTY_SOURCE = 1
LEVEL_CODE	NUMBER(5)			Obsolete, not supported in OPM.
LOT_CTL	*NUMBER(5)	LOT_CONTROL_CODE	NUMBER	= LOT_CTL + 1
LOT_INDIVISIBLE	*NUMBER(5)	LOT_DIVISIBLE_FLAG	VARCHAR2(1)	= Y if LOT_INDIVISIBLE = 0 = N if LOT_INDIVISIBLE = 1
AUTOLOT_ACTIVE_INDICATOR	NUMBER(1)			In Oracle Inventory, the auto lot numbering is always enabled for a lot or lot/child lot controlled items.
LOT_PREFIX	VARCHAR2(32)	AUTO_LOT_ALPHA_PREFIX	VARCHAR2(30)	LOT_PREFIX can be null in OPM. In case the value is NULL in IC_ITEM_MST_B, use the default value from column AUTO_LOT_ALPHA_PREFIX in GMI_MIGRATION_PARMS table.
LOT_SUFFIX	NUMBER(32)	START_AUTO_LOT_NUMBER	VARCHAR2(30)	LOT_SUFFIX can be null in OPM. In case the value is NULL in IC_ITEM_MST_B, use the default value from column START_AUTO_LOT_NUMBER in GMI_MIGRATION_PARMS table.

SUBLOT_CTL	*NUMBER(5)	CHILD_LOT_FLAG
		CHILD_LOT_VALIDATION_FLAG
		COPY_LOT_ATTRIBUTE_FLAG
		PARENT_CHILD_GENERATION_FLAG
			VARCHAR2(1)
			VARCHAR2(1)
			VARCHAR2(1)
				If SUBLOT_CTL = 1 THEN
				CHILD_LOT_FLAG = Y
				CHILD_LOT_VALIDATION_FLAG = Y
				COPY_LOT_ATTRIBUTE_FLAG = N
				PARENT_CHILD_GENERATION_FLAG=C

SUBLOT_PREFIX	VARCHAR2(32)	CHILD_LOT_PREFIX	VARCHAR2(30)	SUBLOT_PREFIX can be null in OPM. In case the value is NULL in IC_ITEM_MST_B, use the default value from column CHILD_LOT_PREFIX in GMI_MIGRATION_PARMS table.
SUBLOT_SUFFIX	NUMBER(32)	CHILD_LOT_STARTING_NUMBER	NUMBER	SUBLOT_SUFFIX can be null in OPM. In case the value is NULL in IC_ITEM_MSTn_B, use the default value from column CHILD_LOT_STARTING_NUMBER in GMI_MIGRATION_PARMS table.
LOCT_CTL	*NUMBER(5)	LOCATION_CONTROL_CODE	NUMBER	= LOCT_CTL + 1

NONINV_IND	*NUMBER(5)	INVENTORY_ITEM_FLAG
		INVENTORY_ASSET_FLAG
		COSTING_ENABLED_FLAG
		STOCK_ENABLED_FLAG
		BUILD_IN_WIP_FLAG
		MTL_TRANSACTIONS_ENABLED_FLAG
			VARCHAR2(1)
			VARCHAR2(1)
			VARCHAR2(1)
			VARCHAR2(1)
				IF NONINV_IND = 1
				Set all of these flags to N.
				ELSE
				No Change

MATCH_TYPE	*NUMBER(5)			Obsolete.

INACTIVE_IND	*NUMBER(5)	BUILD_IN_WIP_FLAG
		PURCHASING_ENABLED_FLAG
		MTL_TRANSACTIONS_ENABLED_FLAG
		STOCK_ENABLED_FLAG
		CUSTOMER_ORDER_ENABLED_FLAG
		INTERNAL_ORDER_ENABLED_FLAG
		INVOICE_ENABLED_FLAG
		RECIPE_ENABLED_FLAG
		PROCESS_EXECUTION_ENABLED_FLAG
		PROCESS_COSTING_ENABLED
		_FLAG
		PROCESS_QUALITY_ENABLED
		_FLAG
			VARCHAR2(1)	IF INACTIVE_FLAG = 1 THEN
				Set all these flags to N.
				ELSE
				No Change.

INV_TYPE	VARCHAR2(4)	ITEM_TYPE	VARCHAR2(30)	OPM Inventory types are migrated to Item Types in discrete. The Item Types are defined as Lookup Codes.

SHELF_LIFE	*NUMBER	SHELF_LIFE_CODE
		SHELF_LIFE_DAYS
			NUMBER
			NUMBER
				If(LOT_CTL = 0 or
				(LOT_CTL =1 and SHELF_LIFE = NULL))
				SHELF_LIFE_CODE = 1 (No Control)
				ELSE
				SHELF_LIFE_CODE = 2 (Shelf Life Days)
				SHELF_LIFE_DAYS = SHELF_LIFE

RETEST_INTERVAL	*NUMBER	RETEST_INTERVAL	NUMBER	
GL_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, GL_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
INV_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, INV_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.

SALES_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, SALES_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
SHIP_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, SHIP_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
FRT_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, FRT_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
PRICE_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, PRICE_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
STORAGE_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, STORAGE_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
PURCH_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, PURCH_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
TAX_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, GL_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
CUSTOMS_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, CUSTOMS_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
ALLOC_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, ALLOC_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
PLANNING_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, PLANNING_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
ITEMCOST_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, ITEMCOST_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
SEQ_DPND_CLASS	VARCHAR2(8)			All Item Classes are converged to Item categories. For example, SEQ_DPND_CLASS is assigned as a category in the MTL_ITEM_CATEGORIES table.
COST_MTHD_CODE	VARCHAR2(4)			Obsolete. Not on OPM form.
UPC_CODE	VARCHAR2(16)			There is no matching column in Oracle Inventory for this field. This column migrates as an obsolete column to the flexfield.
GRADE_CTL	*NUMBER(5)	GRADE_CONTROL_FLAG	VARCHAR2(1)	= Y if GRADE_CTL = 1

				= N if GRADE_CTL = 0
STATUS_CTL	*NUMBER(5)	LOT_STATUS_ENABLED	VARCHAR2(1)	= Y if STATUS_CTL = 1 = N if STATUS_CTL = 0
QC_GRADE	VARCHAR2(4)	DEFAULT_GRADE	VARCHAR2(150)	
LOT_STATUS	VARCHAR2(4)	DEFAULT_LOT_STATUS_ID	NUMBER	Note that the STATUS_ID is obtained from the IC_LOTS_STS.STATUS_ID . This column is populated in OPM by material status migration.
BULK_ID	NUMBER(10)			Obsolete. Not on OPM form.
PKG_ID	NUMBER(10)			Obsolete. Not on OPM form.
QCITEM_ID	NUMBER(10)			Obsolete.
QCHOLD_RES_CODE	VARCHAR2(4)			Obsolete.
EXPACTION_CODE	VARCHAR2(4)	EXPIRATION_ACTION_CODE	VARCHAR2(32)	
FILL_QTY	*NUMBER			Obsolete. Not on OPM form.
FILL_UM	VARCHAR2(4)			Obsolete. Not on OPM form.
EXPACTION_INTERVAL	*NUMBER	EXPIRATION_ACTION_INTERVAL	NUMBER	
PHANTOM_TYPE	*NUMBER(5)			Obsolete. Not on OPM form.
WHSE_ITEM_ID	*NUMBER(10)			Warehouse items were used in OPM Production for plant/warehouse rules. Going forward, these rules are not used by the OPM production in the convergence model. Warehouse items were also used in the OPM MRP module. This module is obsolete in the Converged model.
EXPERIMENTAL_IND	*NUMBER(5)	ENG_ITEM_FLAG	VARCHAR2(1)	= Y, if EXPERIMENTAL_IND = 1 = N, if EXPERIMENTAL_IND = 0
EXPORTED_DATE	*DATE			Obsolete. Not on OPM form.
TRANS_CNT	NUMBER(10)			Obsolete. Not on OPM form.
DELETE_MARK	*NUMBER(5)	See INACTIVE_IND		See INACTIVE_IND. Delete mark is migrated like inactive indicator.
TEXT_CODE	NUMBER(10)			Migrated as edit text.
COMMODITY_CODE	VARCHAR2(9)			Obsolete. A separate migration script existing in OPM from release 11.5.9 onwards which migrates the commodity code as Item Categories and assigns them to Items. No separate migration is needed for this column.
ITEM_ABCCODE	VARCHAR2(4)			The assignment of ABC code in OPM item master is not functional. OPM Physical Inventory uses ABC code from the warehouse rules.

				<>Since the Physical Inventory will be replaced with the Oracle Discrete Physical Inventory, this column doesn’t need to be converged. However, the actual ABC Rank (Classes in discrete) can be converged to the ABC Class master table in discrete. The assignment of the ABC class to item is more complex in discrete and cannot be automatically determined using the OPM setup.
ALLOC_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
CUSTOMS_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
FRT_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
GL_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
INV_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
COST_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
PLANNING_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
PRICE_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
PURCH_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
SALES_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
SEQ_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item

				category in discrete.
SHIP_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
STORAGE_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
TAX_CATEGORY_ID	NUMBER			This category assignments already exist in Item Categories in discrete. If not, then it migrates as an item category in discrete.
Who Columns	*DATE			Standard Who columns
ATTRIBUTE1-30	VARCHAR2(240)	ATTRIBUTE1-30	VARCHAR2(240)	The flexfield columns are migrated if the flexfields are not defined in Discrete Item Master or they are defined but do not conflict with the OPM Item Master flexfield definition. Otherwise, an error is reported in the validation script.
ATTRIBUTE_CATEGORY	VARCHAR2(30)	ATTRIBUTE_CATEGORY	VARCHAR2(30)	This column is migrated only if the attribute category is blank in the discrete item master. Otherwise, an error is reported in the validation script.

IC_PLNT_INV

Source Column **(in IC_PLNT_INV)**	**Data Type** **(*=Not Null)**	**Destination Column** **(in MTL_SYSTEM_ITEMS)**	**Date Type**	**Comments**
ITEM_ID	*NUMBER(10)	INVENTORY_ITEM_ID	NUMBER	System generated id. Note that the id has the same value, if the item is assigned to multiple organizations.
ORGN_CODE	VARCHAR2(4)	ORGANIZATION_ID	NUMBER	Note that the lead time is migrated only if 1.There is record for the item and organization of the OPM warehouse (corresponding to discrete organization) or 2.There is a record for the item and a NULL organization.
FIXED_LEADTIME	*NUMBER	FIXED_LEAD_TIME	NUMBER	
VARIABLE_LEADTIME	*NUMBER	VARIABLE_LEAD_TIME	NUMBER	= VARIABLE_LEADTIME / STD_QTY

IC_ITEM_CPG

Source Column (in IC_ITEM_CPG)	**Data Type (*=Not Null)**	**Destination Column (in MTL_SYSTEM_ITEMS)**	**Date Type**	**Comments**
ITEM_ID	*NUMBER(10)	INVENTORY_ITEM_ID	NUMBER	System generated id. Note that the id has the same value, if the item is assigned to multiple organizations.
		ORGANIZATION_ID	NUMBER	Only the process enabled organizations will be affected by this migration.
IC_MATR_DAYS	NUMBER(5)	MATURITY_DAYS	NUMBER	
IC_HOLD_DAYS	NUMBER(5)	HOLD_DAYS	NUMBER	
Who Columns		Who columns		

IC_ITEM_MST_TL

Source Column (in IC_ITEM_MST_tl)	**Data Type (*=Not Null)**	**Destination Column (in MTL_SYSTEM_ITEMS_tl)**	**Date Type**	**Comments**
ITEM_ID	*NUMBER(10)	INVENTORY_ITEM_ID	NUMBER	System generated id. Note that the id has the same value, if the item is assigned to multiple organizations.
		ORGANIZATION_ID	NUMBER	Only the process enabled organizations will be affected by this migration.
LANGUAGE	*VARCHAR2(4)	LANGUAGE	VARCHAR2(4)	
SOURCE_LANG	*VARCHAR2(4)	SOURCE_LANG	VARCHAR2(4)	
ITEM_DESC1	*VARCHAR2(70)	DESCRIPTION	VARCHAR2(240)	
ITEM_DESC2	VARCHAR2(70)	LONG_DESCRIPTION	VARCHAR2(4000)	We currently don’t synchronize this column. Note that this migration will only happen if the LONG_DESCRIPTION column is null in discrete.
Who Columns		Who columns		

GMI_ITEM_CATEGORIES

Source Column (in GMI_ITEM_CATEGORIES)	**Data Type (*=Not Null)**	**Destination Column (in MTL_ITEM_CATEGORIES)**	**Date Type**	**Comments**
ITEM_ID	*NUMBER	INVENTORY_ITEM_ID	NUMBER	
		ORGANIZATION_ID	NUMBER	
CATEGORY_SET_ID	*NUMBER	CATEGORY_SET_ID	NUMBER	
CATEGORY_ID	*NUMBER	CATEGORY_ID	NUMBER	
Who Columns		Who columns		

GMI_CATEGORY_SETS

Source Column (in gmi_category_sets)	**Data Type (*=Not Null)**	**Destination Column (in MTL_DEFAULT_ CATEGORY_SETS)**	**Date Type**	**Comments**
OPM_CLASS	*VARCHAR2(40)	FUNCTIONAL_AREA_ID	NUMBER	
CATEGORY_SET_ID	NUMBER	CATEGORY_SET_ID	NUMBER	
USER_OPM_CLASS	*VARCHAR2(40)			
RESTRICTED	VARCHAR2(1)			
Who Columns		Who columns		

IC_INVN_TYP

Source Column (in IC_INVN_TYP)	**Data Type (*=Not Null)**	**Destination Column (in FND_LOOKUP_VALUES)**	**Date Type**	**Comments**
		LOOKUP_TYPE	*VARCHAR2(30)	= “ITEM TYPE”
		LANGUAGE	*VARCHAR2(30)	MLS setup
		SOURCE_LANG	*VARCHAR2(4)	MLS setup
		SECURITY_GROUP_ID	*NUMBER(15)	0
		VIEW_APPLICATION_ID	*NUMBER(15)	3 (AU)
		TERRITORY_CODE	VARCHAR2(2)	NULL
		TAG	VARCHAR2(30)	NULL
INV_TYPE	*VARCHAR2(4)	LOOKUP_CODE	*VARCHAR2(30)	
INV_TYPE	*VARCHAR2(4)	MEANING	*VARCHAR2(80)	
INV_TYPE_DESC	*VARCHAR2(40)	DESCRIPTION	VARCHAR2(240)	
DELETE_MARK	*NUMBER(5)	ENABLED_FLAG	*VARCHAR2(1)	= Y, if Delete Mark = 0 = N, if Delete Mark = 1

		START_DATE_ACTIVE	DATE	NULL
		END_DATE_ACTIVE	DATE	NULL
Who Columns		Who Columns		
ATTRIBUTE_CATEGORY	VARCHAR2(30)	ATTRIBUTE_CATEGORY	VARCHAR2(30)	
ATTRIBUTE1-30	VARCHAR2(240)	ATTRIBUTE1-15	VARCHAR2(150)	
		TAG	VARCHAR2(30)	

IC_LOTS_MST

<table>
<tr><th>Source Column
(in IC_LOTS_MST)</th><th>Data Type
(*=Not Null)</th><th>Destination Column
(in MTL_LOT_NUMBERS)</th><th>Date Type</th><th colspan="4">Comments</th></tr>
<tr><td>ITEM_ID</td><td>*NUMBER(10)</td><td>INVENTORY_ITEM_ID</td><td>NUMBER</td><td colspan="4"></td></tr>
<tr><td></td><td></td><td>ORGANIZATION_ID</td><td>NUMBER</td><td colspan="4"></td></tr>
<tr><td>LOT_ID</td><td>*NUMBER(10)</td><td></td><td></td><td colspan="4">No migration needed because discrete uses lot_number as part of the key.</td></tr>
<tr><td rowspan="5">LOT_NO</td><td rowspan="5">*VARCHAR2(32)</td><td rowspan="5">PARENT_LOT_NUMBER
LOT_NUMBER</td><td rowspan="5">VARCHAR2(80)</td><td colspan="4">The following table shows the logic to derive the lot number. Note that additional logic to create new lots for lot status may be needed as discussed in the overview section of this migration. Also, the lot number can already exist in ic_lots_mst_mig table from the pre-migration form.</td></tr>
<tr><td><b>Sublot Control</b></td><td><b>Sublot No</b></td><td><b>Parent Lot</b></td><td><b>Lot (Child)</b></td></tr>
<tr><td>Yes</td><td>NULL</td><td>NULL</td><td>LOT_NO</td></tr>
<tr><td>Yes</td><td>Not NULL</td><td>LOT_NO</td><td>LOT_NO \|\| SUBLOT_NO</td></tr>
<tr><td>No</td><td>-</td><td>NULL</td><td>LOT_NO</td></tr>
<tr><td>SUBLOT_NO</td><td>VARCHAR2(32)</td><td>PARENT_LOT_NUMBER
LOT_NUMBER</td><td>VARCHAR2(80)</td><td colspan="4">Same as above</td></tr>
<tr><td>LOT_DESC</td><td>VARCHAR2(40)</td><td>DESCRIPTION</td><td>VARCHAR2(256)</td><td colspan="4">Description column in discrete inventory is non-functional. However, we will migrate the OPM Lot description to this column.</td></tr>
<tr><td>QC_GRADE</td><td>VARCHAR2(4)</td><td>GRADE_CODE</td><td>VARCHAR2(150)</td><td colspan="4">Foreign Key reference to the MTL_GRADES table.</td></tr>
<tr><td></td><td></td><td>STATUS_ID</td><td>NUMBER</td><td colspan="4">The LOT_STATUS from IC_LOCT_INV for this lot will be used to derive this value. See above for the example and discussion on case where a lot exists in multiple location</td></tr>
<tr><td></td><td></td><td>AVAILABILITY_TYPE</td><td>NUMBER</td><td colspan="4">This value will be derived from the MTL_MATERIAL_STATUSES for the STATUS_ID of the lot.</td></tr>
<tr><td></td><td></td><td>INVENTORY_ATP_CODE</td><td>NUMBER</td><td colspan="4">This value will be derived from the MTL_MATERIAL_STATUSES for the STATUS_ID of the lot.</td></tr>
<tr><td></td><td></td><td>RESERVABLE_TYPE</td><td>NUMBER</td><td colspan="4">This value will be derived from the MTL_MATERIAL_STATUSES for the STATUS_ID of the lot.</td></tr>
<tr><td>EXPACTION_CODE</td><td>VARCHAR2(4)</td><td>EXPIRATION_ACTION_CODE</td><td>VARCHAR2(32)</td><td colspan="4">Foreign Key reference to the MTL_ACTIONS table.</td></tr>
</table>

EXPACTION_DATE	DATE	EXPIRATION_ACTION_DATE	DATE	
LOT_CREATED	DATE	ORIGINATION_DATE	DATE	
EXPIRE_DATE	DATE	EXPIRATION_DATE	DATE	
RETEST_DATE	DATE	RETEST_DATE	DATE	
STRENGTH	*NUMBER			Obsolete Column, not used in OPM.
INACTIVE_IND	*NUMBER(5)	DISABLE_FLAG	NUMBER	= 1, if INACTIVE_IND = 1 = NULL, otherwise
ORIGINATION_TYPE	*NUMBER(5)	ORIGINATION_TYPE	NUMBER(2)	
SHIPVEND_ID	NUMBER(10)	VENDOR_ID VENDOR_NAME	VARCHAR2(240) NUMBER	Use this to get the discrete vendor_id select of_vendor_id from po_vend_mst where vendor_id = <SHIPVEND_ID>
VENDOR_LOT_NO	VARCHAR2(32)	SUPPLIER_LOT_NUMBER	VARCHAR2(150)	
Who Columns		Who Columns		
DELETE_MARK	*NUMBER(5)	DISABLE_FLAG	NUMBER	= 1, if DELETE_MARK = 1 = NULL, otherwise
TEXT_CODE	NUMBER(10)			Open Issue: GMA team is working on providing the text code migration for the OPM objects.
ATTRIBUTE1-30	VARCHAR2(240)			The flexfield columns are migrated if the flexfields are not defined in Discrete Lot Master or they are defined but do not conflict with the OPM Lot Master flexfield definition. Otherwise, an error is reported in the validation script.
ATTRIBUTE_CATEGORY	VARCHAR2(30)			This column is migrated only if the attribute category is blank in the discrete item master. Otherwise, an error is reported in the validation script.
ODM_LOT_NUMBER	VARCHAR2(30)			Obsolete in convergence.

GMD_GRADES_B

Source Column **(in GMD_GRADES_B)**	**Data Type** **(*=Not Null)**	**Destination Column** **(in MTL_GRADES_B)**	**Date Type**	**Comments**
QC_GRADE	*VARCHAR2(4)	GRADE_CODE	VARCHAR2(150)	
QC_GRADE_DESC	VARCHAR2(40)			Description is migrated to the TL table only.
DELETE_MARK	*NUMBER(5)	DISABLE_FLAG	VARCHAR2(1)	= ‘Y’ if delete_mark = 1 = ‘N’ otherwise

TEXT_CODE	NUMBER(10)			Migrates as edit text.
ATTRIBUTE1-30	VARCHAR2(240)	ATTRIBUTE1-30	VARCHAR2(240)	
ATTRIBUTE_CATEGORY	VARCHAR2(30)	ATTRIBUTE_CATEGORY	VARCHAR2(30)	
Who Columns		Who Columns		

GMD_GRADES_TL

Source Column	**Data Type**	**Destination Column**	**Date Type**	**Comments**
(in GMD_GRADES_TL)	**(*=Not Null)**	**(in MTL_GRADES_TL)**		
QC_GRADE	*VARCHAR2(4)	GRADE_CODE	VARCHAR2(150)	
QC_GRADE_DESC	*VARCHAR2(40)	DESCRIPTION	VARCHAR2(240)	
SOURCE_LANG	*VARCHAR2(4)	SOURCE_LANG	VARCHAR2(4)	
LANGUAGE	*VARCHAR2(4)	LANGUAGE	VARCHAR2(4)	
Who Columns		Who Columns		

GMD_ACTIONS_B

Source Column	**Data Type**	**Destination Column**	**Date Type**	**Comments**
(in GMD_ACTIONS_B)	**(*=Not Null)**	**(in MTL_ACTIONS_B)**		
ACTION_CODE	*VARCHAR2(32)	ACTION_CODE	VARCHAR2(32)	
ACTION_INTERVAL	*NUMBER			Currently Non-functional in OPM. Obsolete in the converged model.
DELETE_MARK	*NUMBER(5)	DISABLE_FLAG	VARCHAR2(1)	= ‘Y’ if delete_mark = 1 = ‘N’ otherwise
TEXT_CODE	NUMBER(10)			Open Issue: Need text code migration from the GMA team.
ATTRIBUTE1-30	VARCHAR2(240)	ATTRIBUTE1-30	VARCHAR2(240)	
ATTRIBUTE_CATEGORY	VARCHAR2(30)	ATTRIBUTE_CATEGORY	VARCHAR2(30)	
Who Columns		Who Columns		

GMD_ACTIONS_TL

Source Column	**Data Type**	**Destination Column**	**Date Type**	**Comments**
(in GMD_ACTIONS_TL)	**(*=Not Null)**	**(in MTL_ACTIONS_TL)**		
ACTION_CODE	*VARCHAR2(32)	ACTION_CODE	VARCHAR2(32)	
DESCRIPTION	*VARCHAR2(80)	DESCRIPTION	VARCHAR2(80)	
SOURCE_LANG	*VARCHAR2(4)	SOURCE_LANG	VARCHAR2(4)	
LANGUAGE	*VARCHAR2(4)	LANGUAGE	VARCHAR2(4)	

Who Columns		Who Columns		

IC_LOTS_STS to MTL_MATERIAL_STATUSES_B

Source Column **(in IC_LOTS_STS)**	**Data Type** **(*=Not Null)**	**Destination Column (in** **MTL_MATERIAL_STATUSES_B)**	**Date Type**	**Comments**
LOT_STATUS	*VARCHAR2(4)			Will be migrated to mtl_material_statuses_tl table.
		STATUS_ID		System generated surrogate for the status.
STATUS_DESC	*VARCHAR2(40)			Will be migrated to mtl_material_statuses_tl table.
QCHOLD_RES_CODE	VARCHAR2(4)			Obsolete
		ZONE_CONTROL	NUMBER	= 2 (Means No) Zone refers to the Subinventory Status control
		LOCATOR_CONTROL	NUMBER	= 2 (No)
		LOT_CONTROL	NUMBER	= 1 (Yes)
		SERIAL_CONTROL	NUMBER	= 2 (No)
NETTABLE_IND	*NUMBER(5)	AVAILABILITY_TYPE	NUMBER	= 1, if NETTABLE_IND=1 & REJECTED_IND=0 = 2, otherwise
ORDER_PROC_IND	*NUMBER(5)		NUMBER	T=It will mapped to the allowed/disallowed transaction types in mtl_status_transaction_control as show below in the data mapping.
PROD_IND	*NUMBER(5)		NUMBER	Same as above
SHIPPING_IND	*NUMBER(5)		NUMBER	Same as above
REJECTED_IND	*NUMBER(5)	RESERVABLE_TYPE INVENTORY_ATP_CODE	NUMBER NUMBER	= 1, if NETTABLE_IND = 1 = 2, otherwise
DELETE_MARK	*NUMBER(5)	ENABLED_FLAG	NUMBER	= 1, if DELETE_MARK = 0 = 2 otherwise
TEXT_CODE	NUMBER(10)			Open Issue: Need text code migration from the GMA team.
Who Column		Who Column		

IC_LOTS_STS to MTL_MATERIAL_STATUSES_TL

Source Column **(in IC_LOTS_STS)**	**Data Type** **(*=Not Null)**	**Destination Column (in** **MTL_MATERIAL_STATUSES_TL)**	**Date Type**	**Comments**
LOT_STATUS	*VARCHAR2(4)	STATUS_CODE	VARCHAR2(80)	Will be migrated to mtl_material_statuses_tl table.
		STATUS_ID	NUMBER	System generated surrogate for the

				status.
STATUS_DESC	*VARCHAR2(40)	DESCRIPTION	VARCHAR2(240)	Will be migrated to mtl_material_statuses_tl table.
		LANGUAGE	VARCHAR2(4)	Determine by MLS package when called to insert a row.
		SOURCE_LANG	VARCHAR2(4)	Determine by MLS package when called to insert a row.
Who Column		Who Column		

IC_LOTS_STS to MTL_STATUS_TRANSACTION_CONTROL

Source Column **(in IC_LOTS_STS)**	**Data Type** **(*=Not Null)**	**Destination Column (in** **MTL_STATUS_TRANSACTION_CONTROL)**	**Date Type**	**Comments**
LOT_STATUS	*VARCHAR2(4)			
		STATUS_ID	NUMBER	System generated surrogate for the status.
		IS_ALLOWED	NUMBER	= 1, (Yes) if OPM indicators are 1 = 2, (No) otherwise.
NETTABLE_IND	*NUMBER(5)			
ORDER_PROC_IND	*NUMBER(5)	TRANSACTION_TYPE_ID	NUMBER	This indicator allows or disallows transaction type based on the Lot Status.
PROD_IND	*NUMBER(5)	TRANSACTION_TYPE_ID	NUMBER	This indicator allows or disallows transaction type based on the Lot Status.
SHIPPING_IND	*NUMBER(5)	TRANSACTION_TYPE_ID	NUMBER	This indicator allows or disallows transaction type based on the Lot Status.
REJECTED_IND	*NUMBER(5)	TRANSACTION_TYPE_ID	NUMBER	This indicator allows or disallows transaction type based on the Lot Status..
Who Column				

IC_ITEM_CNV

Source Column **(in IC_ITEM_CNV)**	**Data Type** **(*=Not Null)**	**Destination Column (in** **MTL_LOT_UOM_CLASS** **_CONVERSIONS)**	**Date Type**	**Comments**
		ORGANIZATION_ID	*NUMBER	Call the get_ODM_lot routine to get (or create) the discrete item/ lot for the OPM item/ lot.
LOT_ID	*NUMBER(10)	LOT_NUMBER	*VARCHAR2(80)	Call the get_ODM_lot routine to get (or create) the discrete

				item/ lot for the OPM item/ lot.
ITEM_ID	*NUMBER(10)	INVENTORY_ITEM_ID	*NUMBER	Call the get_ODM_lot routine to get (or create) the discrete item/ lot for the OPM item/ lot.
		FROM_UNIT_OF_MEASURE	*VARCHAR2(25)	Select the value for the base UOM of the UOM Class of the Item’s primary UOM. Select b.UNIT_OF_MEASURE, b.UOM_CODE, b.UOM_CLASS from mtl_system_items_b I, mtl_units_of_measure iu, mtl_units_of_measuer bu where i.organication_id = <organization_id> and i.inventory_item_id = <inventory_item_id> and i.PRIMARY_UOM_CODE = iu.uom_Code and iu.uom_class = bu.uom_class and bu.base_uom_flag = ‘Y’
		FROM_UOM_CODE	*VARCHAR2(3)	Derive the value using the above logic
		FROM_UOM_CLASS	*VARCHAR2(10)	UOM Class of the item’s primary UOM.

		TO_UNIT_OF_MEASURE	*VARCHAR2(25)	Select the base unit of measure for the UM_TYPE.
				SELECT UNIT_OF_MEASURE, UOM_CODE
				FROM MTL_UNITS_OF_MEASURE

				WHERE
				UOM_CLASS = <UM_TYPE> and
				BASE_UOM_FLAG = ‘Y’

		TO_UOM_CODE	*VARCHAR2(3)	Call the get_ODM_lot routine to get (or create) the discrete item/ lot for the OPM item/ lot.
UM_TYPE	*VARCHAR2(10)	TO_UOM_CLASS	*VARCHAR2(10)	
TYPE_FACTOR	*NUMBER	CONVERSION_RATE	*NUMBER	
TYPE_FACTORREV	*NUMBER			Discrete tables do not store the reverse factors
CONVERSION_ID	*NUMBER			System generated surrogate using the MTL_CONVERSION_ID_S sequence.
EVENT_SPEC_DISP_ID	NUMBER	EVENT_SPEC_DISP_ID	NUMBER	
DELETE_MARK	*NUMBER(5)	DISABLE_DATE	DATE	= LAST_UPDATE_DATE, if DELETE_MARK = 1 = NULL, otherwise.
TEXT_CODE	NUMBER(10)			Migrates as edit text.
Who Columns		Who Columns		

GMI_ITEM_CONV_AUDIT

Source Column	**Data Type**	**Destination Column (in**	**Date Type**	**Comments**
(in GMI_ITEM_CONV_AUDIT)	**(*=Not Null)**	**MTL_LOT_CONV_AUDIT)**		
CONV_AUDIT_ID	*NUMBER	CONV_AUDIT_ID	*NUMBER	System generated surrogate using the MTL_CONV_AUDIT_ID_S sequence.
CONVERSION_ID	*NUMBER	CONVERSION_ID	*NUMBER	Use the CONVERSION_ID from the MTL_LOT_UOM_CLASS_CONMVERSIONS table.
CONVERSION_DATE	*DATE	CONVERSION_DATE	*DATE	
REASON_CODE	VARCHAR2(4)	REASON_ID	NUMBER	REASON_ID from the MTL_TRANSACTION_REASONS table for the REASON_CODE
OLD_TYPE_FACTOR	NUMBER	OLD_CONVERSION_RATE	NUMBER	
NEW_TYPE_FACTOR	*NUMBER	NEW_CONVERSION_RATE	*NUMBER	
EVENT_SPEC_DISP_ID	NUMBER	EVENT_SPEC_DISP_ID	NUMBER	
BATCH_ID	NUMBER(10)	BATCH_ID	NUMBER	
UPDATE_BATCH_INDICATOR	NUMBER(10)	UPDATE_TYPE_INDICATOR	*NUMBER	
Who Columns		Who Columns		

GMI_ITEM_CONV_AUDIT_DETAILS

Source Column	**Data Type**	**Destination Column (in**	**Date Type**	**Comments**
(in GMI_ITEM_CONV_AUDIT	**(*=Not Null)**	**MTL_LOT_CONV_AUDIT**		
_DETAILS)		**_DETAILS)**		
CONV_AUDIT_DETAIL_ID	*NUMBER	CONV_AUDIT_DETAIL_ID	*NUMBER	System generated surrogate using the MTL_CONV_AUDIT_DETAIL_ID_S sequence
CONV_AUDIT_ID	*NUMBER	CONV_AUDIT_ID	*NUMBER	CONV_AUDIT_ID from the MTL_LOT_CONV_AUDIT table.
		REVISION	VARCHAR2(3)	= NULL
WHSE_CODE	*VARCHAR2(4)	ORGANIZATION_ID	*NUMBER	= ORGANIZATION_ID from IC_WHSE_MST for the WHSE_CODE.
		SUBINVENTORY_CODE	VARCHAR2(10)	= WHSE_CODE
		LPN_ID	NUMBER	NULL
LOCATION	*VARCHAR2(16)	LOCATOR_ID	NUMBER	= INVENTORY_LOCATION_ID from IC_LOCT_MST for the WHSE_CODE, LOCATION.
OLD_ONHAND_QTY	*NUMBER	OLD_PRIMARY_QTY	*NUMBER	
OLD_ONHAND_QTY2	NUMBER	OLD_SECONDARY_QTY	NUMBER	
NEW_ONHAND_QTY	NUMBER	NEW_PRIMARY_QTY	NUMBER	
NEW_ONHAND_QTY2	NUMBER	NEW_SECONDARY_QTY	NUMBER	
TRANS_QTY	NUMBER	TRANSACTION_PRIMARY_QTY	NUMBER	
TRANS_QTY2	NUMBER	TRANSACTION_SECONDARY_QTY	NUMBER	

TRANS_UPDATE_FLAG	*VARCHAR2(1)	TRANSACTION_UPDATE_FLAG	*VARCHAR2(2)	
Who Columns		Who Columns		

CREATE_TRANSFER_PUB

This table is mapping to the API call.

Destination Column (in Create_transfer_pub)	In/ Out	Data Type	Date Type	Comments
P_api_version	IN	NUMBER		
P_init_msg_list	IN	VARCHAR2 DEFAULT FND_API.G_FALSE		
P_commit	IN	VARCHAR2 DEFAULT FND_API.G_FALSE		
P_validation_level	IN	NUMBER DEFAULT FND_API.G_VALID_LEVEL_FULL		
X_return_status	OUT NOCOPY	VARCHAR2		
X_msg_count	OUT NOCOPY	NUMBER		
X_msg_data	OUT NOCOPY	VARCHAR2		
P_hdr_rec	IN	GMIVDX.hdr_type		
P_line_rec_tbl	IN	GMIVDX.line_type_tbl		
P_lot_rec_tbl	IN	GMIVDX.lot_type_tbl		
x_hdr_row	OUT NOCOPY	gmi_discrete_transfers%ROWTYPE		
x_line_row_tbl	OUT NOCOPY	GMIVDX.line_row_tbl		
x_lot_row_tbl	OUT NOCOPY	GMIVDX.lot_row_tbl		

IC_LOT_INV to Header Type

This table is mapping to the API call.

Source Column (in IC_LOCT_INV)	Data Type (*=Not Null)	Destination (in HDR_TYPE record)	Comments
		orgn_code	= orgn_code of the inventory balance warehouse.
		co_code	API will internally determine the values when determining the charge account.
		transfer_number	Automatically generated number
		transfer_type	= 0 (Process to Discrete) = 1 (Discrete to Process) used for transferring negative balances.
		trans_date	= sysdate
		comments	‘OPM Convergence Migration’
		attribute_category	NULL
		attribute1-30	NULL
		assignment_type	NULL

		transaction_header_id	NULL

IC_LOCT_INV to Line Type

This table is mapping to the API call.

Source Column (in IC_LOCT_INV)	Data Type (*=Not Null)	Destination (in LINE_TYPE record)	Comments
		line_no	Automatically generated line number for each transfer. For a single transfer multiple lines can be specified and they start with 1 for each transfer.
ITEM_ID	NUMBER(10)	opm_item_id	
WHSE_CODE	VARCHAR2(4)	opm_whse_code	
LOCATION	VARCHAR2(16)	opm_location	
LOT_ID	NUMBER(10)	opm_lot_id	= 0 for non-lot controlled items. = NULL for lot controlled items. For lot controlled items, the LOT_TYPE record will be used to specify multiple lots.
		opm_lot_expiration_date	API internally determines the values.
LOT_STATUS	VARCHAR2(4)	opm_lot_status	API internally determines the values.
		opm_grade	API internally determines the values.
		opm_charge_acct_id	API internally determines the values.
		opm_charge_au_id	API internally determines the values.
		opm_reason_code	NULL
		odm_inv_organization_id	MTL_ORGANIZATION_ID from the IC_WHSE_MST for the OPM warehouse.
		odm_item_id	Call the get_ODM_item to get Discrete Item from discrete inventory for the OPM Item and the warehouse.
		odm_item_revision	NULL. No OPM items are revision controlled.
WHSE_CODE	VARCHAR2(4)	odm_subinventory	Determine the subinventory associated to the stock locator linked to the OPM location. If the item is not location controlled, then whse_code will be used as the subinventory. IN most cases this should be the (the default subinventory created by OPM trigger during the warehouse/inventory organization creation in discrete.)
		odm_locator_id	INVENTORY_LOCATION_ID from the IC_LOCT_MST for the OPM Warehouse and Location.
		odm_lot_number	LOT_NUMBER from the IC_LOTS_MST_MIG table for the Inventory organization_id and OPM LOT_ID.

		odm_lot_expiration_date	API internally determines the values.
		odm_charge_account_id	API internally determines the values.
		odm_period_id	API internally determines the values.
		odm_unit_cost	NULL
		odm_reason_id	NULL
LOCT_ONHAND	NUMBER	Quantity	Only for non-lot controlled items. For lot controlled items, multiple lots can be specified in the lot_type record and this quantity should be the sum of the lot quantity.
		Quantity_um	= ITEM_UM from ic_item_mst_b for the item_id.
LOCT_ONHAND2	NUMBER	Quantity2	Only for non-lot controlled items. For lot controlled items, multiple lots can be specified in the lot_type record and this quantity should be the sum of the lot quantity.
		opm_primary_quantity	API internally determines the values.
		odm_primary_quantity	API internally determines the values.
		lot_level	API internally determines the values.
		attribute_category	NULL
		attribute1-30	NULL
		opm_item_no	API internally determines the values.
		lot_control	API internally determines the values.
		odm_quantity_uom_code	API internally determines the values.
		opm_lot_no	API internally determines the values.
		opm_sublot_no	API internally determines the values.

IC_LOCT_INV to Lot Type

This table is mapping to the API call:

Source Column **(in IC_LOCT_INV)**	**Data Type** **(*=Not Null)**	**Destination (in** **LOT_TYPE record)**	**Comments**
		Line_no	The line_no in the LINE_TYPE record for which the lot data is being specified.
LOT_ID	NUMBER(10)	opm_lot_id	Call the get_ODM_lot to get Discrete Item/ Lot from the discrete inventory for the OPM Items and the warehouse.
		opm_lot_expiration_date	API internally determines the values.
		opm_lot_status	API internally determines the values.
		opm_grade	API internally determines the values.

		odm_lot_number	LOT_NUMBER from the IC_LOTS_MST_MIG table for the Inventory organization_id and OPM LOT_ID.
		odm_lot_expiration_date	API internally determines the values.
LOCT_ONHAND	NUMBER	Quantity	
LOCT_ONHAND2	NUMBER	quantity2	
LOCT_ONHAND	NUMBER	opm_primary_quantity	
LOCT_ONHAND	NUMBER	odm_primary_quantity	
		opm_lot_no	API internally determines the values.
		opm_sublot_no	API internally determines the values.

Organization Column Mapping

HR_LOCATIONS_ALL

The address associated with the organization is created in this table.

Column	Value
location_code	Orgn_Code
address_line_1	Addr1
address_line_2	Addr2
address_line_3	Addr3
bill_to_site_flag	NULL
country	Country_Code
description	Orgn_Name
designated_receiver_id	NULL
in_organization_flag	NULL
inactive_date	NULL
inventory_organization_id	Migrated Organization_Id from Hr_Organization_Units this is updated after migrating the organization.
office_site_flag	NULL
postal_code	Postal_Code
receiving_site_flag	NULL
region_1	Province
region_2	County
region_3	State_Code
ship_to_location_id	NULL
ship_to_site_flag	NULL
derived_locale	NULL
style	‘OPM’
tax_name	NULL
telephone_number_1	NULL
telephone_number_2	NULL
telephone_number_3	NULL
town_or_city	Addr4

HR_ORGANIZATION_UNITS

The HR organization is created in this table.

Column	Value
organization_id	hr_organization_units_s.nextval
business_group_id	Hr_operating_unts.business_group_id based on the fiscal policy operating unit.
cost_allocation_keyflex_id	NULL
location_id	Location_Id generated from the above insert.
soft_coding_keyflex_id	NULL
date_from	Creation_Date
name	Orgn_Name
comments	NULL
date_to	Set to SYSDATE if the org is being migrated as Inactive.
internal_external_flag	‘INT’
internal_address_line	NULL
type	NULL

HR_ORGANIZATION_INFORMATION

The HR organization classification for inventory organizations is created in this table if the Inventory_Org indicator is 1.

Column	Value
org_information_id	hr_organization_information_s.nextval
org_information_context	‘CLASS’
organization_id	Organization_Id generated above
org_information1	‘INV’
org_information2	‘Y’
org_information3	NULL

After classifying the accounting information, it is created in this table.

Column	Value
org_information_id	hr_organization_information_s.nextval
org_information_context	‘Accounting Information’
organization_id	Organization_Id generated above
org_information1	Set_Of_Books ID from the template organization
org_information2	Legal Entity ID from the template organization
org_information3	Operation Unit from the template organization

MTL_PARAMETERS

The inventory organization is created in this table.

Column	Value
organization_id	Organization_Id generated above
organization_code	Organization_Code from the Setup
master_organization_id	Master Organization Id from the setup

Column	Value
primary_cost_method	Inherited from the template organization
cost_organization_id	Inherited from the template organization
default_material_cost_id	Inherited from the template organization
calendar_exception_set_id	Inherited from the template organization
calendar_code	Inherited from the template organization
general_ledger_update_code	Inherited from the template organization
default_atp_rule_id	Inherited from the template organization
default_picking_rule_id	Inherited from the template organization
default_locator_order_value	Inherited from the template organization
default_subinv_order_value	Inherited from the template organization
negative_inv_receipt_code	Inherited from the template organization
stock_locator_control_code	Inherited from the template organization
material_account	Inherited from the template organization
material_overhead_account	Inherited from the template organization
matl_ovhd_absorption_acct	Inherited from the template organization
resource_account	Inherited from the template organization
purchase_price_var_account	Inherited from the template organization
ap_accrual_account	Inherited from the template organization
overhead_account	Inherited from the template organization
outside_processing_account	Inherited from the template organization
intransit_inv_account	Inherited from the template organization
interorg_receivables_account	Inherited from the template organization
interorg_price_var_account	Inherited from the template organization
interorg_payables_account	Inherited from the template organization
cost_of_sales_account	Inherited from the template organization
encumbrance_account	Inherited from the template organization
interorg_transfer_cr_account	Inherited from the template organization
matl_interorg_transfer_code	Inherited from the template organization
interorg_trnsfr_charge_percent	Inherited from the template organization
source_organization_id	Inherited from the template organization
source_subinventory	Inherited from the template organization
source_type	Inherited from the template organization
serial_number_type	Inherited from the template organization
auto_serial_alpha_prefix	Inherited from the template organization
start_auto_serial_number	Inherited from the template organization
auto_lot_alpha_prefix	Inherited from the template organization
lot_number_uniqueness	2 (None)
lot_number_generation	Inherited from the template organization
lot_number_zero_padding	Inherited from the template organization
lot_number_length	Inherited from the template organization
starting_revision	Inherited from the template organization
default_demand_class	Inherited from the template organization
encumbrance_reversal_flag	Inherited from the template organization
maintain_fifo_qty_stack_type	Inherited from the template organization
process_enabled_flag	‘Y’

Warehouse – Subinventory Column Mapping

MTL_SECONDARY_INVENTORIES

The data from ic_whse_mst is migrated to this table for the rows where subinventory_ind = Y.

Column	Value
Secondary_Inventory_Name	Whse_Code
Organization_Id	L_orgn_id
Description	Whse_Name
Disable_Date	NULL
Inventory_ATP_Code	1
Availability_Type	1
reservable_type	1
locator_type	If Org Locator Control Code = 4 Then Assign 5 (Determined at item) Else Assign 1
picking_order	NULL
dropping_order	NULL
material_account	Material_Account from Mtl_Parameters for :Organization_Id
material_overhead_account	Material_Overhead_Account from Mtl_Parameters for :Organization_Id
resource_account	Resource_Account from Mtl_Parameters for :Organization_Id
overhead_account	Overhead_Account from Mtl_Parameters for :Organization_Id
outside_processing_accout	Outside_Processing_Account from Mtl_Parameters for :Organization_Id
quantity_tracked	1
asset_inventory	1
source_type	NULL
source_subinventory	NULL
source_organization_id	NULL
requisition_approval_type	NULL
expense_account	Expense_Account from Mtl_Parameters for :Organization_Id
encumbrance_account	Encumbrance_Account from Mtl_Parameters for :Organization_Id
attribute_category	NULL
attribute1..15	NULL
preprocessing_lead_time	NULL
processing_lead_time	NULL
postprocessing_lead_time	NULL
demand_class	NULL
project_id	NULL
task_id	NULL
subinventory_usage	NULL
notify_list_id	NULL
depreciable_flag	2
location_id	NULL
status_id	1
default_loc_status_id	1
lpn_controlled_flag	0
default_cost_group_id	Default_Cost_Group_Id from Mtl_Parameters for :Organization_Id
pick_uom_code	NULL
cartonization_flag	0
planning_level	2
default_count_type_code	2 (Order Quantity)
subinventory_type	1 (Storage)
enable_bulk_pick	N

MTL_ITEM_LOCATIONS

The data from ic_loct_mst is migrated to this table.

Column	Value
inventory_location_id	Mtl_item_locations_s.nextval
Organization_Id	L_orgn_id
Description	Loct_Name
Descriptive_Text	NULL
Disable_Date	NULL
Picking_Order	
Location_Maximum_Units	NULL
Subinventory_Code	:Whse_Code
Location_Weight_uom_Code	NULL
Max_Weight	NULL
Volume_uom_Code	NULL
Max_Cubic_Area	NULL
Segment1	:Location
Segment2..20	NULL
Summary_Flag	N
Enabled_Flag	Y
Start_Date_Active	:Creation_Date
End_Date_Active	NULL
Attribute_Category	NULL
Attribute1..15	NULL
Project_Id	NULL
Task_Id	NULL
Physical_Location_id	NULL
Pick_Uom_Code	NULL
Dimension_Uom_Code	NULL
Length	NULL
Width	NULL
Height	NULL
Locator_Status	1
Status_Id	NULL
Current_Cubic_Area	NULL
Available_Cubic_Area	NULL
Current_Weight	NULL
Available_Weight	NULL
Location_Current_Units	NULL
Location_Available_Units	NULL
Suggested_Cubic_Area	NULL
Empty_Flag	NULL
Mixed_Items_Flag	NULL
Dropping_Order	NULL
Location_Suggested_Units	NULL
Availability_Type	NULL
Inventory_ATP_Code	1
Reservable_Type	1

Column	Value
Inventory_Item_Id	NULL

Cost Management Table Mapping

Global Column Changes

The following shows data model changes necessary for Co_Code, Item_id, Whse_code, Lot_id, UOM columns. In each case, the new column is added to the table. However, the old columns are not dropped and continue to hold the existing data for historical rows.

Co_code is transformed in each table to the Legal Entity. Refer to Cost Management Migration for details. If the suggested option is chosen, the company can be migrated to the Legal Entity.

Column Name	Datatype	Maximum Length	Description
LEGAL_ENTITY_ID	NUMBER	15	Legal Entity Identifier

Item_Id, Whse_Code are transformed to inventory_item_id and organization_id.

Column Name	Datatype	Maximum Length	Description
INVENTORY_ITEM_ID	NUMBER		Inventory Item Identifier
ORGANIZATION_ID	NUMBER		Inventory Organization

Cost_mthd_code is transformed to cost_type_id.

Column Name	Datatype	Maximum Length	Description
COST_TYPE_ID	NUMBER	(15)	Cost Type Identifier

Calendar_code, period_code are transformed to period_id.

Column Name	Datatype	Maximum Length	Description
PERIOD_ID	NUMBER	(15)	Cost Period Identifier

Lot_ID is transformed to lot_number.

Column Name	Datatype	Maximum Length	Description
LOT_NUMBER	VARCHAR2	(80)	Lot Number

UOM is transformed to the 3 character UOM Code in Discrete.

Column Name	Datatype	Maximum Length	Description
XXX_um or um_YYY or um_code	VARCHAR2	(3)	UOM. Note: Unit of Measure is a separate column that is not added. It contains the full 25 character unit of measure name.

Wherever we store the complete code combination in the form of account_key, it is transformed to account_id.

Column Name	Datatype	Maximum Length	Description
ACCOUNT_ID	NUMBER	(15)	Code Combination Id

Cost Management Table Column Additions

CM_ACPR_CTL

Column Name	Data Type	Size	Column Description
INTORDERS_PROCESSED	NUMBER	15	
INVOICES_PROCESSED	NUMBER	15	
MISCRCPT_PROCESSED	NUMBER	15	

CM_ADJS_DTL

Column Name	Data Type	Size	Column Description
ORGANIZATION_ID	NUMBER	15	Organization Id
INVENTORY_ITEM_ID	NUMBER		Inventory Item Id
COST_TYPE_ID	NUMBER		Cost Type Id
PERIOD_ID	NUMBER	15	Cost Period Id
ADJUST_QTY_UOM	VARCHAR2	3	3 Char UOM Code

CM_BRDN_DTL

Column Name	Data Type	Size	Column Description
BURDEN_UOM	VARCHAR2	3	3 Char UOM Code
ITEM_UOM	VARCHAR2	3	3 Char UOM Code

CM_CLDR_HDR_B

In the calendar header table, the existing references to company and cost method are not valid in Release 12 and are dropped from the calendar setup window.The calendars are shared across multiple Legal Entitys and cost types.

Column Name	Data Type	Size	Column Description
LEGAL_ENTITY_ID	NUMBER	15	Legal Entity Id

CM_RLUP_CTL

Column Name	Data Type	Size	Column Description
FROM_ORGANIZATION_ID	NUMBER		From Organization
TO_ORGANIZATION_ID	NUMBER		To Organization

CM_RSRC_DTL

Column Name	Data Type	Size	Column Description
USAGE_UOM	VARCHAR2	3	3 Char UOM Code

CM_SCST_LED

Column Name	Data Type	Size	Column Description
FORM_PROD_UOM	VARCHAR2	3	3 Char UOM Code
ITEM_FMQTY_UOM	VARCHAR2	3	3 Char UOM Code
USAGE_UOM	VARCHAR2	3	3 Char UOM Code

CM_WHSE_SRC

Column Name	Data Type	Size	Column Description
SOURCE_ORGANIZATION_ID	NUMBER	15	Source Organization Id

GL_ALOC_EXP

Column Name	Data Type	Size	Column Description
FROM_ACCOUNT_ID	NUMBER		From GL Account Id
TO_ACCOUNT_ID	NUMBER		To GL Account Id

GL_ALOC_INP

Column Name	Data Type	Size	Column Description
ACCOUNT_ID	NUMBER	15	Account

GL_EVNT_PLC

Column Name	Data Type	Size	Column Description

Column Name	Data Type	Size	Column Description
ENTITY_CODE	VARCHAR2	30	Event Entity
ENTITY_CLASS_CODE	VARCHAR2	30	Event Class

GL_ITEM_DTL

Column Name	Data Type	Size	Column Description
COST_LEVEL	NUMBER	5	Cost Level Indicator

GMF_BURDEN_PRIORITIES

Column Name	Data Type	Size	Column Description
ORGANIZATION_PRI	NUMBER	5	Organization Priority

GMF_LOT_COST_ADJUSTMENTS

Column Name	Data Type	Size	Column Description
LOT_NUMBER	VARCHAR2	80	Lot Number

GMF_LOT_COST_BURDENS

Column Name	Data Type	Size	Column Description
LOT_NUMBER	VARCHAR2	80	Lot Number
RESOURCE_UOM	VARCHAR2	3	3 Char UOM Code
ITEM_UOM	VARCHAR2	3	3 Char UOM Code

GMF_MATERIAL_LOT_COST_TXNS

Column Name	Data Type	Size	Column Description
COST_TRANS_UM	VARCHAR2	3	3 Char UOM Code
TRANSACTION_ID	NUMBER		Transaction_Id from MMT
MIGRATION_IND	NUMBER	3	Migration or Old row indicator. 1 = old row; null = new row in R12

New Cost Management Tables for Release 12

GMF_FISCAL_POLICIES

A new table, GMF_FISCAL_POLICIES, has been introduced to migrate the fiscal policy records to the new Release 12 data model. Following is the GL_PLCY_MST to GMF_FISCAL_POLICIES data mapping.

Source Column in GL_PLCY_MST	Purpose	Destination Column in GMF_FISCAL_POLICIES
CO_CODE	OPM Company	LEGAL_ENTITY_ID, LEGAL_ENTITY_NAME
NUMBER_OF_OPEN_PERIODS	Obsolete in 11i	
AUTO_LINK_FLAG	Obsolete in 11i	

BASE_CURRENCY_CODE	Display Only. Derived from the Set of Books associated to the Operating Unit.	BASE_CURRENCY_CODE In R12, this column will be derived from the legal entity's primary ledger's currency.
FRONT_CURR_IND	Obsolete in 11i	
VOUCHER_REVERSAL_DAY	Obsolete in 11i	
PRIMARY_LEDGER_CODE	Obsolete in 11i	
UPD_OPEN_BAL	Obsolete in 11i	
SEGMENT_DELIMITER	Will be obsolete in R12.	
TRANS_CNT	Will be obsolete in R12.	
TEXT_CODE	Edit Text	TEXT_CODE
DELETE_MARK	Mark for purge indicator	DELETE_MARK
DFLTS_IND	Obsolete in 11i	
SET_OF_BOOKS_NAME	Set of Books name	LEDGER_NAME Name of the primary ledger in R12.
MTL_CMPNTCLS_ID	Company wide default for Actual Cost material component class for raw materials.	MTL_CMPNTCLS_ID
MTL_ANALYSIS_CODE	Company wide default for Actual Cost material analysis code for raw materials.	MTL_ANALYSIS_CODE
GL_COST_MTHD	Stores the cost method used for inventory valuation to GL.	COST_TYPE
CREV_CURR_MTHD	Cost Revaluation current cost type	Obsolete
CREV_CURR_CALENDAR	Cost Revaluation current cost calendar	Obsolete
CREV_CURR_PERIOD	Cost Revaluation current cost period	Obsolete
CREV_PRIOR_MTHD	Cost Revaluation previous cost type	Obsolete
CREV_PRIOR_CALENDAR	Cost Revaluation previous cost calendar	Obsolete
CREV_PRIOR_PERIOD	Cost Revaluation previous cost period	Obsolete
GL_TRANS_DATE	Cost Revaluation GL date for journals	Obsolete
COST_BASIS	Indicates if the cost to be used should come from current period or previous period. No indication that	Obsolete

	customers are using this at this time.	
SOB_ID	Set of Books identifier	LEDGER_ID Id of the primary ledger in R12
OPERATING_UNIT	Obsolete	
ORG_ID	Obsolete	
CREATED_BY	WHO column	CREATED_BY
CREATION_DATE	WHO column	CREATION_DATE
LAST_UPDATE_LOGIN	WHO column	LAST_UPDATE_LOGIN
LAST_UPDATE_DATE	WHO column	LAST_UPDATE_DATE
LAST_UPDATED_BY	WHO column	LAST_UPDATED_BY
ATTRIBUTE1 thru ATTRIBUTE30	Descriptive Flexfield	ATTRIBUTE1 thru ATTRIBUTE30
ATTRIBUTE_CATEGORY	Descriptive Flexfield	ATTRIBUTE_CATEGORY

GMF_CALENDAR_ASSIGNMENTS

This is a new table that stores the association between a calendar and the Legal Entity/cost type combination. A calendar can be associated to any number of Legal Entity/cost type combination as long as for the same combinations no two calendar periods are defined for the same time period.

Column Name	Data Type	Size	Column Description
ASSIGNMENT_ID	NUMBER	15	Assignmnt Id
CALENDAR_CODE	VARCHAR2	4	Calendar Code
LEGAL_ENTITY_ID	NUMBER	15	Legal Entity Id
COST_TYPE_ID	NUMBER		Cost Type Id
CREATION_DATE	DATE		Standard Columns
CREATED_BY	NUMBER	15	Standard Columns
LAST_UPDATE_DATE	DATE		Standard Columns
LAST_UPDATED_BY	NUMBER	15	Standard Columns
LAST_UPDATE_LOGIN	NUMBER	15	Standard Columns
TEXT_CODE	NUMBER	10	Text Code
DELETE_MARK	NUMBER	1	Delete Mark Indicator
ATTRIBUTE1	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE2	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE3	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE4	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE5	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE6	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE7	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE8	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE9	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE10	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE11	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE12	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE13	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE14	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE15	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE16	VARCHAR2	240	Standard Attribute Column

Column Name	Data Type	Size	Column Description
ATTRIBUTE17	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE18	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE19	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE20	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE21	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE22	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE23	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE24	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE25	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE26	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE27	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE28	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE29	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE30	VARCHAR2	240	Standard Attribute Column

GMF_PERIOD_STATUSES

This table stores the period statuses for the Legal Entity/cost type combination. For the calendars associated or assigned to the Legal Entity/cost type this table stores the statuses of the various periods. The table is de-normalized to store the calendar code, period code, and start/end dates so that all code can refer to this table to derive the necessary information.

Column Name	Data Type	Size	Column Description
PERIOD_ID	NUMBER	15	Period Id
LEGAL_ENTITY_ID	NUMBER	15	Legal Entity Id
COST_TYPE_ID	NUMBER		Cost Type Id
CALENDAR_CODE	VARCHAR2	4	Calendar Code
PERIOD_CODE	VARCHAR2	4	Period Code
PERIOD_ID	NUMBER	15	Period Unique Identifier
PERIOD_STATUS	VARCHAR2	1	Period Status
START_DATE	DATE		Period Start Date
END_DATE	DATE		Period End Date
CREATION_DATE	DATE		Standard Columns
CREATED_BY	NUMBER	15	Standard Columns
LAST_UPDATE_DATE	DATE		Standard Columns
LAST_UPDATED_BY	NUMBER	15	Standard Columns
LAST_UPDATE_LOGIN	NUMBER	15	Standard Columns
TEXT_CODE	NUMBER	10	Text Code
DELETE_MARK	NUMBER	1	Delete Mark Indicator
ATTRIBUTE1	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE2	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE3	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE4	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE5	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE6	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE7	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE8	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE9	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE10	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE11	VARCHAR2	240	Standard Attribute Column

Column Name	Data Type	Size	Column Description
ATTRIBUTE12	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE13	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE14	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE15	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE16	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE17	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE18	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE19	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE20	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE21	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE22	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE23	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE24	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE25	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE26	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE27	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE28	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE29	VARCHAR2	240	Standard Attribute Column
ATTRIBUTE30	VARCHAR2	240	Standard Attribute Column

Lot Status Migration Recommendations

In the R12.0 common Inventory product, the Material Status model replaces Lot Status that has been available in OPM Inventory. This model has more flexibility than the Lot Status Model because it can be defined at three levels within Oracle Inventory. A Subinventory, Locator, and Lot can all be assigned a Status. The default Status is called “Active” and all new entities are flagged with this Status. This status does not restrict any transactions and specifies that material is fully reservable, nettable, and ATP-able. Other Statuses that restrict transactions can be assigned to different levels. This means that material may effectively be subject to multiple Statuses – the Status of the Subinventory in which it is stored, the Status of the Locator in which it is stored, and the Status of the Lot of which it is a part. In this case, the restrictions are summed in determining material eligibility. In other words, any transaction prohibited by any applicable Status will be prohibited for that material.

As a result of this new modeling of status, there is no direct mapping feasible when there are multiple material statuses in one location. In the common inventory model, a subinventory and /or locator can only have one status. Therefore, there are several scenarios that may occur during migration from 11i to R12. The following discusses these cases and provides solutions for migration.

Default Lot Migration Behavior

If a lot / sublot exists in multiple locations with a different status, the system will migrate the largest quantity balance with the original Lot number as the Parent Lot and the Lot updates with the lot /sublot number concatenated. An example of this migration is found below for Item I1; Lot L1, Sublot S1 and Warehouse W1 for the location STR with a “Good” status. Because the same Lot / Sublot combination also exists in RCV with a different status, the migration will create a new lot L1-S1-INSP which adds the Status to the Lot/Sublot number.

OPM							Discrete							
Item	**Lot**	**Sublot**	**Warehouse**	**Location**	**Status**	**Bal**	**Org**	**Item**	**Parent Lot**	***Lot***	**Lot Status**	**Locator**	**Locator Status**	**Bal**
I1	**L1**	**S1**	**W1**	**RCV**	**INSP**	100	W1	I1	L1	***L1-S1-INSP***	INSP	RCV	Active	100
				STR	**GOOD**	500				*L1-S1*	GOOD	STR	Active	500
	L2	**S1**	**W1**	**SHP**	**SHIP**	200			L2	***L2-S1-SHIP***	SHIP	SHP	Active	200
				STR	**GOOD**	1000				*L2-S1*	GOOD	STR	Active	1000
				PRD	**PROD**	300				***L2-S1-PROD***	PROD	PRD	Active	300
	L2	**S1**	**W1**	**PRD**	**PROD**	600			L2	*L2-S1*	PROD	PRD	Active	600
				SHP	**GOOD**	50				***L2-S1-GOOD***	GOOD	SHP	Active	50

This default migration model may not suit all business practices, therefore the following scenarios provide some additional solutions:

Case 1: A single lot status is associated with lot number and all locations have the same status.

In this case, there is a single lot status within a warehouse and the lot can be stored in multiple locations that are all “active”.   The system will migrate the largest quantity balance with the original Lot number as the Parent Lot and the Lot updates with the lot /sublot number concatenated. The Locators will all have a status of active. As a result, the material status will be determined by the Lot level status.

OPM							Discrete							
Item	**Lot**	**Sublot**	**Warehouse**	**Location**	**Status**	**Bal**	**Org**	**Item**	**Parent Lot**	**Lot**	**Lot Status**	**Locator**	**Locator Status**	**Bal**
I1	L1	S1	W1	RCV	INSP	100	W1	I1	L1	L1-S1	INSP	RCV	Active	100
				STR	INSP	500						STR	Active	500
	L2	S1	W1	SHP	SHIP	200			L2	L2-S1	SHIP	SHP	Active	200
				STR	SHIP	1000						STR	Active	1000
				PRD	SHIP	300						PRD	Active	300
	L3	S1	W1	STR	GOOD	550			L3	L3-S1	GOOD	STR	Active	550
	L4	S1	W1	B1	BAD	150			L4	L4-S1	BAD	B1	Active	150
I2	L1	S1	W1	RCV	GOOD	700		I2	L1	L1-S1	GOOD	RCV	Active	700
	L2	S1	W1	PRD	PROD	600			L2	L2-S1	PROD	PRD	Active	600
				SHP	PROD	50						SHP	Active	50

Case 2: Lot status varies by location.

In this scenario, each location has a different status to segregate materials by the approved use. We will use the following table in the example:

Warehouse	Location	Discrete Locator Status
W1	RCV	INSP
	STR	GOOD
	SHP	SHIP
	PRD	PROD
	B1	BAD
	Q1	REJ

Run a validation report to identify the items with the same lot / sublot in locations with different statuses (highlighted below in Bold Italics). Next manually change the conflicting status to a status of your choice (such as “migrate”). The system will migrate the Lots and Sublots as previously described in case 1. The Status will be migrated to the Lot Status, and all locators will have an “active” status as displayed in the following table:

OPM								**Discrete**							
Item	**Lot**	**Sublot**	**Warehouse**	**Location**	**Status**	**Bal**	**Change Status**	**Org**	**Item**	**Parent Lot**	**Lot**	**Lot Status**	**Locator**	**Locator Status**	**Bal**
I1	L1	S1	W1	RCV	INSP	100		W1	I1	L1	L1-S1	INSP	RCV	Active	100
				STR	INSP	500							STR	Active	500
	L2	***S1***	***W1***	***SHP***	***SHIP***	200	Migrate			L2	L2-S1	Migrate	SHP	Active	200
				STR	***SHIP***	1000	Migrate						STR	Active	1000
				PRD	***PROD***	300	Migrate						PRD	Active	300
	L3	S1	W1	STR	GOOD	550				L3	L3-S1	GOOD	STR	Active	550
	L4	S1	W1	B1	BAD	150				L4	L4-S1	BAD	B1	Active	150
I2	L1	S1	W1	RCV	GOOD	700			I2	L1	L1-S1	GOOD	RCV	Active	700
	L2	S1	W1	PRD	PROD	600				L2	L2-S1	PROD	PRD	Active	600
				SHP	PROD	50							SHP	Active	50

Because you want to model the status at the locator level in common Oracle Inventory, you will need to manually change the Locator Statuses to the appropriate new status and manually change all of the Lot Statuses to “active”. This will result in the following transactions and data:

Action	**Org**	**Item**	**Parent Lot**	**Lot**	**Lot Status**	**Locator**	**Locator Status**	**Bal**
	W1	I1	L1	L1-S1	Active	RCV	INSP	100
Move 500 from STR to RCV locator						***STR***	***GOOD***	500
			L2	L2-S1	Active	SHP	SHIP	200
Move 1000 from STR to SHIP						***STR***	***GOOD***	1000
						PRD	PROD	300
			L3	L3-S1	Active	STR	GOOD	550
			L4	L4-S1	Active	B1	BAD	150
Move 700 from RCV to STR		I2	L1	L1-S1	Active	***RCV***	***INSP***	700
			L2	L2-S1	Active	PRD	PROD	600
Move 50 from SHIP to PRD						***SHP***	***SHIP***	50

Note: Statuses shown in *Red Bold Italics* indicate a status different from the original OPM status.

Case 3: Lots can be stored in multiple locators and can have different statuses. The locator can contain lots with different statuses.

In this case, the lots you have may have a different status based on the location they are store in and a given location co-mingles items with different statuses. As in Case 2, first run validation report which will identify the items with the same lot / sublot in locations with different statuses (highlighted below in Bold Italics). Next manually change the conflicting status to a status of your choose (such as “migrate”). The system will migrate the Lots and Sublots as previously described in case 1. The Status will be migrated to the Lot Status and all locators will have an “active” status as displayed in the table below:

OPM								**Discrete**							
Item	**Lot**	**Sublot**	**Warehouse**	**Location**	**Status**	**Bal**	**Change Status**	**Org**	**Item**	**Parent Lot**	**Lot**	**Lot Status**	**Locator**	**Locator Status**	**Bal**
I1	***L1***	***S1***	***W1***	***RCV***	***INSP***	100	Migrate	W1	I1	L1	L1-S1	Migrate	RCV	Active	100
				STR	***GOOD***	500	Migrate						STR	Active	500
	L2	***S1***	***W1***	***SHP***	***SHIP***	200	Migrate			L2	L2-S1	Migrate	SHP	Active	200
				STR	***SHIP***	1000	Migrate						STR	Active	1000
				PRD	***PROD***	300	Migrate						PRD	Active	300
	L3	S1	W1	STR	INSP	550				L3	L3-S1	INSP	STR	Active	550
	L4	S1	W1	B1	BAD	150				L4	L4-S1	BAD	B1	Active	150
I2	L1	S1	W1	RCV	GOOD	700			I2	L1	L1-S1	GOOD	RCV	Active	700
	L2	***S1***	***W1***	***PRD***	***PROD***	600	Migrate			L2	L2-S1	Migrate	PRD	Active	600
				SHP	***GOOD***	50	Migrate						SHP	Active	50

Then create new “logical” locators. Create one for each status – locator combination that could occur. For example:

Locator	**Locator Status**
RCV-INSP	INSP
RCV-GOOD	GOOD
STR-GOOD	GOOD
STR-INSP	INSP
SHP-SHIP	SHIP
SHP-GOOD	GOOD
PRD-PROD	PROD
B1-BAD	BAD

Manually change all Lot Statuses to Active and move the inventory to the appropriate locators: This would result in the following transaction and data:

		Parent		**Lot**		**Locator**	
Org	**Item**	**Lot**	**Lot**	**Status**	**Locator**	**Status**	**Bal**
W1	I1	L1	L1-S1	Active	RCV-INSP	INSP	100
					STR-GOOD	GOOD	500
		L2	L2-S1	Active	SHP-SHIP	SHIP	200
					STR-GOOD	GOOD	1000
					PRD-PROD	PROD	300
		L3	L3-S1	Active	STR-INSP	INSP	550
		L4	L4-S1	Active	B1-BAD	BAD	150
	I2	L1	L1-S1	Active	RCV-GOOD	GOOD	700
		L2	L2-S1	Active	PRD-PROD	PROD	600
					SHP-GOOD	GOOD	50