

**Oracle® Fusion Applications Order Fulfillment, Order
Orchestration Guide**

11g Release 1 (11.1.3)

Part Number E22707-03

December 2011

Oracle® Fusion Applications Order Fulfillment, Order Orchestration Guide

Part Number E22707-03

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

Authors: Cynthia Stevens, Deborah West

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

1 Manage Open Sales Orders

Manage Open Sales Orders: Overview	1-1
Oracle Fusion Distributed Order Orchestration Business Objects: How They Fit Together	1-1
Editing Fulfillment Line Attributes: Examples	1-3
Error Recovery: Explained	1-4
Sourcing Items from a Different Warehouse: Points to Consider	1-5
Resolving Fulfillment Line Exceptions: Points to Consider	1-5
Holds: Explained	1-7
Reviewing Exceptions: Examples	1-8
Manual Scheduling: Explained	1-9
Order Orchestration Work Area Analytics: Explained	1-10
Split Fulfillment Lines: Explained	1-12
Disabled Fulfillment Line Actions: Explained	1-13
Models: Explained	1-14
FAQs for Manage Open Sales Orders	1-15

2 Process Sales Orders

Process Sales Orders: Overview	2-1
Changes: How They Are Processed	2-1
Task Layer: Explained	2-2
Shipment Task Layer Service: Explained	2-3
Schedule Task Layer Service: Explained	2-4
Reservation Task Layer Service: Explained	2-4
Invoice Task Layer Service: Explained	2-4
Holds Task Layer Service: Explained	2-5
Activity Task Layer Service: Explained	2-6
Return Task Layer Service: Explained	2-7
Template Task Layer: Explained	2-8
Fulfillment Order Task Layer Explained	2-9
Sales Order Transformation to Orchestration Order: Explained	2-10
Activities: Explained	2-10
Orchestration Process Assignment Failure: Explained	2-10

3 Define Sales Order Fulfillment

Oracle Fusion Distributed Order Orchestration Components: How They Work Together	3-1
Orchestration Lookups: Explained	3-2
Orchestration Profile Management: Points to Consider	3-3

Oracle Fusion Distributed Order Orchestration Extensible Flexfields: Explained	3-4
Oracle Fusion Distributed Order Orchestration Extensible Flexfield Uses: Explained	3-4
Oracle Fusion Distributed Order Orchestration Extensible Flexfield Setup: Explained	3-6
.....	3-6
Manage Hold Codes	3-7
Manage Orchestration Source Systems	3-8
Collect Orchestration Reference and Transaction Data	3-11
Define Orchestration	3-40
Define Processing Constraints	3-74
Define Transformation Details	3-77

Preface

This Preface introduces the guides, online help, and other information sources available to help you more effectively use Oracle Fusion Applications.

Oracle Fusion Applications Help

You can access Oracle Fusion Applications Help for the current page, section, activity, or task by clicking the help icon. The following figure depicts the help icon.



With a local installation of help, you can add custom help files to replace or supplement the provided content. Help content patches are regularly made available to ensure you have access to the latest information. Patching does not affect your custom content.

Oracle Fusion Applications Guides

Oracle Fusion Applications guides are a structured collection of the help topics, examples, and FAQs from the help system packaged for easy download and offline reference, and sequenced to facilitate learning. You can access the guides from the **Guides** menu in the global area at the top of Oracle Fusion Applications Help pages.

Note

The **Guides** menu also provides access to the business process models on which Oracle Fusion Applications is based.

Guides are designed for specific audiences:

- **User Guides** address the tasks in one or more business processes. They are intended for users who perform these tasks, and managers looking for an overview of the business processes. They are organized by the business process activities and tasks.
- **Implementation Guides** address the tasks required to set up an offering, or selected features of an offering. They are intended for implementors. They are organized to follow the task list sequence of the offerings, as displayed within the Setup and Maintenance work area provided by Oracle Fusion Functional Setup Manager.
- **Concept Guides** explain the key concepts and decisions for a specific area of functionality. They are intended for decision makers, such as chief financial officers, financial analysts, and implementation consultants. They are organized by the logical flow of features and functions.

- **Security Reference Manuals** describe the predefined data that is included in the security reference implementation for one offering. They are intended for implementors, security administrators, and auditors. They are organized by role.

To supplement these guides, which cover specific business processes and offerings, the following guides address common areas:

Guide	Intended Audience	Purpose
Common User Guide	All users	Explains tasks performed by most users.
Common Implementation Guide	Implementors	Explains tasks within the Define Common Applications Configuration task list, which is included in all offerings.
Information Technology Management, Implement Applications Guide	Implementors	Explains how to use Oracle Fusion Functional Setup Manager to plan, manage, and track your implementation projects, migrate setup data, and validate implementations.
Technical Guides	System administrators, application developers, and technical members of implementation teams	Explain how to install, patch, administer, and customize Oracle Fusion Applications.

For guides that are not available from the Guides menu, go to Oracle Technology Network at <http://www.oracle.com/technetwork/indexes/documentation>.

Other Information Sources

My Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Use the My Oracle Support Knowledge Browser to find documents for a product area. You can search for release-specific information, such as patches, alerts, white papers, and troubleshooting tips. Other services include health checks, guided lifecycle advice, and direct contact with industry experts through the My Oracle Support Community.

Oracle Enterprise Repository for Oracle Fusion Applications

Oracle Enterprise Repository for Oracle Fusion Applications provides visibility into service-oriented architecture assets to help you manage the lifecycle of your software from planning through implementation, testing, production,

and changes. In Oracle Fusion Applications, you can use the Oracle Enterprise Repository for Oracle Fusion Applications for:

- Technical information about integrating with other applications, including services, operations, composites, events, and integration tables. The classification scheme shows the scenarios in which you use the assets, and includes diagrams, schematics, and links to other technical documentation.
- Publishing other technical information such as reusable components, policies, architecture diagrams, and topology diagrams.

The Oracle Fusion Applications information is provided as a solution pack that you can upload to your own deployment of Oracle Enterprise Repository for Oracle Fusion Applications. You can document and govern integration interface assets provided by Oracle with other assets in your environment in a common repository.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/us/corporate/accessibility/index.html>.

Comments and Suggestions

Your comments are important to us. We encourage you to send us feedback about Oracle Fusion Applications Help and guides. Please send your suggestions to oracle_fusion_applications_help_ww@oracle.com. You can use the **Send Feedback to Oracle** link in the footer of Oracle Fusion Applications Help.

Manage Open Sales Orders

Manage Open Sales Orders: Overview

The Manage Open Sales Orders business process is performed by customer service representatives, order managers, and order administrators to ensure the efficient fulfillment of sales orders. Actions are performed in the Order Orchestration work area. This work area provides information about orders from as many disparate order capture systems and fulfillment systems as your organization uses. In the Order Orchestration work area, you can:

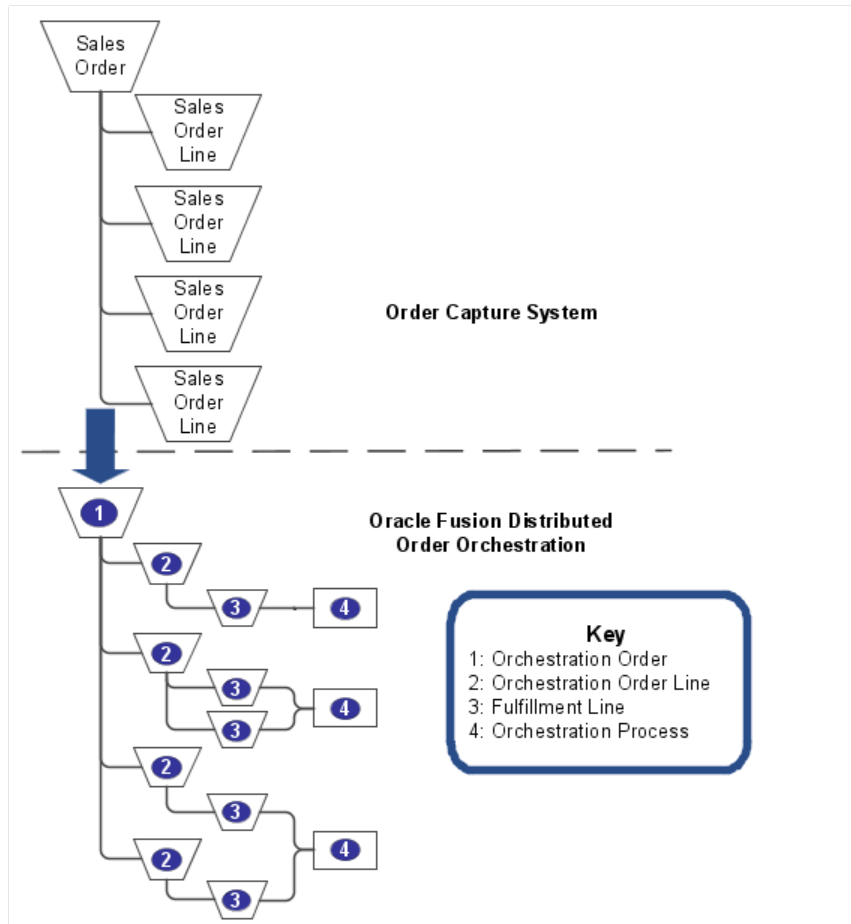
- Monitor order statuses and exceptions.
 - View orchestration orders and orchestration processes.
 - Identify fulfillment lines and orchestration processes with exceptions.
 - View orchestration process status using a Gantt chart and visual alerts for tasks in jeopardy.
 - Check item availability, including a variety of options and their costs and delivery dates.
 - Use embedded intelligence for decision making.
- Take corrective actions, such as substitute an item or split lines, so that supply can be sourced from multiple warehouses.
- Schedule fulfillment lines manually.
- Manage holds.
- Manage order orchestration messages.
- Recover from errors.

Oracle Fusion Distributed Order Orchestration Business Objects: How They Fit Together

Oracle Fusion Distributed Order Orchestration business objects work together to enable you to monitor, manage, and fulfill orders efficiently. This topic discusses the business objects and their relationships.

This figure shows what happens to the structure of a sales order when it enters Distributed Order Orchestration. In this example, a sales order with four sales order lines enters Distributed Order Orchestration from the order capture

system. After decomposition, the sales order becomes an orchestration order with four orchestration order lines. The first orchestration order line has one fulfillment line, which is associated with one orchestration process. The second orchestration order line has two fulfillment lines, both of which are associated with another orchestration process. The third and fourth orchestration order lines each have one fulfillment line, both of which are associated with a third orchestration process.



Orchestration Orders

When Oracle Fusion Distributed Order Orchestration receives a source sales order, it decomposes the order into logical pieces that can be fulfilled. The primary component is an orchestration order, which comprises orchestration order lines and fulfillment lines. During decomposition, the sales order may be broken down into a different structure, so the resulting orchestration order might not exactly mirror the original sales order.

An orchestration order contains orchestration order lines. This relationship is like the relationship between a sales order and a sales order line.

When you want to act on an orchestration order, you do so by taking action on the fulfillment lines that are part of the orchestration order.

Orchestration Order Lines

An orchestration order comprises one or more orchestration order lines. An orchestration order line usually has one fulfillment line. When a fulfillment line

is split, two or more fulfillment lines map back to the same orchestration order line.

The relationship between orchestration order and orchestration order lines preserves some of the original structure of the sales order, making it easier to keep track of the original intent of the order, even when fulfillment lines associated with orchestration order lines are split.

Fulfillment Lines

A fulfillment line represents a request for product to be fulfilled. It contains important information about the request, such as customer, order, item, quantity, scheduled ship date, actual ship date, and shipping method.

A fulfillment line is the main business object that you act on in the Order Orchestration work area. You can schedule or unschedule a fulfillment line; reserve product for the line; substitute an item on the line; change the warehouse, shipping method, or demand class; and other such actions.

A fulfillment line is assigned to an orchestration process, which carries out the necessary set of tasks to fulfill the line.

Most orders are new sales orders, but some are for product returns. Fulfillment lines represent new orders. Return fulfillment lines represent orders for product returns.

Orchestration Processes

An orchestration process is a defined set of tasks, their sequence, and dependencies required to fulfill one or more fulfillment lines. Orchestration processes include user tasks, such as activities, and automated tasks, such as scheduling, reserving, shipping, returning, and invoicing.

Orchestration processes are defined during setup, at which time an order administrator determines the tasks needed to complete an orchestration process. In the Order Orchestration work area, you cannot change the orchestration process definition, but you can see the progression of a fulfillment line through the orchestration process. This visibility enables you to see where in the orchestration process a fulfillment line is.

Source Orders

Source order refers to the sales order from the order capture system. You cannot change a source order in Distributed Order Orchestration.

Return Fulfillment Lines

Return fulfillment lines represent orders for product returns. A return fulfillment line is functionally similar to a fulfillment line, but it has a different set of attributes that are related to returns.

Editing Fulfillment Line Attributes: Examples

One way to manage orders is to edit fulfillment line attributes in the Order Orchestration work area. The following scenarios illustrate when changing fulfillment line attributes is advantageous in facilitating order fulfillment.

Editing the Warehouse for Multiple Fulfillment Lines

You are an order manager at a company that supplies carpeting from various warehouses throughout the United States. You just found out that the Boston warehouse is flooded due to a recent storm, but you know that the Dallas

warehouse has ample supply. You search for all fulfillment lines that are sourcing the carpet from the Boston warehouse. You select all the fulfillment lines and open the Edit Fulfillment Lines window. You change the warehouse to Dallas.

Editing the Demand Class for Multiple Fulfillment Lines

You are an order manager at a company that has been trying to renew the approved supplier status with Corporation ABC. The negotiators at this corporation have stated that they are considering not renewing the approved supplier status due to a number of late orders recently. Because of the importance of this contract, you want to ensure that the corporation's orders are not late. You search for all fulfillment lines for Corporation ABC. You select all the fulfillment lines and open the Edit Fulfillment Lines window. In the Demand Class list of values you select High Priority.

Error Recovery: Explained

Order processing is automated. In most cases, orders are processed to completion without errors. However, some errors may occur. You can identify errors in the various Order Orchestration work area pages. The Manage Order Orchestration Messages page groups messages about a fulfillment line in one location. If you have the error recovery manager role, then you can correct the errors.

The following kinds of errors may occur during order processing:

- Fulfillment lines in an order are not assigned to an orchestration process.
- Orchestration process was not started, or an orchestration process or task resulted in an error.
- Error occurs during internal processing of an orchestration process step.
- Error occurs when sending a task request to a fulfillment system.

Use the following actions in the Order Orchestration work area to correct the errors:

- **Assign Lines to Process:** Use this action when an orchestration process was not assigned to the fulfillment lines of an order. This action attempts to assign fulfillment lines to an orchestration process.
- **Recover Order:** Use this action to correct errors, such as an orchestration process that has not started or an orchestration process or task that resulted in error. The root cause of the error must be corrected first. The action is available only when an error is associated with an orchestration order. You can select one or more orders simultaneously for recovery. The action attempts to recover any orchestration processes or tasks that are in an error state for the selected orders.
- **Recover Process:** Use this action to correct errors, such as an error in internal processing of an orchestration process step or an error in sending a task request to a fulfillment system. You can select one or more orchestration processes for recovery simultaneously. The action attempts to recover any internal errors or tasks that are in an error state for the selected processes.
- **Recover Task:** Use this action to correct errors that result from an attempt to send a request to a fulfillment system. The action is available only when an outbound request fails. If you want to simultaneously select

multiple tasks for recovery, then select multiple orchestration processes or orchestration orders and use Recover Process or Recover Order, respectively.

Some errors must be corrected outside of Oracle Fusion Distributed Order Orchestration, for example:

- Errors in a source order that fail in decomposition must be corrected by resubmitting the order.
- Errors that occur during processing of a response from a fulfillment system must be corrected by resubmitting the response.
- Errors that are identified as recoverable in Oracle Enterprise Manager can be recovered there only.

Sourcing Items from a Different Warehouse: Points to Consider

In the Order Orchestration work area, you can expedite the fulfillment of orders in several ways. One way is to source items that are not available in one warehouse from a different warehouse.

Editing the Warehouse Attribute

If you know that another warehouse has the item and quantity that the fulfillment line requires, then you can change the warehouse manually in the Edit Fulfillment Lines window.

You must find out which warehouse has the item, by consulting the analytics, the Check Availability page, the Supply Availability report, or the Supply Allocations report; or outside the application.

Splitting the Fulfillment Line

If the currently selected warehouse contains only some of the quantity that the fulfillment line requires, then you can manually split the fulfillment line and select another warehouse to source the remainder of the quantity. For example, fulfillment line 1 requires 30 widgets. The fulfillment line is in jeopardy because the currently selected warehouse, Warehouse A, has only 20 widgets. You split the fulfillment line and designate that the second fulfillment line source the remaining 10 widgets from Warehouse B.

Selecting an Option on the Check Availability Page

Order promising provides a variety of options for scheduling. The options may include sourcing the item from a different warehouse. To focus on options for sourcing from a different warehouse, clear the Warehouse attribute on the Check Availability page. Order promising recommends a warehouse based on the defined sourcing rules and warehouse rankings. Consult the Check Availability page for the options.

Resolving Fulfillment Line Exceptions: Points to Consider

Many orders are fulfilled automatically, but at times issues arise during the fulfillment process. Use the Order Orchestration work area to view the reasons and fix the exceptions.

The Manage Fulfillment Line Exceptions page categorizes exceptions as: Past Due, In Jeopardy, Backordered, or On Hold. Past due indicates that the current

date is later than the scheduled ship date. Most exceptions are related to item availability or an orchestration process task. You might have more than one way to resolve an exception; how you resolve exceptions depends on your organization's practices and priorities.

Resolving Item Availability Exceptions

Most fulfillment line exceptions are related to item availability. Consider the various options for resolving exceptions.

Option	What Does It Do?	Why Use It?
Release holds	Releases a hold on a fulfillment line. You can release a hold that was applied in Oracle Fusion Distributed Order Orchestration only.	If a fulfillment line has a hold on some part of it, then it may be unable to proceed through the fulfillment process. A hold that is specific to some part of the orchestration process can continue to proceed but may hit a point at which it will stop and wait for a user to release it.
Reschedule using Check Availability	When you select Check Availability, a variety of scheduling options is presented to you with the best option clearly marked. The best option is determined by a set of business rules that the order administrator defines during implementation.	This is the most effective way to reschedule a fulfillment line because it optimizes choices and provides you with all the information that you need to evaluate trade-offs. You do not have information about supply other than what appears on the Order Orchestration work area.
Override the schedule	Overrides the scheduled ship date or scheduled arrival date assigned by the order promising process.	You are aware of an alternative to the schedule assigned by the order promising process, and you know the best date to schedule the fulfillment line. You can override the schedule of multiple fulfillment lines.
Change the warehouse	Lets you change the warehouse that sources the item on the selected fulfillment lines.	You know the warehouse from which you want to source the item. The analytics provide information that may help you determine the warehouse to choose.
Change the demand class	Determines priority when scheduling the fulfillment line.	You want to schedule fulfillment lines as a group and ensure that important customers get the supply first. You want to expedite fulfillment, but the exact date is not important to you. You can change the demand class of multiple fulfillment lines.

Change the shipping method	Lets you change the shipping method for all the selected fulfillment lines to a more expeditious method.	You know the shipping method that you want to use. You can change the shipping method of multiple fulfillment lines.
Clear selected fulfillment line attributes	Gives the order promising process the freedom to choose the best fulfillment option, without regard to a specific attribute.	You want to use Check Availability to search for the best possible fulfillment options, without having to consider some fulfillment requirements. You can clear the attributes of multiple fulfillment lines.
Substitute the item on the fulfillment line	Substitutes the item on the fulfillment line with an item that you choose. Distributed Order Orchestration presents the best substitution based on rules in the item master.	You see that an allowable substitute will provide you with a better scheduling option. The analytics provide information that may help you determine the item to substitute.
Split the fulfillment line	Splits the selected fulfillment line into two or more fulfillment lines.	You know that you can fulfill part of the fulfillment line from one warehouse and part from another warehouse. The analytics provide information that may help you determine the warehouse to source from.
Unschedule	Releases supply so that it is available for another fulfillment line. This action leaves the line in a state where it is awaiting scheduling.	You want to release supply to make it available for a high-priority fulfillment line.
Schedule	Makes an initial request to the order promising process to schedule a fulfillment line.	The fulfillment line is associated with an orchestration process that has a manual scheduling step.
Unreserve	Releases supply that is earmarked for a specific fulfillment line, so that it is available for another fulfillment line.	You want to release supply reserved for a specific line to make it available for a high-priority fulfillment line.
Reserve	Removes supply from availability and earmarks it for a specific fulfillment line.	You want to ensure that supply is allocated for a particular fulfillment line.

Holds: Explained

Holds prevent an orchestration process from proceeding. You can apply holds to current or future tasks. Holds can come from Oracle Fusion Distributed Order Orchestration or from order capture systems.

Manual Holds

You can apply holds to orchestration orders, orchestration order lines, and fulfillment lines in the Order Orchestration work area.

Automatic Holds

Holds occur automatically when a change from the Order Orchestration work area or the order capture application requires change management to compensate the orchestration process. In this case, a message to apply a hold is sent to the fulfillment system, so that the orchestration process stops while compensation occurs.

Hold Inheritance

A hold on an orchestration order carries over to the fulfillment lines that map to the orchestration order. If a fulfillment line splits, then the hold is mapped to the new fulfillment line. If a hold is applied to one or more lines in a shipment set, then the hold applies to the entire shipment set.

Releasing Holds

Distributed Order Orchestration users can release holds that were applied by a Distributed Order Orchestration user only. The Hold Source System attribute indicates where the hold was applied. If a hold was placed on an orchestration order, then you cannot release the hold on the corresponding lines independently. You must release the hold on the orchestration order. While compensation for a change is occurring, holds are released automatically after all the steps are undone. External systems, such as an order capture application, also can release a hold automatically by sending a message to release a hold that was applied in the external system.

Active Holds

An active hold is a current stop to an orchestration process. A fulfillment line could have a hold on the associated orchestration process, but it continues to be processed because it has not yet reached the step that has the hold. Although a fulfillment line may not have an active hold, it still appears as on hold in the Order Orchestration work area if the line has an unreleased hold.

Reviewing Exceptions: Examples

Errors can come from a variety of modules in Oracle Fusion Distributed Order Orchestration or from an integrated system, such as a billing system. Distributed Order Orchestration captures and reports the order orchestration exceptions, so that you can view exception details and history and address the issues properly and promptly. You can view the exceptions on the Manage Order Orchestration Messages window in the Order Orchestration work area.

Use these scenarios to understand how to use orchestration messages to resolve issues.

Warning from the Shipment System

You are an order manager at a company that manufactures self-cleaning toilets. You see an orchestration order in Distributed Order Orchestration for 100 toilets. The associated fulfillment line was scheduled and reservation was confirmed. The orchestration process sent a shipment request to the shipping system.

The customer asks the sales representative to hold the shipment, due to a fire in its warehouse. The sales representative applies a hold on the order and sends the hold request to Distributed Order Orchestration. The hold cannot be honored by the shipping system because the order already was picked and released. A warning is sent from the shipping system to Distributed Order Orchestration, and the hold request is rejected. The sales representative phones you to discuss the rejected hold request. You see the details in the Manage Order Orchestration Messages window and explain the reason for the hold rejection.

Rejected Change Order

You are an order manager at a company that manufactures grade A pipes. A customer orders 100 pipes, and an orchestration order is created in Distributed Order Orchestration. A few days after Distributed Order Orchestration sent the shipment request to the shipping system, the customer calls to change the order quantity. The sales representative modifies the sales order and sends the change order to Distributed Order Orchestration. Because the order already was shipped, the change order is rejected. The error details are saved. You view the details in the Manage Order Orchestration Messages window and explain to the sales representative the reason for the rejection.

Missing Process Assignment Rules

You are an order manager at a company that provides digital network service. A sales order was received by Distributed Order Orchestration, but an orchestration process was not assigned to the corresponding orchestration order, due to missing process selection rules. You see the exception in the Manage Order Orchestration Messages window. You see that multiple orders are affected by the same error. You notify an order administrator, who adds the necessary process selection rules. Afterwards, the order administrator, using the Error Recovery privilege, activates the assign and launch process. As a result, orchestration processes are assigned to the orchestration orders.

Manual Scheduling: Explained

In most cases, scheduling of a fulfillment line occurs automatically when an orchestration process is assigned to the fulfillment line after a sales order enters Oracle Fusion Distributed Order Orchestration. If a fulfillment line is on a manual scheduling step, then you can schedule the line manually in the Order Orchestration work area.

The Order Orchestration work area provides options for manually scheduling one or more fulfillment lines at a time:

- Select one or more fulfillment lines at the same time and schedule them from a table of fulfillment lines. The fulfillment lines are scheduled in order of the Requested Ship Date or Requested Arrival Date attribute.
- Select one or more fulfillment lines and schedule them by selecting a scheduling option on the Check Availability page. Order promising does not calculate options when you select the **Override Schedule** option.

Note the behavior of the following aspects of scheduling:

- Uncheduling a fulfillment line

- Scheduling shipment sets and models

Unschedulering a Fulfillment Line

Unschedulering releases supply, so that it is available for another fulfillment line. This action leaves the fulfillment line in a state where it is awaiting schedulering. You might want to unscheduler supply when you want to release supply to make it available for a high-priority fulfillment line.

Schedulering Shipment Sets and Models

The manual schedulering options apply to configured items, models, and shipment sets. If you select just one fulfillment line in one of these groupings, then the schedulering option that you select applies to all of the fulfillment lines.

Order Orchestration Work Area Analytics: Explained

Use the analytics in the Order Orchestration work area to understand the status of orders and to help you determine the best way to facilitate fulfillment.

To understand the analytics thoroughly, it is important to know how they are calculated.

Overview Reports

The Overview reports provide a high-level view of the status of fulfillment lines.

The Fulfillment Lines by Exception report shows open fulfillment lines, each of which may have multiple concurrent exception types. That means that a fulfillment line could be counted more than once. For example, a fulfillment line could be both in jeopardy and on hold. In the Fulfillment Lines by Exception report, this line is counted once in the count of lines that have a jeopardy exception and once in the count of lines that are on hold.

The Fulfillment Lines by Status report counts open fulfillment lines only.

The On-Time Shipment Percentage by Time report shows the percentage of orders where all fulfillment lines were shipped on time.

Manage Fulfillment Line Exceptions Reports

The Manage Fulfillment Line Exceptions reports provide information about items, customers, and orders for the selected fulfillment line.

The Item Availability report shows information about the item that is associated with the selected fulfillment line.

- **Available quantity:** Amount of unallocated inventory available that is used to fill customer orders. It is calculated as the shelf quantity minus the allocated quantity.
- **Shelf quantity:** Amount of inventory that is on the shelf for a given item that is currently in the warehouse. It includes both allocated and unallocated quantities.
- **Allocated quantity:** Amount of inventory that was set aside to fill a customer's order.

Count of Holds for This Item by Hold Name counts holds applied to the orchestration order, orchestration order line, and associated fulfillment lines. The analytic is based on the selected record in the master table. A given fulfillment line may have multiple different holds. This report is cached, so the data is not real-time. The data is refreshed in accordance with the caching configuration.

Order Age by Time is calculated as follows: $[(\text{system date}) - (\text{ordered date for open orders})] / [(\text{last updated date}) - (\text{ordered date for closed orders})]$. Last updated date serves as a proxy for closed date. If an order is future dated, then it is possible to get a negative value. The Time axis is based on ordered date, meaning an order placed in May but closed in June appears on the May graph bar.

The Order Value by Time shows the value in the standardized currency. The standardized currency is the currency that is used in the Order Orchestration work area. It might not be the currency of the original transaction.

Open Orchestration Orders for This Customer by Age is calculated the same way as Order Age by Time, except Open Orchestration Orders for This Customer by Age shows only open orders for the selected record in the master table. Negative values are possible.

The Count of Holds for This Customer by Hold Name report is calculated the same way as the Count of Holds for This Item by Hold Name report.

The Fulfillment Line Exceptions for This Order report shows open fulfillment lines, each of which may have multiple concurrent exception types. That means that a fulfillment line could be counted more than once. For example, a fulfillment line could be both in jeopardy and on hold. In the Fulfillment Line Exceptions for This Order report, all the exceptions are counted for the order. Therefore, if an orchestration order has two fulfillment lines with two exceptions each, then the report counts four exceptions. The analytic is based on the selected record in the master table

The On-Time Shipment Percentage by Time report shows the percentage of orders where all fulfillment lines were shipped on time.

The Value of Fulfillment Lines by Hold Type report shows the value in the standardized currency. The standardized currency is the currency that is used in the Order Orchestration work area. It might not be the currency of the original transaction. This report does not count the same hold more than once, in the case of inherited holds. For example, a hold code that is applied to an orchestration order is inherited by the associated fulfillment line, but the hold is counted only once in the report.

Average Age of Orders by Hold is calculated the same way as Order Age by Time. This report counts holds that are applied to an orchestration order; it does not count holds that are applied to associated fulfillment lines.

Manage Orchestration Process Exceptions Reports

The Manage Orchestration Process Exceptions reports provide current and historical information about orchestration processes.

An orchestration process can have more than one exception. In the Orchestration Process Exceptions by Type report, each orchestration process is counted once for each of the exceptions that applies to it.

The Average Process Duration by Time report compares all closed orchestration process instances with the same name historically against the current selected orchestration process.

Order Age by Time is calculated as follows: $[(\text{system date}) - (\text{ordered date for open orders})] / [(\text{last updated date}) - (\text{ordered date for closed orders})]$. Last updated date serves as a proxy for closed date. If an order is future dated, then it is possible to get a negative value. The Time axis is based on ordered date, meaning an order placed in May but closed in June appears on the May graph bar.

The Count of Holds for This Customer by Hold Name report that appears on the Manage Orchestration Process Exceptions page counts orchestration processes that are associated with the selected customer in which a current task is stopped due to a hold.

The On-Time Shipment Percentage by Time report shows the percentage of orders where all fulfillment lines were shipped on time for the customer of the selected orchestration process.

The Orchestration Orders by Time report is filtered based on the customer in the selected orchestration process.

Split Fulfillment Lines: Explained

Splitting a fulfillment line is one of several ways to fulfill an orchestration order. You might want to split a fulfillment line if it is delayed because the full requested quantity is not available in a single warehouse. If you check the availability of the item in the analytics and find another warehouse that has enough of the item to fulfill the order, then you can split the fulfillment line; each fulfillment line sources part of the total desired quantity from a different warehouse. When a fulfillment line splits, the entire orchestration process splits also.

Splits can occur manually in the Order Orchestration work area or automatically as part of a fulfillment step. To split a fulfillment line manually, the orchestration process must be on a manual step. All fulfillment lines that result from a split contain all the attribute values of the original fulfillment line. The privileges assigned to you may determine your ability to override prohibitions on splits. Note the following aspects of split behavior:

- Manual splits
- Automatic splits
- Permitted splits
- Effect on shipment sets
- Effect on attachments

Manual Splits

In the Order Orchestration work area, you can split fulfillment lines using the Split Fulfillment Line action only on a manual task. Change management is not

triggered for manual splits. You cannot explicitly split a fulfillment line manually from the Check Availability page; however, the availability option you select may implicitly split the fulfillment line.

Automatic Splits

Fulfillment lines can be split automatically in several cases. If you select a fulfillment option on the Check Availability page, then the option might require a fulfillment line to be split to achieve the results that you want. Other fulfillment systems also can automatically split fulfillment lines.

Permitted Splits

Not all fulfillment lines can be split. Only fulfillment lines with the Splits Allowed attribute can be split. A split can be prohibited on any fulfillment line. An option is available to override the prohibition, but you must have the override privilege.

Effect on Shipment Sets

You cannot split fulfillment lines within shipment sets. You can split a line within a shipment set only if you break the shipment set first, creating an independent fulfillment line.

Effect on Attachments

When a fulfillment line has an attachment, the fulfillment lines that result from the split inherit the attachment.

Disabled Fulfillment Line Actions: Explained

On the Order Orchestration work area, some fulfillment line actions are disabled under certain circumstances.

The following actions are never disabled:

- View Fulfillment Details
- View Hold Details
- View Charges and Cost
- Check Availability

This table lists actions in the Order Orchestration work area and the conditions under which they are disabled.

Action	Condition Under Which Action Is Disabled
All fulfillment line actions except view	User request status is processing.
All fulfillment line actions except view	The fulfillment line is locked.
All fulfillment line actions except view	The fulfillment line is canceled.
All fulfillment line actions except view	The fulfillment line is closed.
All fulfillment line actions except view	The fulfillment line was shipped.

All fulfillment line actions except view	An orchestration process was not assigned.
All fulfillment line actions except view	The fulfillment line belongs to a model.
Unschedule	The fulfillment line is not scheduled. Scheduled ship date is null.
Reserve	The fulfillment line is not scheduled. Scheduled ship date is null.
Reserve	The fulfillment line or any associated child fulfillment line already was reserved. Reserved quantity is not null.
Reserve	A child fulfillment line is reserved.
Unreserve	Neither the fulfillment line nor any associated child fulfillment line was reserved. Reserved quantity is null.
Substitute Item	Multiple fulfillment lines are selected, and not all are for the same ordered item. When an ordered item already has been substituted, the original item is used to determine whether all the items are the same.
Substitute Item	The fulfillment line does not allow for substitution (Substitute Allowed = False), and you do not have the privilege to override the restriction.
Substitute Item	The fulfillment line is a model.
Substitute Item	The fulfillment line belongs to a shipment set.
Release Hold	The fulfillment line is not on hold.
Split Fulfillment Line	The fulfillment line belongs to a shipment set.
Split Fulfillment Line	The fulfillment line does not allow for partial shipment (Split Allowed = False) and the user does not have the privilege to override.
Split Fulfillment Line	More than one fulfillment line is selected.

Models: Explained

A model is a fulfillment line that has child fulfillment lines. An example of a model is a desktop computer, which is bundled with a monitor, keyboard, mouse, and external speakers. The desktop computer is the model; the peripherals are the child fulfillment lines. In the Order Orchestration work area, the model appears with its child fulfillment lines collapsed. To see the child fulfillment lines, expand the model.

Note the behavior of models when the following tasks are performed:

- Scheduling
- Splitting
- Holding
- Shipping
- Returning

- Billing

Scheduling

Models can be scheduled automatically, like any other fulfillment line. Only a complete model can be scheduled. All the fulfillment lines that belong to a shipment set or a model will be assigned to the same orchestration process.

Splitting

Splits are allowed at the model level only. Splits are cascaded to the child fulfillment lines proportionally.

Holding

Models are treated as a single unit, and the hold is applied at all levels.

Shipping

Shippable fulfillment lines in a model are sent to the shipping system together. A model is treated as a shipment set for shipping purposes.

Partial shipment of fulfillment lines in a model results in:

- Proportional split: A shipped model and an unshipped model.
- Non-proportional split: A model remnant, which is a partially fulfilled model with a broken model structure.

Returning

For partial receipts, model group is split into two orchestration groups, one group that has received fulfillment lines (which will move to invoicing) and the other group that has nonreturnable fulfillment lines and unreceived fulfillment lines.

Billing

Fulfillment lines in a model are sent to the billing system together.

FAQs for Manage Open Sales Orders

What happens if I change a fulfillment line?

A change to a fulfillment line almost always results in a change to the associated orchestration process. Changes include manual and automatic changes, such as change orders, edits to fulfillment line attributes, and rescheduling. Oracle Fusion Distributed Order Orchestration automatically evaluates changes and determines the adjustments that must be made to accommodate them.

Why can't I edit some attributes?

Your organization might be using processing constraints to prohibit changes to some attributes. Processing constraints control who can change an order, what types of changes can be made, and when changes can be made.

How can I search for closed fulfillment lines?

Search for closed fulfillment lines and return fulfillment lines using an advanced search.

Process Sales Orders

Process Sales Orders: Overview

Processing of orchestration orders within Oracle Fusion Distributed Order Orchestration is automated. A sales order enters Distributed Order Orchestration and is transformed into an orchestration order, using business rules defined by an administrator. Business rules are used to assign an orchestration process to one or more fulfillment lines. The orchestration process is a business process defined by an administrator that coordinates the orchestration of physical goods and activities within a single order and automates order orchestration across fulfillment systems. The forward and backward planning of the orchestration process is used to automatically identify the exceptions and spot potential problems. Changes are also processed automatically, in accordance with the change processing rules that were set up by your organization.

Changes: How They Are Processed

Changes are processed automatically using a process called change management. Changes can occur anytime, from a variety of sources: Order Orchestration work area, external fulfillment system, or order capture system. Change management often is automatic and mostly invisible, but you might see some signs of change management in the Order Orchestration work area.

How Changes Are Processed

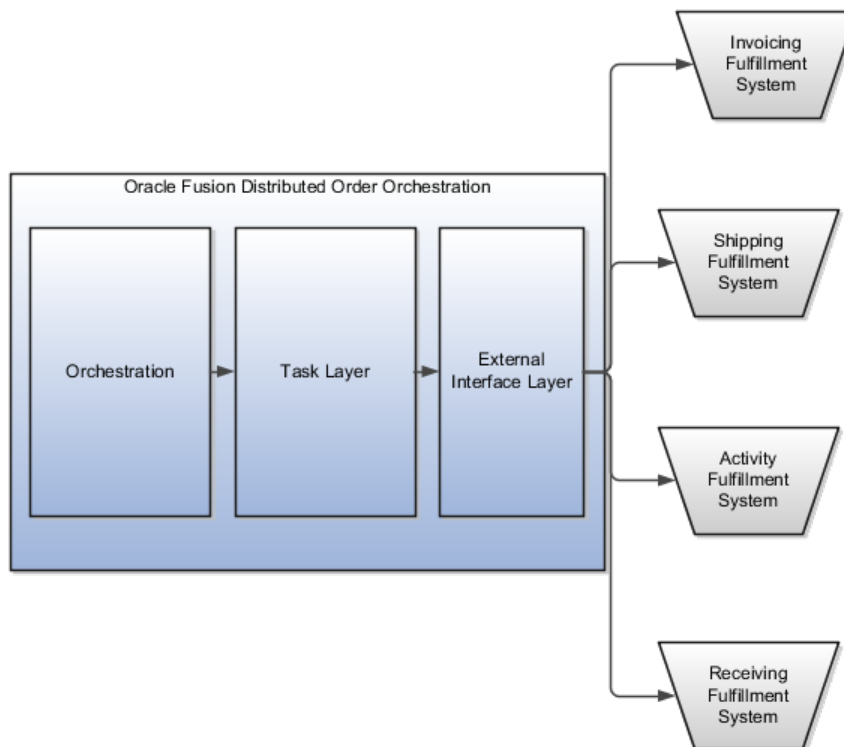
In all cases, processing constraints are used to determine whether a change is allowed. If the change is allowed, then it is considered accepted and Oracle Fusion Distributed Order Orchestration determines whether to adjust the orchestration process to accommodate the change. (Adjustments are not made in response to changes from fulfillment systems.) During this determination, the application checks which orchestration process steps were executed already and which need to be redone. When the application begins the adjustments to compensate for the change, the initial status of tasks that were executed before the change and of the orchestration process is Change Pending. When the compensation request is accepted by the fulfillment system, one by one the status of each task changes to a normal status, such as Completed. Tasks in the

Gantt chart in the Order Orchestration work area could appear to be canceled, but it is because tasks that are redone sometimes are canceled first and then redone.

Task Layer: Explained

The Oracle Fusion Distributed Order Orchestration task layer is the logical module that holds the services that are requested by other Distributed Order Orchestration modules or by external modules. In some cases, the task layer service performs a service internal to Distributed Order Orchestration. In other cases, the task layer calls a fulfillment system as part of its behavior when a service is requested of it. For example, when the task layer service Create Shipment Request is called, the service sends an outgoing Create Shipment Request to the external interface layer Create Shipment Request. The Distributed Order Orchestration external interface layer then directs this request to the appropriate fulfillment system.

This figure shows the progression of fulfillment activity. The steps of an orchestration process take place in the orchestration module, until a service needs to be called. Then a call is made to the task layer. After the service is called, the request goes to the external interface layer, where it is routed to the correct fulfillment system, for example, invoicing, shipping, activity, or receiving.



Shipment Task Layer Service: Explained

The shipment task layer service is responsible for initiating communication with shipping systems and interpreting the responses and updates from those systems. Most shipping task layer services originate in the orchestration module.

The shipment task layer service performs the following:

- Sends requests to the shipping fulfillment system
- Consolidates and sends shipment set and model lines to the shipping fulfillment system
- Receives information and status updates from the shipping system
- Splits fulfillment lines, shipment sets, and models when partial shipment occurs

Sends Requests to the Shipping Fulfillment System

The shipment task layer service sends a request to the fulfillment system where the actual shipment request will be created. When a change order is received, the task layer service changes or cancels the shipment request, if necessary. If Hold on Running Task is selected and the running task is Shipping, then when a shipping hold is applied, the task layer service makes a request to the shipping fulfillment system to hold the in-progress shipment request.

Consolidates and Sends Shipment Set and Model Lines to the Shipping Fulfillment System

The shipment task layer service consolidates the fulfillment lines of a shipment set and sends them to the shipping fulfillment system together. The task layer service treats models as shipment sets; the shippable fulfillment lines of a model are sent to the shipping fulfillment system together.

Receives Information and Status Updates from the Shipping System

The shipment task layer service receives fulfillment line details and status updates that come from the shipping system and updates the appropriate business objects in Distributed Order Orchestration. Before ship confirm when status updates occur in the shipping system, fulfillment line details, such as freight cost, tracking number, and way bill number, may be sent any time from the shipping system to Distributed Order Orchestration. Distributed Order Orchestration produces the status updates after interpreting the update from shipping. The predefined statuses picked, backordered, packed, and shipped are provided. After the delivery lines are shipped, the shipment task layer service continues to interpret updates from the shipping system and provide Distributed Order Orchestration with the relevant information from the update. If costs appear in multiple currencies in the shipping system, then they are converted before the update is sent to Distributed Order Orchestration. If the shipping unit of measure is different from the ordered unit of measure, then the fulfillment systems convert the unit of measure back to the ordered unit of measure before communicating the shipped quantity to Distributed Order Orchestration.

Splits Fulfillment Lines, Shipment Sets, and Models When Partial Shipment Occurs

The shipping task layer service performs a split when partial shipment occurs. If only partial quantity of a fulfillment line is shipped, then the task layer service splits the line into a line with the quantity that was shipped and a line with the unshipped quantity. If only some fulfillment lines of a shipment set are shipped, then the task layer service removes the unshipped lines from the shipment set. If half of the fulfillment lines of a model are shipped, then the task layer splits the model into a shipped model and an unshipped model. If the split is non-proportional, then you get a remnant.

Schedule Task Layer Service: Explained

The schedule task layer service contains the requests from Oracle Fusion Distributed Order Orchestration to order promising. The task layer service includes requests to automatically schedule an order, to unschedule an order, and to check availability of product. The schedule action applies to fulfillment lines that are waiting for manual scheduling and also to fulfillment lines that fail scheduling in the automated or manual process. The schedule action works for only unscheduled lines. No automatic rescheduling is allowed from the Order Orchestration work area.

Reservation Task Layer Service: Explained

The reservation task layer service contains the requests for managing reservations of supply from Oracle Fusion Distributed Order Orchestration to an inventory fulfillment system. A reservation acts as a guarantee of material supply for a specific order line.

Invoice Task Layer Service: Explained

The invoice task layer service is responsible for initiating communication with billing systems and interpreting the responses and updates from those systems. This task layer service sends a request to the billing system where the actual invoice or credit transactions are created. After the request is sent to the billing system, no changes are allowed on those fulfillment lines. Most of the data that is required for billing is passed from upstream order capture systems and downstream fulfillment systems and stored in Distributed Order Orchestration. Distributed Order Orchestration routes the billing request to the appropriate system.

The invoice task layer service performs the following:

- Initiates billing for sales orders and returns.
- Initiates billing for shipment sets and models.
- Receives status updates and additional information from the billing system.

Note

Invoices cannot be updated through adjustments from the feeder system like Distributed Order Orchestration. Changes to invoices are expected to come in the form of credits from corresponding return orders or due to application of prepayments rather than cancellations.

Initiates Billing for Sales Orders and Returns

The invoice task layer service sends eligible fulfillment lines to the billing system using the Create Billing Lines request. The billing information from the original sales order or return, such as price amounts, discounts, charges, payments, and sales credit and fulfillment details, is passed to the billing system.

Each time a fulfillment line is eligible for billing, the unit list price, unit selling price, discounts, and charge information are passed from the fulfillment line. Discounts at the header level are not supported. Header-level charges are supported and are sent to the billing system with the fulfillment line that is fulfilled first. Payment information is passed from the fulfillment line, if it is present. Otherwise, it is sent from the header. Prepayment information that is stored at the header is passed for every fulfillment line that is sent for billing. Sales credits are passed from the fulfillment line, if they are present. Otherwise, it is passed from the header.

Tax attributes are passed from the fulfillment line.

Fulfillment details that come from fulfillment systems also are routed to the billing system.

For return lines, the invoice line reference to the original sales order line is also sent to the billing system, along with the return reason and received and delivered quantities.

Initiates Billing for Shipment Sets and Models

The invoice task layer service sends fulfillment lines in a shipment set or model to the billing system together. If only some lines in the shipment set or model have been fulfilled, then only the fulfilled lines are passed on to the billing system.

Receives Updates from the Billing System

Once the data that is routed to the billing system is processed, the invoice task layer service receives and absorbs invoice and credit memo information, including billing amount, billing date, invoice or credit memo date, number, and legal entity information, from the billing system.

The invoice task layer service receives status updates from the billing system. The following statuses are used:

- Await Billing
- Billed

Holds Task Layer Service: Explained

The holds task layer service contains the requests for controlling holds within Oracle Fusion Distributed Order Orchestration. Holds can be communicated to

external fulfillment systems, for example, a service called HoldShipmentRequest originates in Distributed Order Orchestration and requests a hold in the shipping fulfillment system. Holds to external fulfillment systems are contained by that task layer service, in this case, the Shipping task layer service.

Activity Task Layer Service: Explained

The activity task layer service is responsible for initiating communication with fulfillment systems and interpreting the responses and updates from those systems. An activity represents a human task that must be performed as part of the fulfillment process. The activity request is sent to the activity fulfillment system, where the activity is created and fulfilled. Usually, an activity task is not fulfilled immediately. A wait service is provided to allow the orchestration process to wait for the completion of the activity.

An activity can be associated with one or more fulfillment lines. Each activity comes with attributes, such as subject, activity type, earliest start date, due date, scheduled and actual durations, percent completion, and assignees.

Activity is a predefined task type. You can create new activity task types to accommodate your organization's business needs. You can extend activities through extensible flexfields.

Note

Partial fulfillment is not supported. An activity step must be completed in full before it can move to the next step.

The activity task layer service performs the following:

- Sends requests to the activities management system.
- Receives and absorbs activity status updates.
- Enables creation of new activity task types and activity defaulting.

Sends Requests to the Activities Management System

The activity task layer service sends a Create Activity request to the activities management system where the actual activity will be created. When a change order is received or an order is canceled, the task layer service changes or cancels the activity as necessary. When a hold is applied to an order, the task layer service requests the activities management system to hold the activity that is in progress.

Receives and Absorbs Activity Status Updates

You can schedule a process to obtain an activity status periodically.

Enables Creation of New Activity Task Types and Activity Defaulting

You can create new activity task types to accommodate your organization's business needs. You can also enable activity defaulting at either the task type or process step level. The subject of the activity is defaulted from the process step name, unless the default value is set for either task type or process step.

By defaulting activity type, you can categorize activities, so that the activities management system can perform the designated business logic and validation based on the activity type. An activity template is a blueprint for an activity. Some activities management systems allow you to create activity templates to address the various requirements of a human task. To use this advanced feature in the activities management system, when defining task types and orchestration processes, specify the designated activity template for a task type or process step to create the activity according to the predefined template.

Return Task Layer Service: Explained

The return task layer service is responsible for initiating communication with receiving systems and interpreting the responses and updates from those systems. Some of the requests originate in Distributed Order Orchestration and involve the creation, update, or cancellation of a receipt advice in the receiving system.

The return task layer service performs the following:

- Support simple return.
- Support returned models.
- Receive status updates from the receiving system.
- Split fulfillment line due to partial delivery or return to customer.

Support Simple Return

The return task layer service sends a create receipt advice request to the fulfillment system where the actual receipt advice will be created. If any changes to the original returned sales order are received, then the task layer service requests a change to the existing receipt advice. The request may include one or more attribute updates, such as an increase or decrease in the expected receipt quantity or a change in non-quantity attributes. When items are returned, the task layer service creates a change receipt advice or cancel receipt advice, as necessary. If the original returned sales order line is completely canceled, then the task layer service cancels the receipt advice. Generally, cancellation is allowed until the returned items are received. If the expected delivered quantity is greater than the actual delivered quantity on the receipt advice, then partial cancellation can be requested for the remaining quantity.

Support Returned Models

The return task layer can process partial receipts, such as the return of only some items of a model.

Receive Status Updates from the Receiving System

The return task layer service receives and absorbs status updates from the receiving system for selective triggering events. For example, the receipt of goods on the receiving desk triggers creation of the receipt advice.

The following events in the receiving system can trigger status updates in Distributed Order Orchestration:

- Receipt of goods on the receiving deck, when the actual receipt is created.
- Delivery of goods into inventory.
- Return of goods to a customer.
- Correction after a receipt transaction. (Deliver transactions cannot be returned.)

Split Fulfillment Line Due to Partial Delivery or Return to Customer

If only part of the expected returned quantity is received, then the return task layer splits the fulfillment line into a line with a status of Delivered for the expected returned quantity and another line with the undelivered expected quantity. A model or kit group is split into two orchestration groups, one group that will have received fulfillment lines and the other group that has nonreturnable fulfillment lines and unreceived fulfillment lines.

Template Task Layer: Explained

The template task layer is a Web service wrapper that makes it possible to create and use your own task types, while maintaining data integrity in Oracle Fusion Distributed Order Orchestration. Use the template task layer to extend the business functionality of Distributed Order Orchestration beyond the seeded task types. The result of maintaining data integrity within Distributed Order Orchestration is that the task type that you create appears throughout the relevant areas of the application, for example, in the Order Orchestration work area. In addition, using the template task layer ensures correct behavior of the following functionality: Status update, wait steps, forward planning, jeopardy, holds processing, split processing, change management, and error recovery.

Custom services that are created using the template task layer function the same way as predefined task layer services.

Use the template task layer for:

- Creating custom tasks and services.
- Validating mandatory Distributed Order Orchestration attributes.
- Preprocessing the outbound service request.
- Postprocessing the inbound service response.
- Determining which Distributed Order Orchestration transaction data to update

Creating Custom Tasks and Services

Implementers can define new fulfillment task types, fulfillment tasks, and services that can be called in an orchestration process definition.

Validating Mandatory Distributed Order Orchestration Attributes

Implementers can enable the service to determine whether all of the mandatory data is present in the service data object.

Preprocessing the Outbound Service Request

Implementers can add preprocessing logic to the actions built into the template task layer service. Your organization might want to add preprocessing logic that defaults data onto the outbound request or validates the data.

Postprocessing the Inbound Service Response

Implementers can add postprocessing logic to the actions that are built into the template task layer service. Your organization might want to add postprocessing logic that defaults logic onto the inbound request, validates the inbound response, or interprets any attributes or messages that are returned by the fulfillment system that might indicate the need for split processing.

Determining Which Distributed Order Orchestration Transaction Data to Update

Implementers can determine which of the Distributed Order Orchestration transaction tables must be updated as a result of the external service call.

Fulfillment Order Task Layer Explained

The fulfillment order task layer is designed to enable Oracle Fusion Distributed Order Orchestration to integrate with enterprise resources planning (ERP) systems that are closely coupled with an order management system and which do not have any open-ended interface through which Distributed Order Orchestration can connect. With the fulfillment order task layer service, you can leverage existing ERP systems.

Aggregate Messages to the ERP System

The fulfillment order task layer aggregates and stores messages that are bound for the ERP system. After the aggregator receives a complete set of messages, the aggregator publishes the information from each related message as a single aggregated message. Aggregating the messages minimizes issues with the timing of the messages.

A timeout determines how long the aggregator waits to aggregate messages. When the time expires, the aggregator publishes whatever messages have been accumulated. If an order has several fulfillment lines and the lines complete the task before the timeout occurs, then all messages are sent as soon as the task is completed.

Aggregation is performed in one fulfillment system only.

The default timeout is five minutes. You can set your own timeout using business rules. The timeout must be expressed in minutes.

Receive Status Updates from an ERP System

The fulfillment order task layer receives interim and final status updates from an ERP system.

Publish Updates to the ERP System for Change Orders

The fulfillment order task layer updates the ERP system every time a change order arrives that affects the fulfillment of the orchestration order.

Sales Order Transformation to Orchestration Order: Explained

After an original sales order or change order enters Oracle Fusion Distributed Order Orchestration, the order is broken down and transformed into an orchestration order. The orchestration order consists of orchestration order lines, which correspond to fulfillment lines. You may find that the orchestration order and fulfillment lines that result from the transformation do not necessarily mirror the original sales order because orchestration orders are configured for fulfillment requirements, while sales orders are configured for sales requirements. For example, sales orders may contain loyalty and marketing information that is not required for orchestration. The number of orchestration order lines and sales order lines also might not match. The orchestration order lines could appear in a different sequence than the sales order lines. When you view an orchestration order on the Manage Fulfillment Line Exceptions or Manage Orchestration Process Exceptions page, only the first line of a model appears. You must expand the model to see all the fulfillment lines within it.

In some cases, a sales order might not be transformed to an orchestration order. Transformation could fail for several reasons: Transformation rules were not set up correctly in Distributed Order Orchestration, mandatory attributes were not sent with the sales order, or a validation failure occurred. The order administrator must examine the error messages to determine the exact reason.

Activities: Explained

An activity is an event that takes place outside Oracle Fusion Distributed Order Orchestration. For example, an orchestration process might include an activity task type to configure a network router. An activity contains the details needed to complete the task. Completion of the activity is reported to Distributed Order Orchestration.

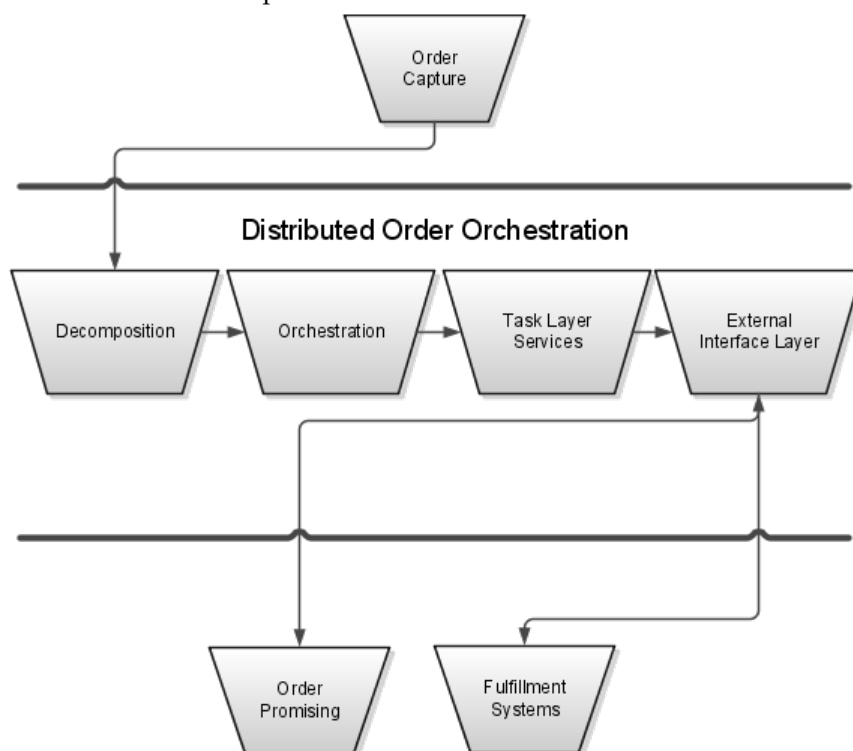
Orchestration Process Assignment Failure: Explained

After a sales order enters Oracle Fusion Distributed Order Orchestration, it is transformed to an orchestration order with one or more fulfillment lines. The application then attempts to assign an orchestration process to a fulfillment line or group of fulfillment lines. An orchestration process might not be assigned due to a system failure or a problem with the process assignment rules. For example, a rule that applies to the orchestration order data set might not exist; if a default orchestration process is not designated, then an orchestration process cannot be assigned. You can determine the reason for the process assignment failure by viewing the messages in the Order Orchestration work area. If you have the Error Recovery Manager role, then you can select Assign Lines to Process. If that does not work, then an administrator might need to edit the process assignment rules.

Define Sales Order Fulfillment

Oracle Fusion Distributed Order Orchestration Components: How They Work Together

The Oracle Fusion Distributed Order Orchestration architecture is situated between one or more order capture systems and one or more fulfillment systems. When a sales order enters Distributed Order Orchestration, the application components process the order, first by breaking it down into logical pieces that can be fulfilled, then assigning an appropriate set of sequential steps to fulfill the order, and, finally, calling services to carry out the steps. Throughout the process, Distributed Order Orchestration continues to communicate with the order capture and fulfillment systems to process changes and update information. This figure shows the components that affect order processing. A sales order enters Distributed Order Orchestration from the order capture application. In Distributed Order Orchestration, the sales order proceeds through decomposition, orchestration, task layer services, and the external interface layer before proceeding to fulfillment systems. The following explanations fully describe the components within Distributed Order Orchestration.



Decomposition

During decomposition, the application breaks down the sales order and uses defined product transformation rules to transform the sales order into an orchestration order. Then the fulfillment lines are grouped and assigned to designated orchestration processes with step-by-step fulfillment plans. An orchestration process is a predefined business process that coordinates the orchestration of physical goods and activities within a single order and automates order orchestration across fulfillment systems.

Orchestration

Orchestration is the automated sequence of fulfillment steps for processing an order. The orchestration process provides the sequence and other important information, such as forward and backward planning, how to compensate for changes, and which statuses to use.

During orchestration, task layer services are called to carry out the steps of the orchestration process.

Task Layer Services

Task layer services execute user-defined fulfillment process steps and manage fulfillment tasks. These services send information to downstream fulfillment systems and interpret the responses and updates from those systems. For example, task layer service Create Shipment request is invoked by a ship order process to send a shipment request to the shipping system.

External Interface Layer

The external interface layer manages the communication between Distributed Order Orchestration and external fulfillment systems. Its primary functions are routing the fulfillment request and transforming the data.

Orchestration Lookups: Explained

Oracle Fusion Distributed Order Orchestration provides lookups that you can use to optionally define values during processing. The majority of lookups are system-level, and they cannot be changed. You can make certain changes to user-level and extensible lookups.

User-Level Lookups

Distributed Order Orchestration provides one user-level lookup: DOO_ACTIVITY_TYPE.

Users can:

- Insert new codes.
- Update the start date, end date, and enabled fields.
- Delete codes.
- Update tags.

Extensible Lookups

The following extensible lookups are provided:

- DOO_HLD_RELEASE_REASON
- DOO_MSG_REQUEST_FUNCTION
- DOO_RETURN_REASON
- DOO_SUBSTITUTION_REASON

With extensible lookups, users can:

- Insert new codes.
- Update the start date, end date, enabled fields and tag, but only if the code is not seed data.
- Delete codes, but only if the code is not seed data.

Users cannot:

- Update the module.
- Delete the lookup type.

Orchestration Profile Management: Points to Consider

Oracle Fusion Distributed Order Orchestration provides several product-specific profile values. Some control behavior in the Order Orchestration work area, while others control the receipt and transformation of sales orders into orchestration orders. Most have predefined values, so you do not need to configure them, unless your organization requires different profile values.

Currency Conversion Type

This profile option defines the value to use during any currency conversion in the Order Orchestration work area. The value is a conversion type. You can update the profile option at the site and user levels.

Display Currency

This profile option defines the currency used to display the amount in the Order Orchestration work area. The value is a currency. You can update the profile option at the site and user levels.

Required Overview Status Filter

This profile option defines the default customer used to filter the summary of status data on the Overview page of the Order Orchestration work area and allows the user to view summary data only one customer at a time by removing the All option. No value is provided, by default. If you need to use it for performance reasons, then enter one of your customer IDs. You can update the profile option at the site level.

Retain Sales Order Number for Orchestration Order Number

This profile option specifies whether to use the sales order number as the orchestration order number during sales order transformation. The default value is N. You can update the profile at the site and user levels.

User Request Waiting Period in Seconds

This profile option specifies the number of seconds to wait after an action is taken to allow asynchronous services to complete before presenting a confirmation or warning message in the Order Orchestration work area. The default value is 5. You can update the profile option at the site level.

Oracle Fusion Distributed Order Orchestration Extensible Flexfields: Explained

An extensible flexfield is similar to a descriptive flexfield in that it is an expandable data field that is divided into segments, where each segment is represented in the application database as a single column. Extensible flexfields support a one-to-many relationship between the entity and its extended attribute rows. Using extensible flexfields, you can add as many context-sensitive segments to a flexfield as you need. You can set up extensible flexfields for a fulfillment line or on other entities that support extensible flexfields. Extensible flexfields are useful primarily when you need to segregate attributes by task layer or capture multiple contexts to group them based on function.

Transactional Entities That Support Extensible Flexfields

You can use extensible flexfields for the following transactional entities on the orchestration order object.

- Headers
- Orchestration order lines
- Fulfillment lines
- Fulfillment line details
- Price adjustments
- Sales credits
- Payments
- Lot serial
- Activities

Oracle Fusion Distributed Order Orchestration Extensible Flexfield Uses: Explained

Use extensible flexfields to send and receive additional information between Oracle Fusion Distributed Order Orchestration and integrated applications, write business rules, process changes, and display additional attributes on the Order Orchestration work area.

Receive Additional Information or Attributes from an Order Capture Application

The sales order that Oracle Fusion Distributed Order Orchestration receives contains a predefined set of attributes. Your business process may require that

you capture additional information or attributes on the sales order to use during order fulfillment. Distributed Order Orchestration uses extensible flexfields to receive the additional set of information or attributes that are captured on the sales order and use them during the fulfillment orchestration process.

Send Additional Information Relevant to a Fulfillment Execution Application

The task layers use a specific fulfillment request object to initiate a fulfillment request in a downstream application. Using extensible flexfields, Distributed Order Orchestration can pass any additional information that you set up and capture on the orchestration order, beyond the predefined set of attributes, during implementation.

Receive Additional Information from a Fulfillment Execution Application

During the response to a fulfillment request, a fulfillment execution application can send various attributes that may have business value and which need to be seen either from the Order Orchestration work area or in the order capture application. This additional information also can be used in the next set of tasks, if that information is relevant to the set of tasks that follow this task. Using extensible flexfields, Distributed Order Orchestration can receive additional sets of attributes from the fulfillment execution applications.

Write Business Rules

You can use extensible flexfield attributes to write business rules for Distributed Order Orchestration. You can use extensible flexfield attributes to write rules for the following rules implementations:

- Transformation rules
- Pretransformation and posttransformation defaulting rules
- Process assignment rules
- External interface routing rules

Process Changes

You can use extensible flexfields during change management. You can designate an extensible flexfield as an order attribute that indicates that a change occurred. An extensible flexfield is interpreted as a single unit for change processing. Changes are not allowed from the Order Orchestration work area and are supported only through the services.

Display Attributes in the Order Orchestration Work Area

The Order Orchestration work area displays the following extensible flexfields.

- Headers
- Orchestration order lines
- Fulfillment lines
- Fulfillment line details
- Activities

The extensible flexfield attributes are read-only. Users cannot edit them in the Order Orchestration work area.

Oracle Fusion Distributed Order Orchestration Extensible Flexfield Setup: Explained

To set up Oracle Fusion Distributed Order Orchestration extensible flexfields, you must define flexfields, deploy them, synchronize them with business rules, synchronize the SOA artifacts, and configure the enterprise business object.

The specific steps follow:

1. Run the Publish Extensible Flexfields Attributes process to create categories for the extensible flexfields.
2. Define categories, contexts, and associated segments along with value sets for each extensible flexfield that you want to enable through the Manage Extensible Flexfields setup.
3. Deploy the flexfield.
4. Run the Publish Extensible Flexfields Attributes process to synchronize the extensible flexfield attributes with Oracle Business Rules.
5. Execute the SOA composite UpdateSOAMDS for synchronizing the SOA artifacts.
6. Extend the enterprise business object.
7. Map the enterprise business object attributes with the extensible flexfield attributes.

Extending the Enterprise Business Object

Oracle Fusion Distributed Order Orchestration uses enterprise business objects to interact with external systems. An enterprise business object is made up of a business component, shared components, common components, reference components, common enterprise business objects, choice components, and attributes. These components are nested as required to create a sophisticated content model with varying cardinality from zero to one or unbounded. A custom element is defined in these component types and can be used to extend the properties of the component. The custom element then can be further mapped to extensible flexfield attributes in the interfaces.

The enterprise business objects are delivered as a set of XSD files. For every enterprise business object, a custom XSD file is provided in which all customer extensions are stored. Using customer extensions, you can include on the sales order additional attributes that your organization needs. For example, assume that you want to add DeliverToParty to the order header because the shipping system can honor this information. To integrate with the shipping system, you must extend the Sales Order enterprise business object. To add this new attribute at the header level, edit the following part of the CustomSalesOrderEBO.xsd schema definition:

```
<xsd:complexType name="CustomSalesOrderScheduleType"/>
```

After adding the attributes, this section of the schema definition looks like:

```
<xs:complexType name="CustomSalesOrderScheduleType">  
<xs:sequence>
```



```
<xsd:element ref="corecom:DeliverToPartyReference" minOccurs="0">
</xsd:element>
</xs:sequence>
</xs:complexType>
```

The Sales Order enterprise business object is now ready to carry the custom attributes for DeliverToPartyReference. The custom attributes can be either from the common components library or can be new elements or attributes that are directly added if they did not exist in the common components library. Note that the extension of the underlying Sales Order enterprise business object also extends all enterprise business messages that reference the Sales Order enterprise business object. In the case of the Receive and Transform service that is used by the order capture application to submit an order, this is ProcessSalesOrderFulfillment.

Transformation Extensions

The default transformations for the existing schemas may not be sufficient for some of your organization's specific business operations. You might want to add elements to the enterprise business object schemas as explained previously and then change transformation maps for the newly added elements to transfer the information from one application to the other.

At implementation time, the transformation maps that are associated with the external-facing interfaces must be modified to map the extensible flexfield attributes to the enterprise business object attributes.

Manage Hold Codes

Hold Definitions: Explained

Holds pause action on the business objects and services to which they are applied. In Oracle Fusion Distributed Order Orchestration, holds can come from an order capture system or the Order Orchestration work area. You define codes for use throughout Distributed Order Orchestration. The codes you define in Distributed Order Orchestration are for holds that originate in this application only. When you define hold codes, you indicate to which services the hold can be applied. You can also create a hold code that applies a hold to all services. Task layer services check to see whether any hold code is attached to the fulfillment line or order for one or more tasks in the orchestration process.

A hold that is applied in Distributed Order Orchestration can be released by the same application only, either by a user or by an orchestration process. A hold is applied by an orchestration process only when there is an existing hold request, either from the order capture application or from the Order Orchestration work area user. For example, an orchestration process is at the scheduling step when an order capture user sends a request to hold the shipping task. Distributed Order Orchestration stores the request until the orchestration process gets to the shipping step. At that point, the application searches for existing requests and applies them. When an orchestration process is canceled, associated holds are released automatically. Otherwise, the Order Orchestration user must release holds manually.

Only an order capture user can release a hold applied in the order capture application.

Hold Codes from Other Applications

When a hold enters Distributed Order Orchestration from an order capture or order fulfillment application, it is transformed and becomes part of the orchestration order.

Manage Orchestration Source Systems

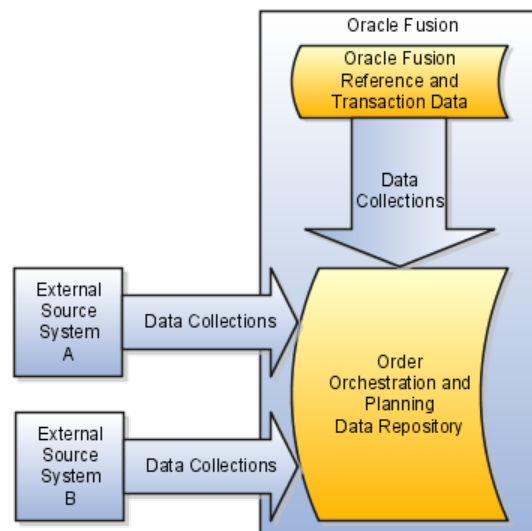
Managing Data Collection Source Systems: Explained

To populate the order orchestration and planning data repository, you collect data from external source systems, such as external fulfillment source systems and external order capture source systems, and from the Oracle Fusion source system. You manage which source systems are data collection source systems by defining collections parameters and enabling which source systems allow collections.

You manage two categories of source systems for data collections:

- External source systems
- The Oracle Fusion source system

The following figure illustrates data collections from three source systems. Two of the source systems are external source systems. One of the source systems is the Oracle Fusion source system.



External Source Systems

Your business may have many external fulfillment and external order capture source systems. For each external source system from which you need to collect data to include in the order orchestration and planning data repository, define the data collection parameters, and enable the source system for collections. For the Version data collection parameter, the choices are Other or Oracle Fusion.

The Oracle Fusion Source System

The order orchestration and order promising processes use data stored in the order orchestration and planning data repository. Some of the data that needs to

be in the repository originates in the Oracle Fusion source system. To collect data from the Oracle Fusion source system, include the Oracle Fusion source system as a source system for data collection. Define the data collection parameters for the Oracle Fusion source system, and enable the source system for collections.

Defining Data Collection Parameters: Points to Consider

For each system from which you intend to collect data to populate the order orchestration and planning data repository, you define and maintain the source system data collection parameters.

For each source system, you complete the following for the data collection parameters:

- Specify the time zone.
- Specify the version, order orchestration type, and planning type.
- Define the number of database connections, parallel workers, rows per processing batch, and cached data entries.
- Enable collections allowed.
- Enable data cross-referencing.

Specify the Time Zone

You must specify the time zone for the source system because the time stamps contained in collected data are converted from the time zone used in the source system to the time zone used for all data stored in the order orchestration and planning data repository. Using the same time zone for all data stored in the order orchestration and planning data repository facilitates correct results when calculations are performed using attributes that store dates. For example, if the source system uses the US Eastern time zone, but the order orchestration and planning data repository stores all data in the US Pacific time zone, then a supply with a due date and time of July 10th 04:00 PM in the source system is stored in the order orchestration and planning data repository with a due date of July 10th 01:00 PM.

Specify the Version, Order Orchestration Type, and Planning Type

You must define one, and only one, source system with the Version attribute equal to Oracle Fusion and the Order Orchestration Type attribute equal to Order Orchestration.

You may define many source systems with the Version attribute equal to Other. For the source systems with the Version attribute equal to Other, the Order Orchestration Type attribute can equal Fulfillment or Order Capture and the Planning Type attribute can equal Fulfillment. Any combination of these values is allowed to describe the purpose of the source system, but you must provide a value for at least one of these type parameters. These parameters do not impact the behavior of the collections process.

Note

Once you have saved a system with the Version attribute equal to Oracle Fusion, you cannot change the value for the Version attribute.

Define the Number of Database Connections, Parallel Workers, Rows Per Processing Batch, and Cached Data Entries

These parameters affect the usage of system resources. The table below defines what each parameter does and provides guidelines for setting it.

Parameter	What the Parameter Does	A Typical Value for the Parameter
Number of Database Connections	Defines the maximum number of database connections the source server can create during the collection process. This controls the throughput of data being extracted into the Source Java program.	10
Number of Parallel Workers	Defines the maximum number of parallel workers (Java threads) used to process the extracted data. The number here directly impacts the amount of CPU and memory used during a collection cycle.	30
Number of Rows per Processing Batch	Define the number of records to process at a time. The idea is to allow the framework to process data in byte-size chunks. A batch too small may cause extra overhead while a batch too big might peak out memory or network bandwidth.	10,000
Cached Data Entries in Thousands	During data collections, various lookup and auxiliary data are cached in the collection server to support validation. For example, currency rate may be cached in memory. This parameter controls the maximum number of lookup entries cached per lookup to prevent the server from occupying too much memory.	10,000

Enable Collections Allowed

Before enabling a source system for collections, ensure your definition of the other parameters are complete for the source system. Ensure you have defined values for all applicable attributes, and where applicable, you have enabled organizations for collections or for ATP Web services.

Enable Data Cross-Referencing

When you enable a source system for data cross-reference, the data collections from the source system requires additional processing steps to check for and to cross-reference data during collections. These additional processing steps can have a performance impact. For data collected from third-party source systems,

it is recommended that any cross-referencing needs be completed externally, before the data is presented for collections.

Enabling Organizations for Data Collections: Points to Consider

From the list of organizations for each source systems, you designate which organizations will have their data collected when a collections process collects data from the source system.

Deciding Which Organizations to Enable for Collections

To determine which organizations to enable for collections, analyze the sourcing strategies for your company, the type of organization for each organization in the list, and any other business requirements that would determine whether system resources should be expended to collect data from that organization. If the data from that organization would never be used by order promising or order orchestration, no need to collect the data.

For example, consider a scenario where the list of organizations for a source system includes 20 manufacturing plants and 10 distribution centers. Because the business requirements specify that the movements of materials from the manufacturing plants to the distribution centers are to be controlled separately from order orchestration and order promising, there are no sourcing rules that include transferring from one of the manufacturing plants. For this scenario, you would only enable the 10 distribution centers for collections.

FAQs for Orchestration Source Systems

Can I add a new source system to the list of data collection source systems?

No. You cannot add additional source systems when managing source systems for data collections for the order orchestration and planning data repository.

Source systems must first be defined in the Trading Community Model. For the system to be listed as one of the systems from which to choose from when managing source systems, the definition of the system in the Trading Community Model must enable the system for order orchestration and planning.

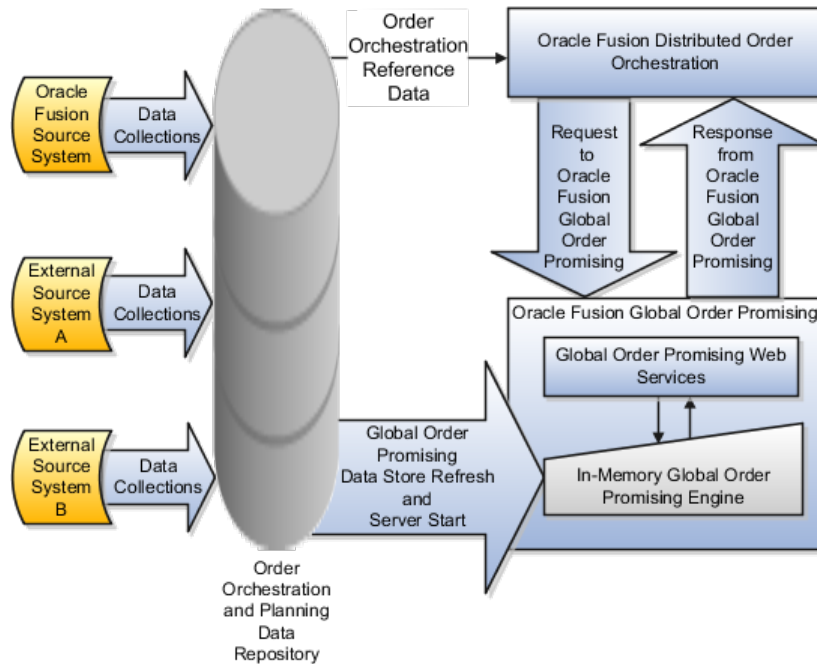
Collect Orchestration Reference and Transaction Data

Data Collections, Order Orchestration, and Order Promising: How They Fit Together

You perform data collections to populate the order orchestration and planning data repository. The collected data is used by Oracle Fusion Distributed Order Orchestration and Oracle Fusion Global Order Promising.

The following figure illustrates that the order orchestration and planning data repository is populated with data from external source systems and from the Oracle Fusion source system when you perform data collections. Oracle Fusion Distributed Order Orchestration uses some reference data directly from the repository, but the Global Order Promising engine uses an in-memory copy of the data. After data collections are performed, you refresh the Global Order

Promising data store with the most current data from the data repository and start the Global Order Promising server to load the data into main memory for the Global Order Promising engine to use. When Oracle Fusion Distributed Order Orchestration sends a scheduling request or a check availability request to Oracle Fusion Global Order Promising, the Global Order Promising engine uses the data stored in main memory to determine the response.



Data Collections

You perform data collections to populate the order orchestration and planning data repository with data from external source systems and from the Oracle Fusion source system.

Order Orchestration

Oracle Fusion Distributed Order Orchestration uses some reference data directly from the order orchestration and planning data repository. You must perform data collections for the order orchestration reference entities even if you are not using Oracle Fusion Global Order Promising.

Important

Before collecting data from an Oracle Fusion source system, you must define at least one organization for the source system. After you have defined at least one organization for the source system, you must update the organization list for the source system on the Manage Planning Source Systems page or Manage Orchestration Source Systems page, and enable at least one organization for collections. If there are no organizations enabled for collections when a collections process runs, the collections process will end with an error.

Order Promising

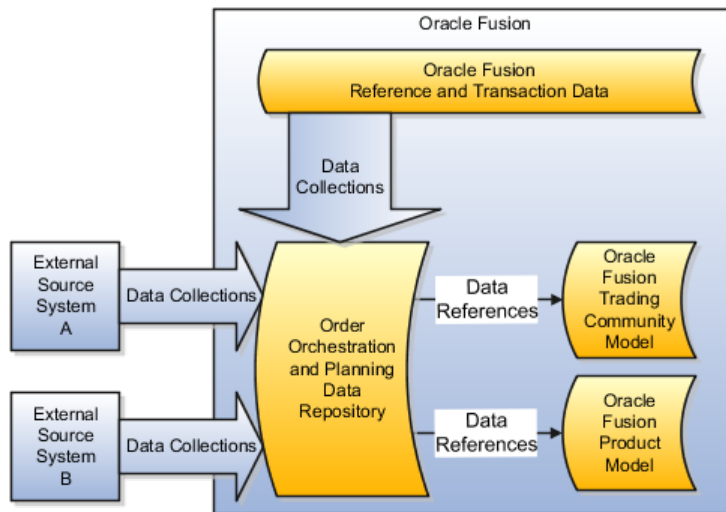
The Global Order Promising engine uses an in-memory copy of the data from the order orchestration and planning data repository. When Oracle

Fusion Distributed Order Orchestration sends a scheduling request or a check availability request to Oracle Fusion Global Order Promising, the Global Order Promising engine uses the data stored in main memory to determine the response to send back to order orchestration. After a cycle of data collections is performed, you refresh the Global Order Promising data store with the most current data from the data repository and start the Global Order Promising server to load the data into main memory for the Global Order Promising engine to use.

Collecting Data for the Order Orchestration and Planning Data Repository: Explained

The order orchestration and planning data repository provides a unified view of the data needed for order orchestration and order promising. You manage data collection processes to populate the data repository with data collected from external source systems and from the Oracle Fusion source system. You manage the data collection processes to collect the more dynamic, transaction data every few minutes and the more static, reference data on a daily, weekly, or even monthly schedule. The data collected into the data repository contains references to data managed in the Oracle Fusion Trading Community Model and to data managed in the Oracle Fusion Product Model. The data managed in these models is not collected into the order orchestration and planning data repository.

The following figure illustrates that the order orchestration and planning data repository is populated with data collected from external source systems and from the Oracle Fusion source system. The data repository does not contain data managed by the Oracle Fusion Trading Community Model and the Oracle Fusion Product Model. The data collected into the data repository references data managed in the models.



When you plan and implement your data collections, you determine which entities you collect from which source systems, the frequency of your collections from each source system, which data collection methods you will use to collect which entities from which source systems, and the sequences of your collections. Consider these categories of data when you plan your data collections:

- Data collected for order promising
- Data collected for order orchestration
- Data not collected into the order orchestration and planning data repository

Data Collected for Order Promising

The following data is collected and stored to support order promising:

- Existing supply including on-hand, purchase orders, and work orders
- Capacity including supplier capacity and resource capacity
- Related demands including work order demands and work order resource requirements
- Planned supply including planned buy and make orders
- Reference data including calendars, transit times, and routings

Important

After performing data collections, you must refresh the Order Promising engine to ensure it is using the data most recently collected.

Data Collected for Order Orchestration

The following data is collected and stored to support order orchestration:

- Order capture and accounts receivable codes
- Accounting terms and currencies

Tip

Use the Review Planning Collected Data page or the Review Order Orchestration Collected Data page to explore many of the entities and attributes collected for the order orchestration and planning data repository.

Data Not Collected into the Order Orchestration and Planning Data Repository

Data collected into the order orchestration and planning data repository includes attributes, such as customer codes, that refer to data not collected into the data repository. Most of the data references are to data in the Oracle Fusion Trading Community Model or in the Oracle Fusion Product Model. Some of the data references are to data outside the models, such as item organizations and inventory organizations. To manage data collections effectively, especially the sequences of your collections, you must consider the data dependencies created by references to data not collected into the data repository.

References to data in the Oracle Fusion Trading Community Model include references to the following:

- Source systems
- Geographies and zones
- Customers
- Customer sites

References to data in the Oracle Fusion Product Model include references to the following:

- Items, item relationships, and item categories
- Item organization assignments
- Structures

Data Collection Entities: Explained

When you collect data for the order orchestration and planning data repository, you specify which of the data collection entities to collect data for during each collection. When you plan your data collections, you plan which entities to collect from which source systems and how frequently to collect which entities. One of the factors you include in your planning considerations is the categorizations of each entity. One way entities are categorized is as reference entities or transaction entities. You typically collect transaction entities much more frequently than reference entities.

Another way entities are categorized is as source-specific entities or global entities. For global entities the order in which you collect from your source systems must be planned because the values collected from the last source system are the values that are stored in the data repository.

When you plan your data collections, you consider the following categorizations:

- Source-specific entities
- Global entities
- Reference entities
- Transaction entities

You also consider which entities can be collected from which types of source systems using which data collection methods as follows:

- Entities you can collect from the Oracle Fusion source system and from external source systems
- Entities you can collect only from external source systems

Source-Specific Entities

When you collect data for a source-specific entity, every record from every source system is stored in the order orchestration and planning data repository. The source system association is maintained during collections. The data stored in the data repository includes the source system from which the data was collected.

For example, you collect suppliers from source system A and source system B. Both source systems contain a record for the supplier named Hometown Supplies. Two different supplier records will be stored in the data repository for the supplier named Hometown Supplies. One record will be the Hometown Supplies supplier record associated with source system A and the second record will be the Hometown Supplies supplier record associated with source system B.

The majority of the data collections entities are source-specific entities.

Global Entities

When you collect data for a global entity, only one record for each instance of the global entity is stored in the order orchestration and planning data repository. Unlike source-specific entities, the source system association is not maintained during collections for global entities. The data stored in the data repository for global entities does not include the source system from which the data was collected. If the same instance of a global entity is collected from more than one source system, the data repository stores the values from the last collection.

For example, you collect units of measure (UOM) from three source systems and the following occurs:

1. During the collection of UOM from source system A, the Kilogram UOM is collected.

This is first time the Kilogram UOM is collected. The Kilogram record is created in the data repository.

2. During the collection of UOMs from source system B, there is no collected UOM with the value = Kilogram

Since there was no record for the Kilogram UOM in source system B, the Kilogram record is not changed.

3. During the collection of UOMs from source system C, the Kilogram UOM is also collected.

Since the collections from source system C include the Kilogram UOM, the Kilogram record in the data repository is updated to match the values from source system C.

The following entities are the global entities:

- Order orchestration reference objects
- Units of measure (UOM) and UOM conversions
- Demand classes
- Currency and currency conversion classes
- Shipping methods

Tip

When you collect data for global entities from multiple source systems, you must consider that the last record collected for each occurrence of a global entity is the record stored in the order orchestration and planning data repository. Plan which source system you want to be the source system to determine the value for each global entity. The source system that you want to be the one to determine the value must be the source system that you collect from last.

Reference Entities

Reference entities are entities that define codes and valid values that are then used regularly by other entities. Units of measure and demand classes are two examples of reference entities. Reference entities are typically static entities

with infrequent changes or additions. Whether an entity is reference entity or a transaction entity does not impact how it is stored in the order orchestration and planning data repository.

You consider whether an entity is a reference entity or a transaction entity when determining which collection method to use to collect data for the entity. You typically use the staging tables upload method to collect data for reference entities from external source systems. You typically used the targeted collection method to collect data for reference entities from the Oracle Fusion source system unless the reference entity is one of the entities for which the targeted collection method is not possible.

Transaction Entities

Transaction entities are the entities in the data repository that store demand and supply data. Because the data for transaction entities changes frequently, you typically use the web services upload method to collect data for transaction entities from external source systems. You typically use the continuous collection method to collect data for transaction entities from the Oracle Fusion source system.

Entities You Can Collect From the Oracle Fusion Source System and From External Source Systems

Many of the data collection entities can be collected from both types of sources systems. For the following entities you can use any of the collections methods:

- Approved supplier lists
- Calendars
- Calendar associations
- Interlocation shipping networks
- Item costs
- On hand
- Organization parameters
- Purchase orders and requisitions
- Subinventories
- Suppliers
- Units of measure

For the following entities you can only use the Web service upload method to collect data from external source systems:

- Currencies
- Order orchestration reference objects
- Shipping methods

Entities You Can Collect only From External Source Systems

Many of the data collection entities can be only collected from external sources systems. For these entities, you can use both methods for collecting data from

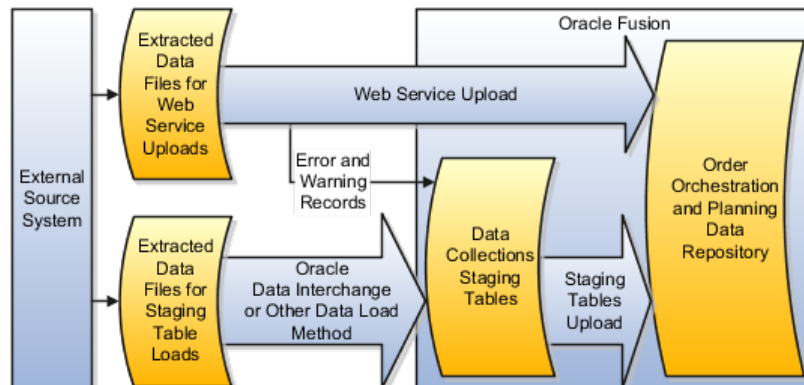
external source systems. Remember to consider frequency of change and volume of data in your considerations of which methods to use to collect which entities. The following are the entities you can only collect from external sources systems:

- Customer item relationships
- Demand classes
- Planned order supplies
- Routings
- Resources
- Resource availability
- Sourcing
- Supplier capacities
- Work-in-process supplies
- Work-in-process component demands
- Work-in-process resource requirements

Data Collection Methods for External Source Systems: Explained

To populate the order orchestration and planning data repository with data collected from external source systems, you use a combination of two data collection methods. The two methods are Web service uploads and staging tables uploads.

The following figure illustrates the two data collection methods, Web service uploads and staging tables uploads, used to collect data from external source systems. The figure illustrates that both methods require programs to be written to extract data from the external source systems. For Web service uploads, you load the data from the extracted data files directly into the order orchestration and planning data repository. Any records with errors or warnings are written to the data collections staging tables. For staging table uploads, you load the data from the extracted data files into the data collections staging tables, and then you use the Staging Tables Upload program to load the data from the staging tables into the data repository.



You determine which entities you collect from which source systems and at what frequency you need to collect the data for each entity. The data for different

entities can be collected at different frequencies. For example, supplies and demands change frequently, so collect data for them frequently. Routings and resources, are more static, so collect data for them less frequently.

Which data collection method you use for which entity depends upon the frequency of data changes as follows:

- Web service upload
Use for entities with frequent data changes.
- Staging tables upload
Use for entities with more static data.

Web Service Upload Method

Use the Web service upload method for entities that change frequently, such as supply and demand entities. You determine the frequency of collections for each entity. For certain entities, you may implement Web services to run every few minutes. For other entities, you may implement Web services to run hourly.

To implement and manage your Web service uploads, you must design and develop the processes and procedures to extract the data in the format needed by the data collection web services. For more information regarding the data collection Web services, refer to the Oracle Enterprise Repository. For additional technical details, see Oracle Fusion Order Promising Data Collection Staging Tables and Web Service Reference, document ID 1362065.1, on My Oracle Support at <https://support.oracle.com>.

Staging Tables Upload Method

Use the staging tables upload method for entities that do not change frequently, such as routings and resources. You determine the frequency of collections for each entity. You may establish staging table upload procedures to run daily for some entities, weekly for some entities, and monthly for other entities.

To implement and manage your staging table uploads, you must develop the processes and procedures you use to extract data from an external source system. You use Oracle Data Interchange, or another data load method, to load the extracted data into the data collection staging tables. For additional technical details, such as the table and column descriptions for the data collection staging tables, see Oracle Fusion Order Promising Data Collection Staging Tables and Web Service Reference, document ID 1362065.1, on My Oracle Support at <https://support.oracle.com>.

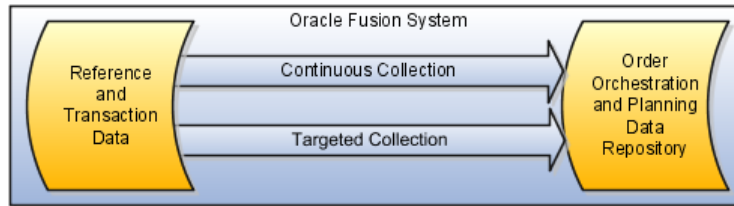
For the final step of the staging tables upload method, you initiate the Load Data from Staging Tables process from the Manage Data Collection Processes page or via the Enterprise Scheduling Service.

Data Collection Methods for the Oracle Fusion Source System: Explained

To populate the order orchestration and planning data repository with data collected from the Oracle Fusion source system, you use a combination of two data collection methods: continuous collection and targeted collection. You

typically use continuous collection for entities that change frequently and targeted collection for entities that are more static.

The following figure illustrates the two data collection methods, continuous collection and targeted collection, used in combination to collect data from the Oracle Fusion source system.



Continuous Collection

When you use the continuous collection method, you are only collecting incremental changes, and only for the entities you have included for continuous collection. Because continuous collection only collects incremental changes, you usually set up the continuous collection to run frequently, such as every five minutes.

Note

Prior to including an entity for continuous collection, you must have run at least one targeted collection for that entity.

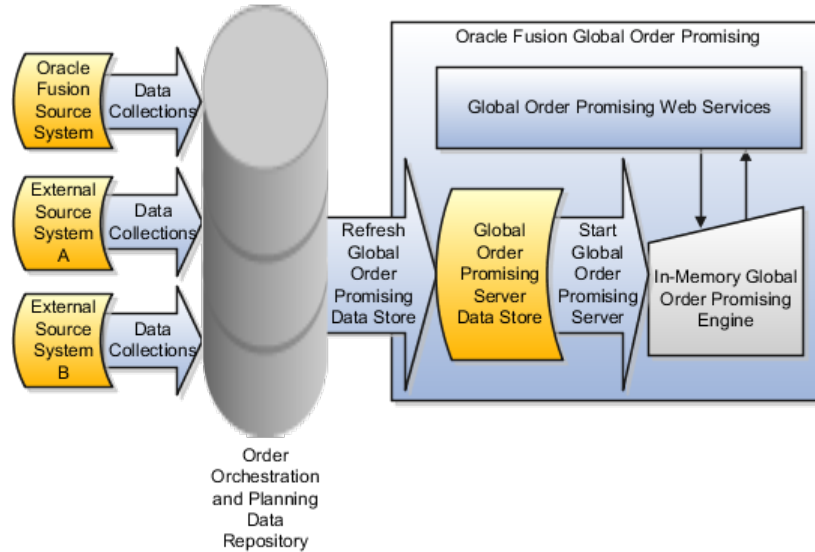
Targeted Collection

When you collect data using the targeted collection method, you specify which entities to include in the targeted collection. For the included entities, the data in the data repository that was previously collected from the Oracle Fusion source system is deleted and replaced with the newly collected data. The data for the entities not included in the targeted collection is unchanged. You typically use the targeted collection method to collect data from entities that do not change frequently.

Refreshing the Global Order Promising Engine: Explained

The Global Order Promising engine is an in-memory engine that uses an in-memory copy of the data collected into the order orchestration and planning data repository. To ensure the in-memory data reflects the latest supply and demand data collected into the data repository, you should refresh the Global Order Promising data store and start the Global Order Promising server at least once a day.

The following figure illustrates that you perform data collections to populate the order orchestration and planning data repository with current data from multiple source systems. After you complete a cycle of data collections, you refresh the Global Order Promising data store with the latest data from the data repository. After you refresh the Global Order Promising data store, you start the Global Order Promising server to load a copy of the refreshed data from the data store into main memory.



To refresh the in-memory copy of the collected data with the most recently collected data, perform these two steps:

1. Refresh the Global Order Promising data store.
2. Start the Global Order Promising server.

Refresh the Global Order Promising Data Store

To refresh the Global Order Promising data store, complete these steps:

1. Navigate to the Schedule New Process page by following this navigation path:
 - a. Navigator
 - b. Tools
 - c. Schedule Processes
 - d. Schedule New Process
 - e. Click the more link
2. Select the **Schedule Processes** link.
3. Click the **Submit New Request** button.
4. In the popup window, select Job for the type.
5. Search for and select the process named RefreshOpDatastore.
6. Select the entities you want to refresh and submit the job.

Start the Global Order Promising Server

To start the Global Order Promising server, you use an Oracle Fusion Global Order Promising instantiation of Oracle Enterprise Manager.

You do not need to stop the server before you start it. If the Global Order Promising server is already running when you start the Global Order Promising server, the Global Order Promising engine currently in memory continues to run until the start process is complete. The Start Global Order Promising Server process updates another engine with the current data from the Global Order

Promising Server data store. When the updated engine comes up, the existing engine with the old data is automatically shut down.

Important

The Current Date attribute stored within the Global Order Promising engine is also updated when you start the Global Order Promising server. If the Global Order Promising engine is not updated at least once a day, the Global Order Promising engine may have a wrong current date, and there may be issues with promising results.

Note

You also use an Oracle Fusion Global Order Promising instantiation of Oracle Enterprise Manager to monitor performance of the Global Order Promising server, to access log files, and to stop the server when necessary.

Manage Orchestration Data Collection Processes

Managing Data Collection Processes: Overview

For your data collections from the Oracle Fusion source system, you use the Manage Planning Data Collection Processes page or the Manage Orchestration Data Collection Processes page. From these pages you perform the following:

- Manage your continuous collections from the Oracle Fusion source system.
- Manage your collections destination server.
- Perform your targeted collections from the Oracle Fusion source system.

For your data collections from external source systems, most of the management of your Web services uploads and staging tables uploads is performed external to the Oracle Fusion application pages. If you choose to perform staging tables uploads, you initiate the Perform Data Load process from the Manage Planning Data Collection Processes page, from the Manage Orchestration Data Collection Processes page, or from the Oracle Fusion Enterprise Scheduler.

Continuous Collection Publish Process: Explained

To enable continuous collections, you must set up the publish data processes for the Oracle Fusion source system. The publish process performs the incremental data collections from the Oracle Fusion source system. You can start, stop, and pause the publish process. To review statistics regarding the publish process, view process statistics from the **Actions** menu on the Continuous Collection - Publish tab on the Manage Planning Data Collection Processes page or the Manage Orchestration Data Collection Processes page.

Note

Because continuous collections only collects net changes, you must perform at least one targeted collection for an entity before you include the entity for continuous collections.

Publish Process Parameters: Points to Consider

You define the publish process parameters to determine the frequency and scope of the continuous collections publish process.

You define the frequency and scope of continuous collections by specifying the following:

- Process Parameters
- Process Entities

Process Parameters

You determine how frequently the continuous collections publish process executes by specifying the frequency in minutes. The continuous collections publish process will publish incremental changes based on the frequency that was defined when the publish process was last started.

You determine which organizations will be included in the set of organizations for which data is collected by specifying an organization collection group. You can leave it blank if you want data collected from all organizations.

Process Entities

You determine which entities are collected during the continuous collections cycles by selecting which entities you want included in the collections. The continuous collections publish process collects incremental changes for the business entities that were included when the publish process was last started.

Collections Destination Server: Explained

The collections destination server is applicable to all four data collection methods. For the continuous collections method the collections server is the subscriber to the continuous collections publish process. From the Actions menu on the Collections Destination Server tab you can access a daily statistic report with statistics regarding each of the collection methods. You also can access a data collections summary report.

Destination Server Collections Parameters: Points to Consider

The collection parameters are initially set to what was defined for the Oracle Fusion system when your planning source systems or order orchestration source systems were initially managed. You can fine tune the parameters for your data collections.

Data Collection Parameters

The data collection parameters affect the usage of system resources. This table define what each parameter does and provides guidelines for setting it.

Parameter	What the Parameter Does	A Typical Value for the Parameter
Number of Database Connections	Defines the maximum number of database connections the source server can create during the collection process. This controls the throughput of data being extracted into the Source Java program.	10

Number of Parallel Workers	Defines the maximum number of parallel workers (Java threads) used to process the extracted data. The number here directly impacts the amount of central processing units and memory used during a collection cycle.	30
Cached Data Entries in Thousands	During data collections, various lookup and auxiliary data are cached in the collection server to support validation. For example, currency rate may be cached in memory. This parameter controls the maximum number of lookup entries cached per lookup to prevent the server from occupying too much memory.	10,000

Cross-Referencing Data During Data Collections: Explained

When you collect data from multiple source systems, you often collect a variety of values for the same instance of an entity. You cross-reference data during data collections to store a single, agreed value in the order orchestration and planning data repository for each instance of an entity.

Caution

Cross-referencing data during data collections can impact the performance of your collections. For collections from external source systems, consider performing the cross-references as part of your processes to extract data into data files.

The following information explains why you might need to cross-reference your data during data collections, and what you need to do to implement cross-referencing:

- Cross-reference example
- Cross-reference implementation

Cross-Reference Example

The following table provides an example of why you might need to cross-reference your data during data collections. In the example, the Kilogram unit of measure is collected from two source systems. The source systems use a different value to represent kilogram. You decide to store kg for the value for Kilogram in the order orchestration and planning repository.

Source System	Collections Entity	Source Value	Target Value
System A	Unit of measure	kilogram	kg
System B	Unit of measure	k.g.	kg

Cross-Reference Implementation

To implement cross-referencing, you must complete the following actions:

1. Decide which business object to enable cross reference.
2. For each object, work with business analyst to decide value-to-value maps.
3. Use the Oracle Fusion Middleware Domain Value Map user interface to upload mappings to the corresponding domain value map.
4. On the Manage Planning Data Collection Processes page, enable the corresponding entity for cross-reference.
5. Determine an ongoing procedure for adding new values into the domain value map.

For more information, see the articles regarding order promising or data collections on My Oracle Support at <https://support.oracle.com>.

FAQs for Manage Planning Data Collection Processes

Can I use continuous collection to collect item costs?

The continuous collection data collection method is partially supported for item costs. Item costs are collected in the next incremental collection cycle for previously existing items when one or more item organization attributes in addition to item cost have changed.

When a new item is defined, the item cost for the new item is not collected in the next incremental collection cycle. If an existing item is not changed other than an update to the item cost, the item cost change is not picked up in the next incremental collection cycle.

Tip

If items are added frequently, item costs are changed frequently, or both, then targeted collection of item costs should be routinely performed, perhaps once a day.

Perform Orchestration Data Collections

Loading Data into the Data Collections Staging Tables Using Oracle Data Integrator: Explained

To use the staging tables upload method, you must load the data you extract from your external source systems into the staging tables. You can use Oracle Data Integrator to load the extracted data into the staging tables.

If you have installed Oracle Data Integrator (ODI), and configured ODI for use by Oracle Fusion applications, you can load data to the staging tables by scheduling the Perform Data Load to Staging Tables process, PerformOdiSatagingLoad. To use this process, you must perform these steps and understand these details:

- Steps to use the Perform Data Load to Staging Tables process
- Steps to manually prepare and update the required dat files
- Details regarding the Perform Data Load to Staging Tables process
- Steps to verify execution status after starting the Perform Data Load to Staging Tables process

- Details regarding verifying the Perform Data Load to Staging Tables process execution status
- List of interface ODI scenarios run for each business entity

Steps to Use the Perform Data Load to Staging Tables Process

The Perform Data Load to Staging Tables process invokes an ODI data load. To use this process, follow these steps:

1. Create a data file for each business entity for which you are extracting data from your external source system. The file type for the data files must be dat. Use the sample dat files provided on My Oracle Support as templates. The data in the files you create must conform to the exact formats provided in the sample files.
 - To obtain the sample dat files, see Oracle Fusion Order Promising Data Collections Sample ODI Data Files, document ID 1361518.1, on My Oracle Support <https://support.oracle.com>.
 - You can open the sample dat files in a spreadsheet tool to review the sample data. The sample data shows the minimum required fields for an entity.
2. Place the dat files in the host where the Supply Chain Management (SCM) ODI agent is installed. The dat files must be placed at this specific location: /tmp/ODI_IN.
 - The path for this location is configured for the SCM ODI Agent. The SCM ODI Agent is an ODI software agent that services ODI related client requests. More information about this agent can be found in the ODI product documentation.
 - After ODI is installed, you must use the ODI console to refresh the variables C_LAST_UPDATED_BY and C_CREATED_BY.
3. Schedule the Perform Data Load to Staging Tables, PerformOdiStagingLoad, process.

Steps to Manually Prepare and Update the Required dat Files

You can develop data extract programs to extract data from your external source systems and store the extracted data into the required dat files in the required format. To manually add data to the dat files, follow these steps:

1. Open the applicable dat file in a spreadsheet tool. When you open the file, you will be prompted to specify the delimiter.

Use the tilde character, ~ , for the delimiter.

2. Add any data records you want to upload to the staging tables into the spreadsheet. Data for date type columns must be in the DD-MON-YY date format.
3. Save the worksheet from the spreadsheet tool into a text file.
4. Use a text editor and replace spaces between columns with the tilde character.
5. Verify that every line terminates with a CR and LF (ASCII 000A & 000D respectively.)

6. Upload the dat file to the /tmp/ODI_IN directory where the SCM ODI agent is running. The location is seeded in the ODI topology. Upload (FTP) the dat file in binary mode only.
7. Review the file in vi after the FTP upload to detect junk characters and, if any, remove them.

Details Regarding the Perform Data Load to Staging Tables Process

The Perform Data Load to Staging Tables process invokes the ODI scenario MASTER_PACKAGE that internally invokes all four projects defined in ODI for collections. Each of these four projects invokes various interfaces. Data is loaded from flat files to staging tables for all the business objects enabled for Oracle Fusion 11.1.2.0.0 through Oracle Data Integrator.

The following are specific details for the process:

- Process Name: PerformOdiStagingLoad
- Process Display Name: Perform Data Load to Staging Tables
- Process Description: Collects planning data from flat files and loads to staging tables using Oracle Data Integrator.
- ODI Project Name: SCM_BulkImport
- ODI scenario Name: MASTER_PACKAGE
- SCM Scheduler: SCM_ESS_ODI_SCHEDULER
- Agent URL: your_host_name:your_port_no/oracleodiagent (substitute your host name and your port number)

Steps to Verify Execution Status after Starting the Perform Data Load to Staging Tables Process

To verify the execution status after starting the Perform Data Load to Staging Tables process, perform these steps:

1. The Perform Data Load to Staging Tables process does not log messages to the scheduled processes side. To check for a log message, query the Request_History table using this select statement:

```
Select * from fusion_ora_ess.request_history where requestid=
<request_id>;
```

- Check the Status column for the overall execution status of the job and the Error_Warning_Detail column for a detailed error message, if any.
2. Check the ODI scenario execution status details in the ODI operator window. The scenario names are listed in the table in the List of Interface ODI Scenarios Run for Each Business Entity section of this document.
3. If log directories are accessible, check the following ODI logs for specific information on ODI scenario execution path:

```
/slot/emsYOUR_SLOT_NUMBER/appmgr/WLS/user_projects/domains/
wls_appYOUR_SLOT_NUMBER/servers/YOUR_ODI_SERVER_NAME/logs
```

- Diagnostic: for any errors in execution
- Server: for all the logs specific to ODI console

- Agent: for scenario entry and exit and for session ID

Details Regarding Verifying the Perform Data Load to Staging Tables Process Execution Status

When verifying the Perform Data Load to Staging Table process, remember the following:

- No logs will be written at the scheduled processes side. Also, the session id for ODI scenario cannot be found at the scheduled processes side.
- When viewing the process status on the Scheduled Processes page, a Success status does not mean that all the data got into the staging tables successfully. The Success status only indicates that the scenario is launched successfully. Scenario status must be checked from ODI logs.
- You cannot determine the refresh_number generated by ODI for the current process run from the Scheduled Processes page. To obtain the refresh number, you must use this query to query the msc_coll_cycle_status table and check for the ODI collection_channel:

```
Select * from msc_coll_cycle_status order by refresh_number desc;
```

List of Interface ODI Scenarios Run for Each BusinessEntity

One or more interface ODI scenarios are run for each business entity. Each interface scenario maps to one entity. If any interface Scenario fails in ODI, that entity data is not collected to the staging tables. This table lists the business entities and the interface ODI scenarios run within each business entity.

Business Entity	Interface ODI Scenarios
Work-in-Process Requirements	WIP_COMP_DEMANDS_SCEN WIP_OP_RESOURCE_SCEN
Calendars	CALENDAR_SCEN CALENDAR_WORKDAYS_SCEN CALENDARDATES_SCEN CALENDAR_EXCEPTIONS_SCEN CALENDARSHIFTS_SCEN CALENDAR_PERIODSTARTDAYS_SCEN CALENDAR_WEEKSTARTDAY_SCEN CALENDAR_ASSIGNMENTS_SCEN
Demand Classes	DEMAND_CLASS_SCEN
Global Supplier Capacities	GLOBAL_SUP_CAPACITIES_SCEN
Interorganization Shipment Methods	SHIPMENT_METHODS_SCEN
Item Cost	ITEM_COST_SCEN
Item Substitutes	ITEM_SUBSTITUTES_SCEN
Item Suppliers (Approved Supplier List)	ITEM_SUPPLIERS_SCEN
On Hand	ONHAND_SCEN

Organizations	ORGANIZATIONS_SCEN
Purchase Orders and Requisitions	SUPPLY_INTRANSIT_SCEN PO_IN_RECEIVING_SCEN PO_SCEN PR_SCEN
Planned Order Supplies	PLANNEDORDERSUP_SCEN
Resources	RESOURCES_SCEN RESOURCE_CHANGE_SCEN RESOURCE_SHIFTS_SCEN RESOURCE_AVAILABILITY_SCEN
Routings	ROUTING_OPERATION_RESOURCES_SCEN ROUTINGS_SCEN ROUTING_OPERATIONS_SCEN
Sourcing Rules	SOURCING_ASSIGNMENTS_SCEN SOURCING_RULES_SCEN SOURCING_ASSIGNMENTSETS_SCEN SOURCING_RECEIPT_ORGS_SCEN SOURCING_SOURCE_ORGS_SCEN
Subinventories	SUB_INVENTORIES_SCEN
Trading Partners	TRADING_PARTNERS_SCEN TRADING_PARTNER_SITES_SCEN
Units of Measure	UOM_SCEN UOM_CONVERSION_SCEN UOM_CLASS_CONVERSION_SCEN
Work Order Supplies	WORKORDER_SUPPLY_SCEN

Parameters for the Perform Data Load Process: Points to Consider

To perform a data load from the data collection staging tables, you invoke the Perform Data Load from Staging Tables process. When you invoke the process, you provide values for the parameters used by the process

Parameters for the Perform Data Load from Staging Tables Process

When you perform an upload from the staging tables, you specify values for a set of parameters for the Perform Data Load from Staging Tables process including specifying Yes or No for each of the entities you can load. For the parameters that are not just entities to select, the table below explains the name of each parameter, the options for the parameter values, and the effect of each option.

Parameter Name	Parameter Options and Option Effects
Source System	Select from a list of source systems.
Collection Type	<ul style="list-style-type: none"> • Net change Data in the data repository is updated with the data uploaded from the staging tables. <ul style="list-style-type: none"> • Existing records are updated. For example, on hand is updated with current quantity. • New records are added to the data repository. For example, New purchase orders are added to the data repository. • Targeted Existing data in the data repository is deleted and replaced with the data uploaded from the staging tables. For example, a targeted data load for purchase orders will replace all existing purchase order data with the purchase order data from the staging tables.
Group Identifier	Leave blank or select from the list of collection cycle identifiers. Leave blank to load all staging table data for the selected collection entities. Select a specific collection cycle identifier to load data for that collection cycle only.
Regenerate Calendar Dates	<ul style="list-style-type: none"> • Yes You loaded calendar patterns into the staging tables so you need the concurrent process to generate and store individual dates to run. • No You loaded individual dates into the staging tables so you do not need the concurrent process to generate and store individual dates to run.
Regenerate Resource Availability	<ul style="list-style-type: none"> • Yes You loaded resource availability patterns into the staging tables so you need the concurrent process to generate and store individual dates to run. • No You loaded individual dates into the staging tables so you do not need the concurrent process to generate and store individual dates to run.

The parameters presented for the Perform Data Load from Staging Tables process also include a yes-or-no parameter for each of the entities you can collect

using the staging tables upload method. If you select yes for all of the entities, the data collections will be performed in the sequence necessary to avoid errors caused by data references from one entity being loaded to another entity being loaded.

Important

If you do not select yes for all of the entities, you need to plan your load sequences to avoid errors that could occur because one of the entities being loaded is referring to data in another entity not yet loaded. For more information, see the articles regarding order promising or data collections on My Oracle Support at <https://support.oracle.com>.

Collections Cycle Identifier: Explained

The collection cycle identifier is a unique number that identifies a specific data collection cycle, or occurrence. One cycle of a data collection covers the time required to collect the set of entities specified to be collected for a specific data collection method. The collection cycle identifier is then used in statistics regarding data collections, such as the Data Collection Summary report. The collection cycle identifier is also used for a parameter in various processes related to data collections, such as the Purge Staging Tables process and the Perform Data Load process.

This topic explains the population of the collection cycle identifier when you use collecting data from external source systems as follows:

- Web Service Uploads and the Collection Cycle Identifier
- Staging Tables Uploads and the Collection Cycle Identifier

Web Service Uploads and the Collection Cycle Identifier

When you use the Web service upload data collection method, a collection cycle identifier is included as part of the collected data. You can then use the collection cycle identifier to review statistics regarding the Web service collections, or to search for error and warning records written to the data collection staging tables.

Staging Table Uploads and the Collection Cycle Identifier

If you use the Oracle Data Integrator tool to load your extracted data into the data collections staging tables, a collection cycle identifier is created for each load session. Each record loaded into the staging table during the load session will include the collection cycle identifier for that session.

If you populate the data collection staging tables using a method other than the Oracle Data Integrator tool, you must follow these steps to populate the collection cycle identifier.

1. Groupid is to be populated in column refresh_number of each data collections staging table. In one cycle of loading data into the staging tables, the column should be populated with same value. Get the group id value as follows:

```
SELECT ...NEXTVAL FROM DUAL;
```

2. After a cycle loading data into the data collections staging tables, insert a row as follows into table msc_cycle_status for that cycle as follows:

```

INSERT INTO MSC_COLL_CYCLE_STATUS
(INSTANCE_CODE, INSTANCE_ID, REFRESH_NUMBER, PROC_PHASE, STATUS,
COLLECTION_CHANNEL, COLLECTION_MODE, CREATED_BY, CREATION_DATE,
LAST_UPDATED_BY, LAST_UPDATE_DATE)
SELECT a.instance_code, a.instance_id, :b1, 'DONE', 'NORMAL',
'LOAD_INTERFACE', 'OTHER', 'USER', SYSTIMESTAMP, USER, SYSTIMESTAMP
FROM msc_apps_instances a
WHERE a.instance_code= :b2 ;
:b1 is instance_code for which data is loaded
:b2 is the groupid value populated in column refresh_number in all
interface tables for this cycle

```

Collecting Calendars and Resource Availability: Points to Consider

When you collect calendars and net resource availability from external source systems, you decide whether to collect patterns or individual dates. Order promising requires individual calendar dates and individual resource availability dates to be stored in the order orchestration and planning data repository. If you collect calendar patterns or resource shift patterns, you must invoke processes to populate the order orchestration and planning data repository with the individual dates used by order promising.

You invoke the necessary processes by specifying the applicable parameters when you run data collections. The processes generate the individual dates by using the collected patterns as input. The processes then populate the order orchestration and planning data repository with the individual calendar dates and the individual resource availability dates.

Calendar Collections

When you collect calendars from external source systems, you decide whether to collect calendar patterns or individual calendar dates. Both methods for collecting data from external source systems, Web service upload and staging tables upload, include choosing whether individual calendar dates must be generated as follows:

- The Web service to upload to calendars includes a parameter to run the Generate Calendar Dates process.

You control whether the process will run. If the parameter is set to yes, then after the Web service upload completes, the process will be launched to generate and store individual calendar dates.

- The parameters for the Perform Data Load from Staging Tables process also include a parameter to run the Generate Calendar Dates process.

You control whether the process will run. If the parameter is set to yes, then after the load from staging tables completes, the process will be launched to generate and store individual calendar dates.

- In both scenarios, calendar data is not available while the Generate Calendar Dates process is running.

When you collect calendars from the Oracle Fusion system, the Generate Calendar Dates process is run automatically.

Restriction

Only calendar strings that are exactly equal to seven days are allowed. Calendar strings with lengths other than seven are not collected. Only calendars with Cycle = 7 should be used.

Resource Availability Collections

When you collect net resource availability from external source systems, you decide whether to collect resource shift patterns or individual resource availability dates. Both methods for collecting data from external source systems, Web service upload and staging tables upload, include specifying whether individual resource availability dates must be generated as follows:

- The Web service to upload to net resource availability includes a parameter to run the Generate Resource Availability process.

You control whether the process will run. If the parameter is set to Yes, then after the Web service upload completes, the process will be launched to generate and store individual resource availability dates.

- The parameters for the Perform Data Load from Staging Tables process also include a parameter to run the Generate Resource Availability process.

You control whether the process will run. If the parameter is set to Yes, then after the load from staging tables completes, the process will be launched to generate and store individual resource availability dates.

- In both scenarios, new resource availability data is not available while the Generate Resource Availability process is running.

You cannot collect net resource availability from the Oracle Fusion source system.

Parameters for the Perform Data Collection Process: Points to Consider

To perform a targeted data collection from the Oracle Fusion system, you use the Perform Data Collection process. When you invoke the process, you provide values for the parameters used by the process.

The Perform Data Collection Process

When you perform a targeted collection, you specify the Oracle Fusion source system to be collected from and the organization collection group to collect for. When you invoke the process, the parameters also include each of the fourteen entities you can collect from the Oracle Fusion source system with yes or no for the parameter options. The table below explains the other two parameters.

Parameter Name	Parameter Options
Source System	The source system presented for selection is determined by what system has been defined as the Oracle Fusion source system when the manage source systems task was performed.

Organization Collection Group	The organization collection groups presented for selection are determined by what organization groups were defined when the manage source systems task was performed for the selected source system.
--------------------------------------	--

The parameters presented also include a yes-or-no parameter for each of the entities you can collect. If you select yes for all of the entities, the data collections will be performed in the sequence necessary to avoid errors caused by data references from one entity being loaded to another entity being loaded.

Important

If you do not select yes for all of your entities, you need to plan your load sequences to avoid errors that could occur because one of the entities being loaded is referring to data in another entity not yet loaded. For more information, see the articles regarding order promising or data collections on My Oracle Support at <https://support.oracle.com>.

Organization Collection Group: Explained

When you perform a targeted collection from the Oracle Fusion source system, you use an organization collection group to contain the collections processing to only the organizations with data that is needed for the order orchestration and planning data repository. Organization collection groups limit targeted collections from the Oracle Fusion source system to a specific set of organizations.

You perform the following actions for organization collection groups:

- Define an organization collection group.
- Use an organization collection group.

Define an Organization Collection Group

You define organization groups when managing source systems for the source system where the version equals Oracle Fusion. For each organization in the organization list for the Oracle Fusion source system, you can specify an organization group. You can specify the same organization group for many organizations.

Use an Organization Collection Group

You use an organization collection group when you perform a targeted collection from the Oracle Fusion source system and you want to contain the collections processing to a specific set of organizations. You specify which organization group to collect data from by selecting from the list of organization groups defined for the Oracle Fusion source system. Data will only be collected from the organizations in the organization group you specified.

For example, if only certain distribution centers in your Oracle Fusion source system are to be considered for shipments to your customers by the order promising and order orchestration processes, you could create a DC123

organization group and assign the applicable distribution centers to the DC123 organization group when managing source systems. When you perform a targeted collection for the Oracle Fusion source system, you could select DC123 for the organization collection group.

Review Orchestration Collected Data

Data Collections Daily Monitoring: Explained

When you manage the data collection processes, you use the Process Statistics report and the Data Collection Summary report to routinely monitor your collections. When error records are reported, you query the data staging tables for further details regarding the error records. You can also review most of your collected data using the review collected data pages.

The following information sources are available for you to monitor data collections:

- Process Statistics report
- Data Collection Summary report
- Review collected data pages
- Staging table queries

Process Statistics Report

You view the Process Statistics report to monitor summary of statistic for the daily collections activity for each of your source systems. This report is available on the **Actions** menu when managing data collection processes for either the continuous collection publish process or the collections destination server. The day starts at 00:00 based on the time zone of the collection server.

For the Oracle Fusion source system, statistics are provided for both the continuous collection and the targeted collection data collection methods. For each external source system, statistics are provided for the Web service upload and for the staging tables upload data collection methods. The following statistics are provided in the Process Statistics report:

- Number of collection cycles for the current day
- Average cycle time in seconds
- Average number of records
- Average number of data errors

Note

The process statistics provide summary information, and are not intended for detailed analysis of the collections steps. Use the Oracle Enterprise Scheduler Service log files for detailed analysis.

Data Collection Summaries

You view the Data Collection Summary report to monitor statistics regarding the data collection cycles for each of your source systems. The summary report

shows last the results of the last 20 cycles of all collection types. This report is available on the Action menu when managing data collection processes for the collections destination server.

The Data Collection Summary report provides information for each source system. If a source system was not subject to a data collection cycle for the period covered by the summary, an entry in the report states that there are no cycles in the cycle history for that source system. For each source system that was subject to a data collection cycle for the period covered by the summary, the following information is provided for each data collection method and collected entity value combination:

- The data collection method
- The collection cycle number
- The entity collected and, for that entity, the number of records collected, the number of records with data errors, and collection duration
- Time started
- Time ended

Review Collected Data Pages

You can review most of your collected data by using the Review Planning Collected Data page or the Review Order Orchestration Collected Data page. Both pages include a list of entities from which you select to specify the entity for which you want to review collected data. The list of entities is the same on both pages. Most of the entities listed on the review collected data pages are identical to the entities you select from when you run collections, but there are a few differences.

Some of the entities on the list of entities you select from when you review collected data are a combination or a decomposition of the entities you select from when you run collections. For example, the Currencies data collection entity is decomposed into the Currencies entity and the Currency Conversions entity on the review collected data pages. For another example, the Supplies entity on the review collected data pages is a combination of data collection entities including the On Hand entity and the Purchase Orders and Requisitions entity.

A few of the data collection entities cannot be reviewed from the review collected data pages. The data collection entities that are not available for review on the review collected data pages are Resources, Resource Availability, Routings, Work-in-Process Resource Requirements, and Customer Item Relationships.

Staging Table Queries

If errors or warnings have been encountered during data collections, you can submit queries against the staging tables to examine the applicable records. For more information regarding the staging tables and staging table columns, see the articles regarding order promising or data collections on My Oracle Support at <https://support.oracle.com>.

Errors and Warnings When Collecting Data from External Source Systems: How They Are Handled

When you are collecting data from external source systems, the data collection processes perform many data validation checks. If the data validations fail with errors or warnings, the steps taken by the data collection processes vary slightly depending upon whether the Web service upload data collection method or the staging tables upload data collection method is used.

In both cases, records where errors are found are not loaded into the order orchestration and planning data repository. Instead records are loaded into, or remain in, the applicable staging tables with an appropriate error message. Records where only warnings are found are loaded to the data repository, and records are loaded into, or remain in, the applicable staging tables with an appropriate warning message.

Settings That Affect Error Handling When Collecting Data from External Source Systems

The handling of errors and warnings encountered when the data collection processes validate data during collections from external source systems depends upon which data collection method is used, Web service upload or staging tables upload.

How Errors and Warnings Are Handled

When you are running data collections using the Web services method, the following error and warning handling steps occur:

- **Errors:** Records are loaded to the applicable staging tables instead of the data repository and are marked with the appropriate error message.

A record with an error due to mandatory missing mandatory fields, such as organization or supplier or item, is first marked as retry. After several unsuccessful retry attempts, the record will be marked as error.

- **Warnings:** Records are loaded into the data repository and into the applicable staging tables with the appropriate warning message.

When you are running data collections using the staging tables upload method, the following error and warning handling steps occur:

- **Errors:** Records remain in the staging tables without being loaded to the data repository and are marked with the appropriate error message.

A record with an error due to mandatory missing mandatory fields, such as organization or supplier or item, is first marked as retry. After several unsuccessful retry attempts, the record will be marked as error.

- **Warnings:** Records are loaded into the data repository and remain in the staging tables with the appropriate warning message.

Error Handling Example

When a Planned Order Supplies record is collected, many validations occur for which an error is recorded if the validation fails.

For example, the supplier name is validated against the suppliers data in the order orchestration and planning data repository. If the supplier name is not found, the validation fails with an error condition, and the following steps occur:

- The Planned Order Supplies record is not loaded into the data repository.
- The Planned Order Supplies record is loaded into the applicable staging table, or remains in the applicable staging table, with an error message stating invalid supplier or invalid supplier site.

Warning Handling Example

When a Planned Order Supplies record is collected, many validations occur for which a warning is recorded if the validation fails.

For example, the Firm-Planned-Type value in the record is validated to verify that the value is either 1 for firm or 2 for not firm. If the validation fails, the failure is handled as a warning, and the following steps occur:

- The Planned Order Supplies record is loaded into the data repository with the Firm-Planned-Type value defaulted to 2 for not firm.
- The Planned Order Supplies record is also loaded into the applicable staging table, or remains in the applicable staging table, with a warning message stating invalid firm planned type.

Purge Collected Data Processes: Points to Consider

You use the Purge Data Repository Tables process to delete all collected data from the order orchestration and planning data repository that was collected from a specific source system. You use the Purge Staging Tables process to remove data that you no longer need in the data collections staging tables.

The Purge Data Repository Tables Process

You use the Purge Data Repository process to delete all data for a source system from the order orchestration and planning data repository. The process enables you to delete data for a specific source system. You typically use the Purge Data Repository process when one of your source systems becomes obsolete, or when you decide to do a complete data refresh for a set of collection entities.

The Purge Data Repository process has only two parameters, both of which are mandatory. This table explains the two parameters.

Parameter Name	Parameter Options
Source System	Select a source system for the list of source systems. All data for the selected system will be deleted from the data repository.
Purge Global Entities	Yes or No If you select yes, in addition to the applicable data being deleted for the source-specific entities, all data from global entities will also be deleted. If you select no, data will be deleted from the source-specific entities only.

The Purge Staging Tables Process

You use the Purge Staging Tables process to delete data from the data collection staging tables.

The following table explains the parameters you specify when you run the Purge Staging Tables process. In addition to the five parameters explained below, you specify yes or no for each of the twenty-five data collection entities.

Parameter Name	Parameter Options
Source System	Select a source system for the list of source systems. Data will be deleted for this source system only.
Record Type	The record type specifies which type of records to purge as follows: <ul style="list-style-type: none"> • Error Purge only error records. • Warning Purge only warning records. • Retry Purge only records marked as retry. • Complete Purge only records that have been successfully processed and data stored in the data repository. • All Purge all records.
Collection Cycle ID	Specify a value for the collection cycle identifier to purge data for a specific collection cycle only, or leave blank.
From Date Collected	Specify a date to purge data from that date only, or leave blank.
To Date Collected	Specify a date to purge data up to that date only, or leave blank.

FAQs for Review Planning Collected Data

What's an order orchestration reference object?

One of the objects in the set of objects used by the orchestration processes to determine the meaning and descriptions for names or codes, such as payment terms names, freight-on-board codes, and mode-of-transport codes.

The sales order data passed to the orchestration processes contains the names or codes, but the processes need to display the meanings or descriptions. The

data to determine the meanings or descriptions for the names or codes must be collected into the order orchestration and planning data repository.

For example, sales order information is passed to the Order Orchestration processes containing a freight-on-board code equal to 65, and the order orchestration and planning data repository contains a record with freight-on-board code equal to 65. The processes use the matching codes to determine that the freight-on-board code meaning is equal to Origin, and the description is equal to Vendors responsibility.

Tip

For the full list of order orchestration reference objects, review collected data for the order orchestration reference objects, and view the list of values for the **Lookup Type** field.

Define Orchestration

Define Orchestration: Overview

Oracle Fusion Distributed Order Orchestration automates order orchestration across fulfillment systems using highly adaptable, flexible business processes. The following setups are required for orchestration:

- **Task type:** Mechanism used to group fulfillment tasks together. Each task type contains a selection of services that communicate with a specific type of fulfillment system, for example, a billing system.
- **Orchestration process definition:** Business process required to fulfill a fulfillment line. It includes the sequence of task layer service calls, as well as planning details, change management parameters, and status conditions.
- **Change logic:** Set of rules that control how changes to booked orders are handled by the orchestration process.
- **Process planning:** Schedule that shows the completion date of each task and of the orchestration process itself.
- **Jeopardy threshold and priority:** Mechanism for indicating late-running tasks at runtime.
- **Statuses and status conditions:** Indicators of the statuses of order and process objects, including task, orchestration process, fulfillment line, orchestration order line, and orchestration order. Status conditions are the rules that determine when any of these order and process objects reaches a particular status.

Manage External Interfaces

Manage External Interfaces: Overview

The external interface layer is the functional component within Oracle Fusion Distributed Order Orchestration that manages the communication between

Distributed Order Orchestration and external fulfillment systems. Its primary functions are routing the fulfillment request and transforming the data.

The external interface layer enables loose coupling between Distributed Order Orchestration and fulfillment systems:

- Abstracts external systems from the orchestration process definition to minimize changes when adding new order capture or fulfillment systems.
- Provides an extensible, SOA-enabled framework for flexible integration to external applications.
- Provides a complete, open, and integrated solution that lowers total cost of ownership.

When the setup is done, Distributed Order Orchestration can connect to any fulfillment system.

Template Task Layer Mandatory Setup: Explained

Some setup is required to use the template task layer. Some of these setup activities are mandatory because processing cannot occur without the setup information. Other setup activities are optional and will depend on the desired behavior of the services that are associated with the new task type that you are creating. You may set up as many different uses of the template task layer as you need.

The following setup steps are mandatory:

- Create a custom task type.
- Assign status codes to the task type.
- Create the connector.
- Register the connector Web service.
- Use the task type in orchestration process definitions.

Create a Custom Task Type

Create a custom task type on the Manage Task Types page. When you create a custom task type, two services are created, one that corresponds to the Create (outbound) operation code and the other that corresponds to the Inbound operation code. You can specify names for these two services, and you can add services that correspond to the other available operation codes (Cancel, Change, Get Status, Apply Hold, and Release Hold). Create at least one task for each new task type.

Assign a Status Code to the Task Type

Assign status codes to each custom task type. A few system status codes are defaulted, for example, Canceled, Change Pending, Cancel and Pending. The status codes that are associated with each task type also control the values for exit criteria for wait steps that use this task type and for the value of the next expected task status in the orchestration process step definition. You can create new status codes, or you can assign existing status codes to the new custom task type.

Create the Connector

Create the connector that integrates Distributed Order Orchestration with the fulfillment system that will perform the tasks and services of the new task type.

Register the Connector Web Service

Register the Web service that is required for integration to the fulfillment system that will perform the tasks and services of the new task type.

Use the Task Type in Orchestration Process Definitions

You use the new task type and the tasks and services within it by building them into an orchestration process definition, just as you would with the predefined task types, tasks and services. Because splits are allowed for these services, the services may be used only in one branch that is then defined as the single branch that allows services that can be split.

Template Task Layer Optional Setup: Explained

Some setup is required to use the template task layer. Some of these setup activities are mandatory because processing cannot occur without the setup information. Other setup activities are optional and will depend on the desired behavior of the services that are associated with the new task type that you are creating. You may set up as many different uses of the template task layer as you need.

The following setups are optional:

- **Preprocessing service:** You can add preprocessing logic to the actions built into the template task layer service. Your organization might want to add preprocessing logic that defaults data onto the outbound request or validates it.
- **Postprocessing service:** You can add postprocessing logic to the actions that are built into the template task layer service. Your organization might want to add postprocessing logic that defaults logic onto the inbound request, validates the inbound request, or interprets any attributes or messages that are returned by the fulfillment system that might indicate the need for split processing.
- **Change management:** If you want change management processing for orchestration process steps that use one of these task types, then specify the attributes for the task type on the Manage Order Attributes That Identify Change pages. You also need to be sure there are Update and Cancel services and their associated connectors.
- **Define hold codes:** If you want to apply holds to the new services, then create hold codes for them. Hold All applies to the new services, as well as to the existing ones.
- **Define jeopardy thresholds:** If you want to compute specific jeopardy scores on the new custom tasks, then define jeopardy thresholds for them.
- **Define processing constraints:** If you know of some circumstances under which a new custom task should not be called, then define processing constraints to control the behavior.

- Define preprocessing validation of mandatory attributes: You may want to use processing constraints to declare which attributes in the outbound request payload are mandatory.
- Define postprocessing validation of mandatory attributes: You may want to use processing constraints to declare which attributes in the inbound response payload are mandatory.
- Define data set used as part of outbound request: By default, the template task layer uses a complete data structure to communicate Oracle Fusion Distributed Order Orchestration attributes between GetValidFLineData, the preprocessing service, and the external interface layer routing rules. You can trim the data set to a functionally appropriate size to make processing more efficient.
- Register error messages: If your external fulfillment systems send Distributed Order Orchestration error messages that you want to process and display in the Distributed Order Orchestration messaging framework, then you must register these messages.

Sales Order Attachments: Explained

A sales representative may add attachments while creating a sales order. An attachment might be a document with requirements for manufacturing, a memo for price negotiation, or a URL for product assembling instructions, to name just a few possibilities. Oracle Fusion Distributed Order Orchestration accepts the attachments as part of the sales order. You can view attachments in the Order Orchestration work area and subsequently send them to the necessary fulfillment system. Attachments cannot be sent from the fulfillment system to Distributed Order Orchestration or from Distributed Order Orchestration to an order capture system.

Attachment Configuration: Explained

Sales order attachments can be transmitted from the order capture system to Oracle Fusion Distributed Order Orchestration and from Distributed Order Orchestration to fulfillment systems. To enable transmission of sales order attachments to Distributed Order Orchestration, you must collect the document category during orchestration data collection. To enable transmission from Distributed Order Orchestration, you must invoke the AttachmentsAM public service. Use this service to select and send attachments to the designated fulfillment system, based on the type of the fulfillment request and the category of the attachment.

Web Service Setup: Explained

Web services are used to integrate fulfillment applications with Oracle Fusion Distributed Order Orchestration. Distributed Order Orchestration has a Web service broker that routes requests from the fulfillment task layer to one or more fulfillment systems and vice versa. The following explains how Web services are set up.

1. Create the connector.

2. Deploy the connector.
3. Register the connector.
4. Create external interface routing rules.

Create the Connector

Define an XSLT transformation file to transform the Distributed Order Orchestration fulfillment task message to a Web service-specific message. You can use the Oracle JDeveloper mapper tool or any other tool of choice. Similarly, define the XSLT transformation file to transform the response from the Web service to a message specific to Distributed Order Orchestration.

Deploy the Connector

Make a copy of the connector template, and replace the XSLT transformation files with the files you created for the connector.

Register the Connector

Register the connector on the Manage Web Services page. You must create the source system, so that it is available for selection from this page.

Create External Interface Routing Rules

Create external interface routing rules on the Manage External Interface Routing Rules page. These are the business rules that determine to which fulfillment system requests are routed.

User Credential Key: Explained

The user credential key is a user and password combination created in the credential stores, or Oracle Wallet Manager. This key provides for secure authenticated communication between Oracle Fusion Distributed Order Orchestration and fulfillment systems. You must create a user credential key to integrate Oracle Fusion Distributed Order Orchestration with external services.

Follow the instructions for adding a key to a credential map in the Oracle Fusion Middleware Security Guide 11g Release 1 (11.1.1). You must have the administration privilege and administrator role. In the Select Map list, select oracle.wsm.security. Enter the key, user name, and password from the service that is being integrated with Oracle Fusion Distributed Order Orchestration. Register the user credential key on the Manage Web Service Details page of Distributed Order Orchestration.

Web Service Invocation from Oracle Fusion Distributed Order Orchestration: Explained

To invoke external Web services from Oracle Fusion Distributed Order Orchestration, you must ensure that the user credential is valid in the target system and the security certificate to encrypt and decrypt messages.

User Credential

Obtain a user credential key, and add it to the invoking server's identity store.

A user credential is a user name and password defined in the target system and is used for authenticating incoming requests. This means that the consumer of the service must pass in these credentials as part of the request.

Ask the service provider for the user credentials to be used when invoking their service. The IT administrator must add the user credentials provided by the service provider to the service consumer's server and provide a reference, which is called a CSF-KEY.

Register the external system in the Manage Source System Entities flow. For each service hosted on the external system that you plan to interact with, register the service on the Manage Web Service Details page. Provide a name (Connector Name) for the service, the physical location (URL) of the service; and the CSF-KEY pointing to the user credential that will be used when interacting with the external service. This key applies to all services offered by the target system.

Security Certificate

Oracle recommends that you configure servers that are running external Web services that must be invoked to advertise the security certificate in the WSDL. The default setting in Oracle WebLogic Server is to advertise the security certificates. Check whether your servers support this feature; if so, then enable the feature.

If you cannot set up the server this way, then use the keystore recipient alias. Ask the service provider for the security certificate. An IT administrator imports the target server security certificate into the invoking server and provides a reference, which is called a keystore recipient alias. Add this alias to the external service entry that was created when you specified the user credential. Register this keystore recipient alias on the Manage Web Service Details page against the records created for that system. This key applies to all services offered by the target system.

If the other options do not work, then configure the target servers to use the Oracle security certificate. Import this certificate into your target servers. No setup is required on the invoking server for the security certificate.

External Interface Routing Rules: Explained

Use external interface routing rules to determine to which fulfillment system a fulfillment request must be routed. You can use order, fulfillment line, and process definition attributes to select the fulfillment system connectors. The rules are executed in Oracle Business Rules engine.

Creating External Interface Routing Rules: Examples

Use these scenarios to understand how to use external interface routing rules.

Task Type Determines Routing of Request

You want orchestration orders that are ready to be shipped to go to the shipping fulfillment system. You write an external interface routing rule that requires that if the task type code of an orchestration order is Shipment, then route the request to the ABCShippingSystem connector.

Customer Attribute Determines Routing of Request

Your company has two invoicing systems. When it is time to send out an invoice, you want Widget Company always to be invoiced by system ABC. You write an external interface routing rule that requires that if the customer is Widget Company and the task type code is Invoice, then route the request to ABCInvoicingSystem.

Define Orchestration Processes

Orchestration Process Definitions: Explained

An orchestration process is a process flow that approximates your organization's fulfillment process. An orchestration process contains a sequence of steps that takes a fulfillment line through the fulfillment process. An orchestration process contains the instructions for how to process an order, such as which steps and services to use, step dependencies, conditional branching, lead-time information, how to handle change orders, which status values to use, and more. You define orchestration processes according to your organization's needs. You create rules, so that at run time the appropriate orchestration process is automatically created and assigned to the new fulfillment lines.

If you want to use the ShipOrderGenericProcess and ReturnOrderGenericProcess predefined orchestration processes, then you must generate them and deploy them; it is not necessary to release them.

Defining Orchestration Processes: Examples

Orchestration process definitions include the sequence of task layer service calls, as well as planning details, change management parameters, and status conditions. Use the following examples to understand how you can use orchestration process definitions to model your business processes.

Sequence of Task Layer Service Calls

You are an order administrator at a company that sells widgets. You list the logical steps that are required to fulfill an order. Then you create an orchestration process that mirrors your business process.

The first steps are:

- Step 1: Plant Acknowledgement
- Step 2: Assemble
- Step 3: Wait for Assemble COMPLETE
- Step 4: Ship
- Step 5: Wait for SHIPPED

(Statuses are represented in all uppercase letters.)

Your company requires that a representative call the customer when an invoice exceeds \$100,000. You continue creating the orchestration process by adding a conditional step (which is not a task layer service call):

- Step 6: Conditional node where the orchestration process branches

Afterwards, the steps continue as follows:

- Step 7: Call Customer
- Step 8: Wait for Call Customer COMPLETE
- Step 9: Send High Value Invoice
- Step 10: Wait for High Value Invoice BILLED

Otherwise, the steps are:

- Step 7: Invoice
- Step 8: Wait for BILLED

A merge node ends the branch.

Planning Details

You are an order administrator at a company that sells carpet. Your company has established lead times that enable representatives to monitor the fulfillment process and determine when orders will be fulfilled.

- Schedule: 2 days
- Reservation: 1 day
- Shipment: 6 days
- Invoice: 2 days

You create an orchestration process that contains this information by adding the default lead time to each orchestration process step. When a step is delayed, a background process automatically triggers replanning and expected completion dates are reset.

Status Definitions

You are an order administrator at a company that sells carpet. You have an important customer who requires that you notify the receiving clerk one day before the carpet is shipped. You create an orchestration process for this customer's orders. You use the Carpet Process orchestration process class, which contains the statuses: SHIPPED, RESERVED, READY TO SHIP, SHIPPED, INVOICED. On the Orchestration Process Status tab, you create status conditions for the orchestration process for the special customer, such as: If the status of the CreateShipment step is PRESHIP READY, then use the READY TO SHIP status to represent the status of the orchestration process. Now, the order manager can see in the Order Orchestration work area when the orchestration process status is READY TO SHIP.

Task Types: Explained

A task type is a grouping of services that perform fulfillment tasks. Task types represent common business functions that are required to process an order from its receipt from the order capture application to its execution in the fulfillment application. The following task types are provided by default: Schedule,

Reservation, Shipment, Activity, Invoice, Return. You can create additional task types by using the Custom and Activity task types. Task types are made up of services and tasks. Service refers to an internal Web service that communicates with the task layer. A task is a representation of the services of a task type. Tasks and services are the building blocks of an orchestration process.

Seeded task types are read-only. You cannot delete task types. You can change the names of task types you create, but it is not recommended.

You can edit the service names of the Activity and Custom task types. You can add services from the pool of available services, but you cannot edit or delete services for custom task types.

Use tasks to represent the services that belong to a task type. For example, you can define a Ship Goods task to represent services from the Shipment task type. When one of the Shipment services is running, Ship Goods appears in the Order Orchestration work area, regardless of whether the Create Shipment or Update Shipment service is called; the services do not appear in the Order Orchestration work area. You can define several tasks for a task type to represent your real-world user tasks, such as ShipGoods or ShipWidgets. Both tasks and services appear in the orchestration process definition.

Task Type Management

Task type management is the registration of internal service references for the task layer.

Change Processing: How It Processes Changes that Occur During Order Fulfillment

During fulfillment of an order, changes can originate from a variety of sources, such as from the customer through the order capture application or by the order manager in the Order Orchestration work area. Oracle Fusion Distributed Order Orchestration processes changes automatically, but you can influence some aspects of change processing through some of the setup options.

Settings That Affect Change Processing

Change processing occurs according to certain settings. The following parameters are set at the orchestration process level:

- **Order attributes that identify change:** Attributes that, when changed, require compensation of the orchestration process. Changes to certain attributes do not always require compensation of the orchestration process. For example, an addition of a suite number to a customer's address might not require adjustments to the orchestration process. You can refine how a change is processed for one of these attributes through compensation patterns for orchestration process steps.
- **Processing constraints:** Rules that control attempted changes to an order, for example, what can be changed, when, and by whom. Changes that are forbidden by processing constraints are disallowed, and an error message appears.
- **Change mode:** Setting that determines how frequently state snapshots are taken during an orchestration process. Snapshots are compared during

compensation of the orchestration process. For example, if you select Simple, then only the beginning and end states are compared. If you select None, then all changes to the orchestration process are rejected.

- Cost of change rule: Rule that specifies the costs to the business due to a requested change.

The following parameters are set at the orchestration process step level:

- Hold on wait: Directs the runtime engine to request a hold in an external system for a wait service task.
- Use dynamic attributes: Indicates whether the product-specific dynamic attributes need to be considered for change.
- Use flexfield attributes: Indicates whether flexfield attributes need to be considered for change.
- Compensation pattern: Rule that specifies the adjustments to make when an order is changed. For example, if Distributed Order Orchestration receives a change order with a new warehouse for the Create Shipment step, then Distributed Order Orchestration runs the Cancel service and Create service again. If a compensation pattern is not designated for a process step, then the default compensation pattern is used, which is to run the Update service (or the Cancel and Create services).

How Changes Are Processed

When a change order is received from an order capture system, Distributed Order Orchestration performs a lookup to determine whether the order key has been received before. Distributed Order Orchestration sends a request to the fulfillment system that is responsible for the task that was running when the change order was received. The request has several components: Hold current processing, designate whether a change can be accepted, and send the current status.

Change orders are decomposed and orchestrated in the same manner as new orders. If rules were set up for special processing of change orders, then the rules are applied at this time.

Distributed Order Orchestration checks for header-level processing constraints that prevent change processing. If change processing is allowed, then the delta service is called. The delta service checks the attributes that indicate whether the change must be compensated. If the change requires compensation, then compensation begins after line-level attributes are checked.

Distributed Order Orchestration checks line-level processing constraints. If constraints are violated for even one fulfillment line, then the entire change order is rejected.

When an action on the Order Orchestration work area requires change processing, all the above actions occur except decomposition. After the changes are identified by the delta service, Distributed Order Orchestration analyzes and compensates the process, step-by-step, analyzing the state of each step to determine what processing changes are needed to incorporate the changes to the order. To determine the steps to compensate, Distributed Order Orchestration uses the process step state snapshots taken at each task layer service invocation while the orchestration process was running.

The process delta service identifies all orchestration process steps that are associated with delta attributes. You can opt for the default behavior (context-based undo or update) or specify a business rule that determines the appropriate action as the compensation pattern for each process step. Distributed Order Orchestration evaluates the compensation pattern identified for the step to identify what processing to run in a change scenario. Compensation patterns include undo, redo, update, cancel, and none. The default compensation sequence is first in, first out, based on the orchestration process sequence. If the entire order is canceled, then a last in, first out sequence is used. After the compensating services are completed, processing continues using the original orchestration process specification or the appropriate orchestration process dictated by the changes is started. Expected dates are replanned, unless the entire order is canceled. At this point, change processing is completed.

Order Attributes That Identify Change: Points to Consider

Order attributes that identify change are attributes that, when changed by the order capture application or Order Orchestration work area user, require compensation of an orchestration process step. A change to any one of these attributes requires compensation of a step if the attribute is assigned to the task type associated with the step. For example, if the quantity of a sales order is increased, then additional supply must be scheduled and shipped. The Schedule and Shipment steps of the orchestration process are affected because the quantity is an attribute assigned to those task types.

Attribute Selection

Select an attribute from the list of entities: Fulfillment line, orchestration order line, or orchestration order. Selection of this attribute means that at run time, when a change order is received, the application searches for this attribute on the entity that you associated it with to determine whether it was changed on the change order. For example, if you select Scheduled Ship Date on the orchestration order line, then when a change order is received, the application compares the Scheduled Ship Date attribute on the line of the change order with the Scheduled Ship Date attribute of the most recent version of the orchestration order line.

By default, the application searches for a set of attributes, which are indicated by selected check boxes. You cannot deselect them; you can only add more attributes.

If you want flexfield attributes or dynamic attributes associated with specific products to be considered for change, then select Use Flexfield Attributes and Use Dynamic Attributes on the orchestration process definition. You cannot select these attributes individually.

Task Type Selection

The task type selection defines the attributes that will be used to evaluate whether a step using that task type requires compensation. Attributes are predefined for predefined task types, but additional attributes can be added. When you add a new task type, no attributes are defaulted. The task is not evaluated to determine compensation requirements unless you set up these attributes first.

Click **Add All** to add attributes to all existing task types.

Status Values: Explained

Status denotes the progress of an order from beginning to completion. The status of an orchestration order is determined by the status of its fulfillment lines, orchestration processes, and tasks. Status values appear in the Order Orchestration work area, where order managers can use them to quickly understand the progress of an orchestration order or its components.

You create a list of all the statuses that can be used in Oracle Fusion Distributed Order Orchestration. For each status code, you create a display name, which is how the status will appear in the Order Orchestration work area. Then, using the list of defined statuses, create a separate list of statuses that an administrator is allowed to use for each of the following: Fulfillment lines, task types, and orchestration processes. When administrators create status conditions for orchestration processes, they can choose from these status values only. You must define the status values in the Manage Status Values page to make them available for selection when creating status conditions.

Fulfillment Line Status: How It Is Determined

During processing of a fulfillment line, the tasks of the assigned orchestration process are fulfilled step by step. You can determine the status that will be assigned to a fulfillment line at each stage of the process. For example, you can indicate that if a Schedule Carpet Installation task has a status of Pending Scheduling, then the fulfillment line status will be Unscheduled.

Settings That Affect Fulfillment Line Status

You can designate the statuses that represent a fulfillment line when you define an orchestration process. These statuses are used to represent the status of the fulfillment line throughout the application. You create a status rule set that lists a sequence of the statuses that will be attained during the orchestration process and the conditions under which they are assigned to the fulfillment line. For example, you could designate the status Scheduled to be used for the fulfillment line when the Schedule Carpet task reaches a status of Completed.

How Fulfillment Line Status Is Determined

At run time, the application evaluates each of the status conditions sequentially. The true condition with the highest sequence number determines the status of the fulfillment line.

Split Priority: Explained

When an orchestration process splits, two or more instances of the same task result. Split priority is a ranking that is used to evaluate multiple instances of a task that splits. The ranking determines which task status represents the status of the orchestration process. A lower number corresponds to a higher rank. The status with the lower number is used to represent the status of the orchestration process.

For example, an orchestration process splits and results in two instances of the Schedule task. One Schedule task has a status of complete, and the other has a status of pending. Because pending has a split priority of two and complete has a split priority of three, pending is used to represent the status of the orchestration process.

Orchestration Order Status: How It Is Determined

An orchestration order can have one or more orchestration order lines, each in its own status. The status of the orchestration order is based on the orchestration order lines that are mapped to it

How Orchestration Order Status Is Determined

The following table shows how the orchestration order status is determined, given the statuses of the associated orchestration order lines.

Orchestration Order Line Statuses	Orchestration Order Status
All orchestration order lines are completed.	Closed
All orchestration order lines are not completed.	Open
Some, but not all, orchestration order lines are completed.	Partial
All orchestration order lines are canceled.	Canceled
Some orchestration order lines are canceled.	Ignore canceled orchestration order lines, and determine status based on the open orchestration order lines.

For example, if no orchestration order lines are completed, then the orchestration order status is open.

Orchestration Order Line Status: How It Is Determined

An orchestration order line can have one or more fulfillment lines, each with its own status. The status of the orchestration order line is based on the fulfillment lines that are mapped to it.

How Orchestration Order Line Status Is Determined

The following table shows how the orchestration order line status is determined, given the statuses of the associated fulfillment lines.

Fulfillment Line Statuses	Orchestration Order Line Status
All fulfillment lines are completed.	Closed
All fulfillment lines are not completed.	Open
Some, but not all, fulfillment lines are completed.	Partial
All fulfillment lines are canceled.	Canceled
Some fulfillment lines are canceled.	Ignore canceled fulfillment lines, and determine status based on the open fulfillment lines.

Jeopardy Priorities: Explained

Jeopardy priority indicates the level of risk associated with the delay of a task of an orchestration process. It appears in the Order Orchestration work area as Low, Medium, and High.

Create a jeopardy priority by mapping a jeopardy score range to one of the three severity levels. For example, you could map the jeopardy priority Low to a minimum jeopardy score of 0 and a maximum jeopardy score of 100. Jeopardy priorities are provided by default. You can change the values in the ranges to meet your business needs. You cannot delete or add priorities, or change jeopardy priority names; only Low, Medium, and High are available.

Jeopardy Score: Explained

Jeopardy score is a numerical ranking associated with a delay in the completion of a task of an orchestration process. Jeopardy score indicates how severe a delay is deemed. The jeopardy score is mapped to jeopardy priorities of Low, Medium, and High, which appear in the Order Orchestration work area. The indicator provides a quick visual cue to order managers, so that they can take appropriate action to mitigate a delay.

You determine jeopardy score when you create jeopardy thresholds.

Assignment to Tasks

Jeopardy score is assigned to tasks based on jeopardy thresholds. When a task is delayed, the difference between the required completion date and the planned completion date is calculated. Then the application searches for a threshold that applies to the most number of entities of the task. It searches for a threshold in the following order:

1. Combination of all four elements: Process name, process version, task name, and task type.
2. Process name, process version, and task name.
3. Process name and task name.
4. Process name, process version, and task type.
5. Process name and task type.
6. Task name.
7. Process name and process version.
8. Process name.
9. For task type.
10. For null keys.

The application searches for a threshold that applies to all four entities of the task: Task type, task name, process name, and process version. If a threshold for that combination is not found, then the application searches for a threshold that applies to the process name, process version, and task name of the task, and so

on. After an appropriate threshold is located, the score dictated by the threshold is assigned to the task.

Appearance in Order Orchestration Work Area

The jeopardy priority that appears in the Order Orchestration work area maps back to the task with the highest jeopardy score. In other words, if multiple tasks are in jeopardy within an orchestration process, then the highest jeopardy score is used to represent the jeopardy of the orchestration process. For example, in an orchestration process called Carpet Processing, insufficient supply in the warehouse causes several tasks to be delayed, including the Deliver Carpet task and the Invoice Carpet task. A three-day delay to the Deliver Carpet task maps to a jeopardy score of 100 and a jeopardy priority of Medium; a three-day delay to the Invoice Carpet task carries a jeopardy score of 200 and a jeopardy priority of High. Two hundred is the higher score, so this task's jeopardy score is used to represent the jeopardy of the Carpet Processing orchestration process. In the Order Orchestration work area, this orchestration process displays a jeopardy priority of High.

Task Status Mappings: Explained

Fulfillment tasks have predefined status codes. You can choose to display different status codes from the predefined ones by mapping the predefined status codes to your preferred status codes. The status codes that you change them to appear in the Order Orchestration work area and in other status management areas, as well, such as the Status Conditions tab of an orchestration process definition.

Jeopardy Thresholds: Explained

Jeopardy thresholds are used to monitor and measure orchestration processes. Jeopardy thresholds are ranges of time that a task is delayed. You define a set of ranges for each task of an orchestration process and then assign a score that indicates the severity of the delay. These setups are used to create indicators that appear on the Order Orchestration work area. These indicators help order managers to quickly see the severity of a delay, enabling them to take appropriate action.

When an orchestration process is assigned to an orchestration order, the process is planned taking into account the lead time of steps in the orchestration process and certain key dates from the sales order, such as required completion date. Each task of the process has a planned start and completion date. When a task of the orchestration process is delayed, the whole process is replanned. When a task in the process is expected to be completed after the required completion date of the task, a jeopardy score is automatically assigned to each task based on the jeopardy thresholds you define.

You can define jeopardy thresholds for any combination of the following:

- Task type
- Task name
- Process name
- Process version

You are not required to choose any of the above options. If you leave them at their default setting of All, then the jeopardy threshold applies to all tasks.

Prerequisites

If you want to apply the threshold to a task or orchestration process, then orchestration processes, tasks, and task types must be defined so that you can select them when creating jeopardy thresholds.

Orchestration Process Definition: Points to Consider

Orchestration process definitions contain the information that is used to create an orchestration process at run time. When defining an orchestration process, your choices affect how a fulfillment line is processed to completion.

Oracle Fusion Distributed Order Orchestration provides the following predefined orchestration processes:

- Ship order
- Return order

The Ship Order orchestration process contains the following sequential tasks:

1. Schedule
2. Reservation
3. Shipment
4. Invoice

The Return Order orchestration process contains the following sequential tasks:

1. Return Receipt
2. Invoice

Prerequisites

Before you define an orchestration process, perform the following prerequisite tasks:

- Execute mandatory tasks in Functional Setup Manager.
- Define any additional task types and their associated tasks and services.
- Define any additional status codes and how they will be used for task types, fulfillment lines, and orchestration processes.
- Define any subprocesses that will be used in the process to be defined.
- Define status catalogs that will be used for status conditions. Define the catalogs in Oracle Fusion Product Model, Oracle Fusion Product and Catalog Management, or Oracle Fusion Product Hub.

Header

The header contains basic information that applies to the entire orchestration process. During step definition, you will determine the information that applies to individual steps.

- **Orchestration process class:** This value is required. It is a set of statuses that can be used for this orchestration process.
- **Change mode:** When a change order enters the system, the delta service analyzes the state of each step to determine what processing changes are needed. After this analysis, compensation occurs on the necessary steps. Your selection determines how often snapshots of the orchestration process are taken.
 - **None:** Snapshot of the orchestration process is not taken, and change is not allowed.
 - **Simple:** Snapshot is taken when orchestration process starts and at step where change order is received.
 - **Advanced:** Snapshots are taken at each orchestration process step.

Caution

If you used the Functional Setup Manager migration tool to port test instance data to a production environment, then do not change the process name in either instance. Changing the name could prevent references to other data in the orchestration process from being updated.

Orchestration Process Classes: Explained

An orchestration process class is a set of statuses that can be used for an orchestration process. Use orchestration process classes to simplify orchestration process definition. You can assign the complete set of statuses to any number of orchestration process definitions without having to list the statuses one by one. You do not have to use all the status values in the orchestration process class.

When an orchestration process class is assigned to an orchestration process, you can use only the statuses in that class. The status values that are defined in the orchestration process class are only for the statuses at the orchestration process level, not for the tasks or fulfillment lines.

Status Catalog: Explained

Your organization might need for different fulfillment lines within the same orchestration process to have different status progressions. For example, a model with shippable fulfillment lines and nonshippable fulfillment lines may require different statuses for each type of fulfillment line. A status catalog provides a means to group like items, so that they can achieve the same statuses at the same time. Status catalogs are defined in Oracle Fusion Product Model.

Cost of Change: Explained

Cost of change is a numerical value that represents the impact of a change to an orchestration process. Cost could be the monetary cost to a company or the difficulty associated with the change. This value is calculated and returned to the order capture application, so that the customer service representative can understand how costly it is to make the customer's requested change. The

cost of change value can be requested by the order capture application before a change order is submitted to determine whether it should be submitted. Cost of change is calculated also after compensation of a change order is completed. Cost of change is most often used in a postfactor analysis to change practices or processes, or in a business-to-business negotiation.

You assign the cost of change to the orchestration process using a business rule. When the order capture application requests a cost of change determination, the value is calculated and returned, but it is not stored. If you choose not to use cost of change values, then zero is the value used to calculate the cost of change when a change order is submitted.

Creating Cost of Change Rules: Worked Example

This example demonstrates how to create a cost of change rule for an orchestration process, so that order managers are aware of how costly to the company certain changes are. The order administrator of a flooring company wants a few rules that indicate the cost to the company if a change is requested when the fulfillment line is at a certain status. The cost of change is low if the fulfillment line is in Scheduled status, and it is much higher if the fulfillment line is in Shipped status.

Note

The following is an example of a simple rule, which is well suited for rules for an orchestration process with a single line. If you want to write a rule for an orchestration process that has multiple lines, then use advanced mode rules. For more information, see Oracle Fusion Middleware User's Guide for Oracle Business Rules.

Summary of the Tasks

Create If and Then statements for the following rules:

- If the fulfillment line status value is Shipped, then the cost of change is 50.
- If the fulfillment line status value is Scheduled, then the cost of change is 5.

Creating the If Statement of the First Rule

Create the If statement: If the fulfillment line status value is Shipped.

1. On the header of the Orchestration Process Definition page, click **Click for Rule** next to Cost of Change.
2. Click the **New Rule** icon.
3. Optionally, enter a rule name.
4. Click the arrows to expand the rule.
5. Click the **Left Value** icon to search for the left value.
6. Expand DOOFLine.
7. Select Status Value.
8. Click **OK**.

9. Select Is.
10. In the field to the right of your Is selection, enter "SHIPPED". You must surround the text with quotation marks.

Creating the Then Statement of the First Rule

Create the Then statement: Then the cost of change is 50.

1. Click **Insert Action**.
2. Select Assert New.
3. Select Result.
4. Select the **Edit Properties** icon.
5. In the **Value** column of resultObjKey, enter 50.
6. Click **OK**.

Creating the If Statement of the Second Rule

Create the If statement: If the DOO Fulfillment Line status value is Scheduled.

1. Click the **New Rule** icon.
2. Click the arrows to expand the rule.
3. Click the **Left Value** icon to search for the left value.
4. Expand DOOFLine.
5. Select Status Value.
6. Click **OK**.
7. Select Is.
8. In the field to the right of your "is" selection, enter "SCHEDULED". You must surround the text with quotation marks.

Creating the Then Statement of the Second Rule

Create the Then statement: Then the cost of change is 5.

1. Click **Insert Action**.
2. Select Assert New.
3. Select Result.
4. Select the **Edit Properties** icon.
5. In the **Value** column of resultObjKey, enter 5.
6. Click **OK**.
7. Click **Save**.

Creating Line Selection Rules: Worked Example

This example demonstrates how to create a line selection rule that determines which lines to process for a particular step in a case where not all lines should

be processed by that step. The order administrator of a company that sells DVRs wants an orchestration process that handles orders for this equipment. The orchestration order is broken into several fulfillment lines for each of the following: DVR, remote control, instruction manual, and extended warranty. The extended warranty is a contract purchased online, but it is not a shippable item. Therefore, it should not be sent to the fulfillment system during the Shipment task.

Note

The following is an example of a simple rule, which is well suited for rules for an orchestration process with a single line. If you want to write a rule for an orchestration process that has multiple lines, then use advanced mode rules. For more information, see Oracle Fusion Middleware User's Guide for Oracle Business Rules.

Summary of the Requirements

Create the rule while defining the SetUpShipment step. To create the rule, you must construct If and Then statements.

1. Create the If statement: If the item is shippable.
2. Create the Then statement: Then select the fulfillment line.

Creating the If Statement

1. In the **Line-Selection Criteria** column of the Manage Orchestration Process Definition page, select **Click for Rule** for the step that you are defining. Disregard all the other information above the line. Do not change the rule set name. This rule set will contain all the line selection rules you write for this step.
2. Click the **New Rule** icon.
3. You may enter a rule name. This is optional.
4. Click the arrows to expand the rule.
5. Click the **Left Value** icon to search for the left value.
6. Expand DOOFLine.
7. Select ShippableFlag.
8. Select Is.
9. In the next blank field to the right, enter "Y". You must surround the text with quotation marks.

Creating the Then Statement

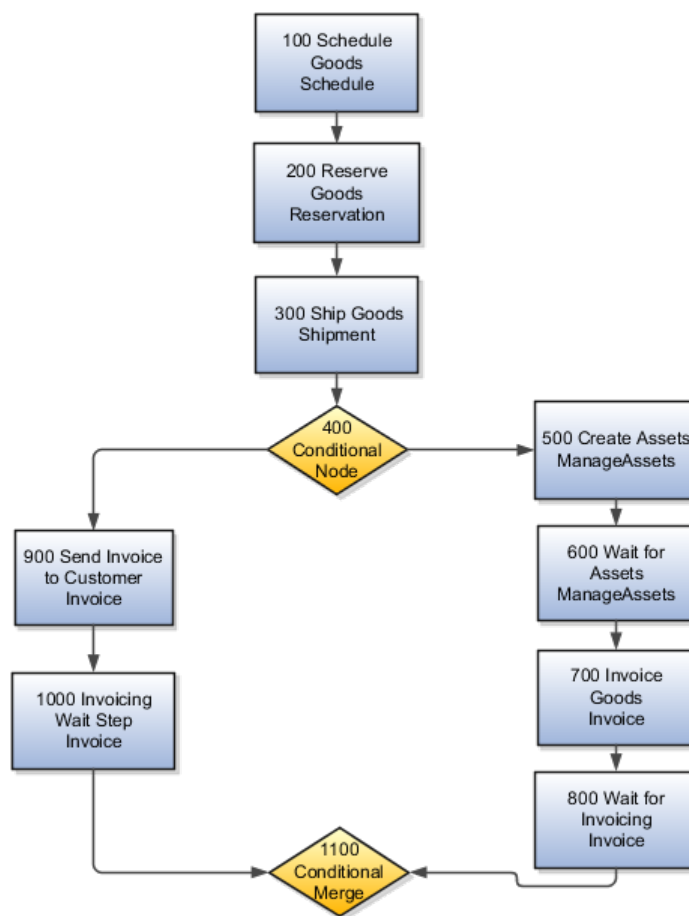
1. Click **Insert Action**.
2. Select Assert New. You must select this option for all line selection rules.
3. Select Result.
4. Click the **Edit Properties** icon.
5. In the **Value** field for ResultObjKey, search for DOOFLine.

6. Select FulfillLineId.
7. Click **OK**.

Branching: Explained

Use branching to create a sequence of steps that are executed only under certain conditions. A branch contains one or more steps. For example, your company sells laptop computers. If a customer buys a service agreement with the laptop computer, then you create an asset, so that the computer can be tracked. If a service agreement is not purchased, then the customer is invoiced without creating an asset.

The following figure shows an orchestration process flow that models this scenario. Each step contains the step number, task name, and task type. This example includes the ManageAssets custom task type. The conditional node indicates that an orchestration process is about to branch. The first step of the branch contains the condition. If the condition is met, then the application executes the steps on the branch that includes the Create Asset and Wait for Asset steps. Otherwise, the other branch is executed, and an invoice is created without creating an asset.



You do not need to set an **Otherwise** condition in the orchestration process definition if you have only one branch. When the orchestration process artifacts are generated, an empty default branch is added.

Creating Branching Condition Rules: Worked Example

This example demonstrates how to create a branching condition that determines whether to branch from the parent process to execute a branch. In this scenario, the order administrator of a flooring company wants an orchestration process for carpet orders. The company has a policy stipulating that a representative call a customer before sending an invoice over \$50,000.00.

Note

The following is an example of a simple rule, which is well suited for rules for an orchestration process with a single line. If you want to write a rule for an orchestration process that has multiple lines, then use advanced mode rules. For more information, see Oracle Fusion Middleware User's Guide for Oracle Business Rules.

Summary of the Requirements

Create a rule on the invoicing step of the orchestration process definition. To create the rule, you must construct If and Then statements.

1. Create the If statement: If invoice is greater than \$50,000.00
2. Create the Then statement: Then execute the branch.

This example assumes that an orchestration process is created that contains the steps necessary to carry out the fulfillment of a customer's order for carpet. This example begins with a Call Customer step. Ensure that the Call Customer step is the step after the conditional step.

Creating the If Statement

Create the If statement: If invoice (price) is greater than \$50,000.00.

1. On the Create Orchestration Process Definitions page, go to the invoicing step.
2. In the **Branching Condition** column, select **Click for Rule**.
3. Do not change the rule set name.
4. Click the **New Rule** icon.
5. Optionally, enter a rule name.
6. Click the arrows to expand the rule.
7. Click the **Left Value** icon to search for the left value.
8. Expand DOOFLine.
9. Select ExtendedAmount.
10. Click **OK**.
11. Select Is Greater.
12. In the field to the right, enter 50,000.

Creating the Then Statement

Create the Then statement: Then execute the branch.

1. Click **Insert Action**.
2. Select **Assert New**.
3. Select **Result**.
4. Select the **Edit Properties** icon.
5. In the **Value** field for **ResultObjKey**, search for **Boolean**.
6. Select **True**.
7. Click **Save**.

Lead Time: Explained

Lead time is the expected amount of time needed to complete a step. It is used to plan the orchestration process and predict expected completion dates. When real completion dates are available, they are used instead of the estimates in the orchestration process definition. The planned orchestration process appears in the Gantt chart in the Order Orchestration work area. Lead time is also used during jeopardy calculation where jeopardy is determined by considering the number of days past lead time a step is taking.

Creating Lead-Time Rules: Worked Example

This example demonstrates how to create a lead-time rule that determines lead time for a step based on a set of conditions. The order administrator of a flooring company wants an orchestration process that handles carpet orders. The lead time for shipping the carpet is two days if the inventory organization is Houston and four days for any other inventory organization.

Note

Often, if you write a rule for an orchestration process that has multiple lines, then you should use advanced mode rules. In the following example, however, all the lines are being treated the same way, so an advanced mode rule is not required.

Creating Lead-Time Rules

In this example, you create the rule while defining the Shipment step **Create Shipment**. Ensure that the unit of measure is days. You must create two rules, one for when the inventory organization ID is Houston, and the other for an inventory organization ID with any other value.

Note

The Shipment task has a wait step, where a lead time can be defined, too. The lead time for the task is the sum of the lead times defined for each of the

steps within the task. In this example, lead time is defined only on the Create Shipment step.

1. Create the If statement for the first rule: If inventory organization ID is 1234440.
2. Create the Then statement for the first rule: Then the lead time is equal to 2.
3. Create the If statement for the second rule: If inventory organization ID isn't 1234440.
4. Create the Then statement for the second rule: Then the lead time is equal to 4.

Creating the If Statement for the First Rule

1. In the **Lead-Time Expression** column of the Manage Orchestration Process Definition page, select **Click for Rule** for the step that you are defining. Disregard all the other information above the line. Do not change the rule set name. This rule set contains all the lead-time rules that you write for this step.
2. Click the **New Rule** icon.
3. Optionally, enter a rule name.
4. Click the arrows to expand the rule.
5. Click the **Left Value** icon to search for the left value.
6. Expand DOOFLine.
7. Select InventoryOrganizationId.
8. Click **OK**.
9. Select Is.
10. Enter 1234440.

Creating the Then Statement for the First Rule

1. Click **Insert Action**.
2. Select Assert New. You must select this option for all lead-time rules.
3. Select Result.
4. Select the **Edit Properties** icon.
5. In the **Value** column for ResultObjKey, enter 2.

Creating the Second Rule

In the same window, repeat the steps above to create a rule for the following If and Then statements. Start by clicking the **New Rule** icon. Substitute "isn't" for "is" in the first statement, and substitute 4 for 2 in the second statement.

1. If the inventory organization ID isn't 1234440.
2. Then the lead time is equal to 4.
3. Click **OK**.

Orchestration Process Planning: How It Works

You can create customized processes to manage each stage of order processing after the order is released from the order capture system. These orchestration processes include automated planning. Process planning sets user expectation of completion date of each step, task, and the orchestration process itself.

Settings That Affect Orchestration Process Planning

If you select **Replan Instantly** for an orchestration process, then the planning engine is called and the data is refreshed after each step is completed. For performance reasons, you might not want automatic replanning of some processes, especially where the step definition sequence is long or complex. Consider using this option for orchestration processes that are for high priority customer orders, or with jeopardy thresholds of less than a day. If you do not select **Replan Instantly**, then the planning data is refreshed during its normal, scheduled replan.

The following attributes affect step-specific planning:

- **Planning Default Branch:** Identifies the default path for planning. This attribute is used only if the orchestration process has conditional branches.
- **Fulfillment Completion Step:** Identifies the step at which the customer considers the lines fulfilled. This attribute is used in planning calculations to satisfy the customer request date. It is not necessarily the last step in the process definition. The chronological last step may not be last step that the customer cares about. The orchestration process is planned with the customer request date as the completion date for the step identified in the process as the last step.
- **Default Lead Time/Lead Time UOM:** Lead time is the expected duration for a given unit of work to be completed. If a lead-time expression is not defined for a step, then the default lead time is used.
- **Lead-Time Expression:** Define lead times using Oracle Business Rules. This method provides flexibility when defining complex lead-time expressions.

How Orchestration Process Planning Is Calculated

When an order enters Oracle Fusion Distributed Order Orchestration it is transformed into fulfillment lines. Then orchestration processes are created and assigned to the fulfillment lines. The orchestration process is first planned when the orchestration process is created. Planning is based on the requested date of the sales order. The requested date becomes the required completion date for the last step (step identified by the Last Fulfillment Completion Step indicator and not the chronological last step) of the orchestration process. The application then calculates the planned dates for each step and task, starting from the first chronological step, using the lead time you define. The schedule appears in the Order Orchestration work area.

The orchestration process is replanned every time an update is received from the fulfillment system. You can control when process planning occurs by scheduling a regular planning update for the frequency you want using a scheduled process.

Creating Compensation Patterns: Worked Example

This example demonstrates how to create a compensation pattern that determines what adjustments to make for a processing task in response to a requested change. The order administrator of a flooring company wants a rule that indicates that if the requested ship date is 11/20/2010, then cancel and redo the ShipGoods task.

Note

The following is an example of a simple rule, which is well suited for rules for an orchestration process with a single line. If you want to write a rule for an orchestration process that has multiple lines, then use advanced mode rules. For more information, see Oracle Fusion Middleware User's Guide for Oracle Business Rules.

Summary of the Requirements

To create the rule, you must construct If and Then statements.

1. Create the If statement: If the RequestShipDate is 11/20/2010.
2. Create the Then statement: Then redo the ShipGoods task.

Creating the If Statement

Create the If statement: If the RequestShipDate is 11/20/2010.

1. In the **Compensation Rule** column of the Manage Orchestration Process Definition page, select **Click for Rule**.
2. Do not change the rule set name.
3. Click the **New Rule** icon.
4. Optionally, enter a rule name.
5. Click the arrows to expand the rule.
6. Click the **Left Value** icon to search for the left value.
7. Expand FulfillLineTLVO.
8. Select RequestShipDate.
9. Click **OK**.
10. Select Is.
11. Click the **Right Value** icon to search for the right value.
12. Expand CurrentDate.
13. Select Date.
14. Click **OK**.

Creating the Then Statement

Create the Then statement: Then redo the ShipGoods task.

1. Click **Insert Action**.
2. Select **Assert New**.
3. Select **Result**.
4. Select the **Edit Properties** icon.
5. In the **Value** column of resultObjKey, enter "Redo". You must surround the text with quotation marks.
6. Click **OK**.

Orchestration Process Status Definition: Points to Consider

When you define an orchestration process, you must select an orchestration process class, which provides a defined set of statuses for any orchestration process to which it is applied. Use orchestration process-specific statuses to apply different sets of statuses and rule logic for different items to show the progression of the process. For example, you could have a set of statuses and rule logic for orchestration processes for textbooks for customers that are colleges and a different set of statuses and rule logic for orchestration processes for textbooks for customers that are primary schools.

If you choose not to customize the status condition for an orchestration process, then the default statuses are used. If you customized the name of the default status, then the status appears in the application.

Orchestration Process Classes

The orchestration process class is a set of status codes. When you select a process class in the header, the status codes from that class are available for selection when you create the status conditions. These are the status codes that will represent the status of the orchestration process and will be seen throughout the application. The status code is also used for grouping orchestration processes by status to enable order managers to quickly identify orchestration processes that are in the same status.

Orchestration Process Status Rules

You can set up rules that govern under what conditions a status is assigned to an orchestration process. For example, you could create a rule that says if the status of the Schedule task is Not Started, then assign the orchestration process the status **Unscheduled**. You must designate a status or set of statuses to indicate that a task is complete. You can only select from those that were defined to mark a task complete.

Orchestration Process Status: How It Is Determined

During processing of an orchestration order, the tasks of the assigned orchestration process are fulfilled step by step. A default set of sequential statuses is provided for the fulfillment tasks, but you can also create your own fulfillment task statuses and sequences for an orchestration process. You must determine the status that will be assigned to an orchestration process at each

stage of the process. For example, if a Schedule Carpet task has a status of Unsourced, what status should the orchestration process have?

Settings That Affect Orchestration Process Status

You can designate the statuses that represent an orchestration process when you define the orchestration process. These statuses are used to represent the status of the orchestration process throughout the application. You can select a preset group of orchestration process statuses by selecting an orchestration process class. You can create rules that govern how statuses are attained during the orchestration process and the conditions under which they are assigned to the orchestration process task.

How Orchestration Process Status Is Determined

If rules are created, then at run time the application evaluates each of the statements sequentially. The true condition with the highest sequence number determines the status of the orchestration process.

When a fulfillment line splits, the resulting fulfillment lines have duplicate tasks. At some point, the tasks could have different statuses. For example, the Schedule task for fulfillment line A1 is in status Not Scheduled, and the Schedule task for fulfillment line A2 is Scheduled. In this case, the application searches for the split priority of the task statuses. The status with the higher split priority (lower number) becomes the status of the orchestration process.

Creating Orchestration Process Status Conditions: Worked Example

This example demonstrates how to create status conditions for an orchestration process. A company that sells flooring needs an orchestration process that reflects the steps required to fulfill such orders. The orchestration process definition must designate how to reflect the status of the orchestration process at any point in time. The status of the orchestration process is based on the status of the tasks. This example shows how to create the conditions that designate the status of the orchestration process.

When you create an orchestration process status condition, you must decide which orchestration process class to use and which statuses you want to reflect the status of the orchestration process. An orchestration process class is a set of statuses that can be used for an orchestration process.

Prerequisites

This example assumes that an administrator has created an orchestration process class called Carpet Class on the Manage Status Values page.

1. On the Manage Status Values page, create an orchestration process class called Carpet Class.
2. For the Schedule task type, include the following statuses: Started, Canceled, Not Started.

Summary of the Requirements

1. Create an orchestration process definition called Standard Carpet.
2. Select the Carpet Class orchestration process class.

3. Create the orchestration process status condition: If the ScheduleGoods task status is Scheduled, then use Scheduled to represent the status of the orchestration process.

Creating an Orchestration Process Status Condition

This example shows you how to create one orchestration process status condition. Repeat these steps for all the status conditions you need. You are not required to use all the statuses in the orchestration process class.

1. On the Manage Orchestration Process Definitions page, click **Create**.
2. In the **Process Name** field, enter Standard Carpet.
3. In the **Process Class** list, select Carpet Class.
4. On the Step Definition tab, select **Add Row**.
5. In the **Step Name** field, enter ScheduleGoods.
6. In the **Step Type** field, select Service.
7. In the **Task Type** field, select Schedule.
8. In the **Task Name** field, select ScheduleGoods.
9. In the **Service Name** field, select Create Scheduling.
10. Click **Save**.
11. Select the Status Conditions tab.
12. On the Orchestration Process Status Values tab, select **Add Row**.
13. In the **Sequence** field, insert 1.
14. In the **Status Value** list, select Scheduled to represent the first status of the orchestration process: .
15. Click the **Expression** icon.
16. On the Tasks tab, select ScheduleGoods.
17. Click **Insert Into Expression**.
18. On the Operators tab, select =.
19. Click **Insert Into Expression**.
20. On the Tasks tab, expand Schedule.
21. Select Scheduled.
22. Click **Insert Into Expression**.
23. Click **OK**.

Fulfillment Line Status Definition: Points to Consider

When you create an orchestration process definition, you can opt to define status conditions for certain types of fulfillment lines that can be processed by the orchestration process. Use fulfillment line-specific status conditions to apply different sets of statuses and rule logic for different items. For example, you could have one set of status conditions for textbooks and another set for paperback books.

If you choose not to create status conditions for a fulfillment line, then the status conditions with the status rule set that is assigned to the default category dictates the status progression.

Status Catalogs and Categories

Your organization might need for different fulfillment lines within the same orchestration process to have different status progressions. For example, a model with shippable fulfillment lines and nonshippable fulfillment lines may require different statuses for each type of fulfillment line. A status catalog provides a means to group like items, so that they can achieve the same statuses at the same time. Status catalogs are defined in Oracle Fusion Product Model.

You can select a status catalog when you create an orchestration process definition. Status catalogs that meet the following criteria are available for selection:

- An item exists in only one category within a catalog.
- A category contains items or subcategories but not both.
- A catalog is controlled at the master level only, not at the organization level.

You can use catalogs and categories in multiple orchestration process definitions. Use a category to ensure that the same set of status conditions is applied to specific sets of fulfillment lines. The same status conditions are applied to all fulfillment lines that have the item that belongs to that category.

Status Rule Set

Whether or not you use status catalogs, you can use status rule sets to apply a set of sequential statuses to the fulfillment line that is processed by the orchestration process. A status rule set is a set of rules that govern the conditions under which status values are assigned to fulfillment lines. When you create a status rule set, you determine the status that will be assigned to a fulfillment line at each stage of the process. For example, if an item has a status of Unsourced, then the fulfillment line will have the status Unscheduled. A status rule set streamlines administration by enabling you to use a rule set with any number of fulfillment lines, rather than by entering separate rules for each fulfillment line. You can also apply the same logic to multiple categories.

In the case where a parent and a child category refer to different status rule sets, the child takes priority. This allows you to define an All category to handle all items in one definition, as well as to add an additional subcategory for a subset of products that needs to use a different status rule set.

During order processing, the application assigns an overall status to each orchestration order. This status is determined by assigning the orchestration order the status of the fulfillment line that has progressed the furthest in the order life cycle. To determine the fulfillment line status, the application evaluates each of the status conditions of the fulfillment line sequentially. The true condition with the highest sequence number determines the status of the fulfillment line.

Caution

If you used the Functional Setup Manager migration tool to port test instance data to a production environment, then do not change the status rule set name in either instance. Changing the name could prevent references to other data in the orchestration process from being updated.

Creating Fulfillment Line Status Conditions: Worked Example

This example demonstrates how to create status conditions for a fulfillment line with several items that require different statuses. A flooring company is setting up orchestration processes to handle orders for different types of flooring. The same orchestration process could be used for multiple types of flooring, but the administrator wants to define statuses for each type of flooring separately because they require slightly different statuses. This example demonstrates how to select the status catalog and create the status conditions for a single category of items within an orchestration process.

When you create an orchestration process, you need to decide whether you want different fulfillment lines that get assigned to the process to have different statuses as they progress through fulfillment. If so, you must determine how to group the fulfillment lines using catalogs and categories.

The Flooring catalog has the following categories: Carpet, Tile, Hardwood. You select the category for Carpet. You create a status rule set with conditions that will yield the following statuses: Not Scheduled, Scheduled, Awaiting Shipment, Shipped, Billed.

Prerequisites

This example assumes that an administrator has created fulfillment line status values on the Manage Status Values page. This example also assumes a Flooring catalog was created in Oracle Fusion Product Model.

1. On the Manage Status Values page, create fulfillment line status values.
2. In Product Model, create a Flooring catalog with the following categories: Carpet, Tile, Hardwood.
3. Assign the carpet items to the carpet category.

Summary of the Requirements

1. Create an orchestration process definition called Standard Flooring.
2. Select Flooring as the status catalog.
3. Create the status rule set Carpet Rule Set, and assign it to the Carpet category within the Standard Flooring orchestration process.
4. Create the following fulfillment line status condition for Carpet Rule Set: If the Schedule task status is Started, then use Scheduled to reflect the status of the fulfillment line.

Creating a Fulfillment Line Status Condition

This example shows you how to create one fulfillment line status condition. Repeat these steps for all the status conditions you need.

1. On the Manage Orchestration Process Definitions page, click **Create**.

2. In the **Process Name** field, enter Standard Flooring.
3. In the orchestration process definition header, select the Flooring status catalog.
4. On the Step Definition tab, select **Add Row**.
5. In the **Step Name** field, enter ScheduleGoods.
6. In the **Step Type** field, select Service.
7. In the **Task Type** field, select Schedule.
8. In the **Service Name** field, select Create Scheduling.
9. Click **Save**.
10. In the Fulfillment Line Status Values tab, select and add the Carpet status category.
11. In the **Status Rule Set** column, click the arrow and select Create.
12. In the field type Carpet Status Rule Set, ensure that Create New is selected, and click **Save and Close**.
13. Click **Save** at the top of the page.
14. Click **Edit Status Rule Set**.
15. Add a row.
16. On the Tasks tab, select ScheduleGoods.
17. Click **Insert Into Expression**.
18. On the Operators tab, select =.
19. Click **Insert Into Expression**.
20. On the Tasks tab, expand Schedule.
21. Select Scheduled.
22. Click **Insert Into Expression**.
23. Click **OK**.
24. In the Status Value column, select Scheduled.
25. Click **OK**.

Orchestration Process Definition Deployment: Explained

After you finish creating or updating an orchestration process definition, you must release it and then deploy it on an instance of Oracle Fusion Distributed Order Orchestration. Deploying the orchestration process makes it available for use by the application.

If you want to use the ShipOrderGenericProcess and ReturnOrderGenericProcess predefined orchestration processes, then you must generate them and deploy them; it is not necessary to release them.

After your orchestration process is defined, you must take the following steps to deploy it:

- Release the orchestration process definition.

- Download the orchestration process definition.
- Modify the SOA configuration plan.
- Deploy the JAR file using the modified configuration plan.

Do not modify orchestration process definitions outside of the Manage Orchestration Process Definition pages.

Release the Orchestration Process Definition

When an orchestration process is released it is automatically validated. After you release the orchestration process definition, batch-level validations are performed to ensure that the orchestration process was constructed correctly. If any errors are generated during validation, the release process stops and an error icon appears next to the orchestration process name. The list of errors is retained until the next time the batch validation runs. If the orchestration process is valid, then release of the process continues. An orchestration process is valid if no error messages are generated; warning messages may be associated with a valid process. After validation is complete, the orchestration process definition becomes read-only. At this point, the orchestration process is given Released status, and the BPEL artifacts needed to deploy and run the orchestration process are created and stored.

Download the Orchestration Process Definition

After you release an orchestration process definition, you deploy the downloaded artifacts to the server. Use Oracle Fusion Setup Manager to export the artifacts. Oracle Fusion Middleware is used to deploy artifacts.

1. On the Manage Orchestration Process Definitions page, select the orchestration process that you want to deploy.
2. Click the **Edit** icon.
3. In the Download Generated Process window, click **Download**.
4. Save the archive file that appears to a local directory.
5. Open the archive file in a local directory.

The JAR file is located in a Deploy folder within a folder bearing the name of the orchestration process that you downloaded.

Modify the SOA Configuration Plan

Modify the SOA configuration plan, replacing the host names and ports with your organization's Distributed Order Orchestration ADF server and port and Distributed Order Orchestration (Supply Chain Management) SOA server and port. Use the external-facing URLs of the servers. The configuration plan enables you to define the URL and property values to use in different environments. During process deployment, the configuration plan is used to search the SOA project for values that must be replaced to adapt the project to the next target environment.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAConfigPlan
xmlns:jca="http://platform.integration.oracle/blocks/adapter/fw/metadata"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy"
xmlns="http://schemas.oracle.com/soa/configplan">
```

```

<composite name="*">
<import>
<searchReplace>
<search/>
<replace/>
</searchReplace>
</import>
<service name="client">
<binding type="ws">
<attribute name="port">
</attribute>
</binding>
</service>
<reference name="*">
<binding type="ws">
<attribute name="location">
<searchReplace>
<search>http://localhost_am:port</search>
<replace>http://actualDOOADFserver:port</replace>
</searchReplace>
<searchReplace>
<search>http://localhost_soa:port</search>
<replace>http://actualDOOSOAServer:port</replace>
</searchReplace>
</attribute>
</binding>
</reference>
</composite>
</SOAConfigPlan>

```

Deploy the JAR File

To deploy the JAR file, you can use any of the following: Oracle Enterprise Manager Fusion Middleware Control, ant command line tool, or Oracle WebLogic Scripting Tool. For more information about deploying SOA composite applications, see Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite.

Process Assignment Rules: Explained

After Oracle Fusion Distributed Order Orchestration creates an orchestration order, the application assigns orchestration processes to fulfillment lines based on process assignment rules. Process assignment rules are executed in the Oracle Business Rules engine. Process assignment rules are built based on orchestration groups and using orchestration order attributes.

You do not need to specify versions or effectivity dates in the process assignment rules because versions and effectivity dates are controlled at the orchestration process level.

Orchestration Groups

A fulfillment line belongs to an orchestration group. Distributed Order Orchestration contains the following predefined orchestration groups: Shipment Set, Model/Kit, and Standard. Standard is used for standard items or finished items. All the fulfillment lines that belong to a shipment set or a model are assigned the same orchestration process.

Assign a process for each set of unique conditions. You can set up a default orchestration process for each orchestration group using the Otherwise construct.

Before you create process assignment rules, you must define orchestration processes or at least know the names you will give to orchestration processes. You will add the orchestration process names to bucket sets to make them available for selection when you create a process assignment rule.

Creating Process Assignment Rules: Examples

Use these scenarios to understand how to use process assignment rules.

Assigning an Orchestration Process According to Product

All orders for ceramic tile must undergo the same processing steps, so you write a process assignment rule that assigns the Tile Processing orchestration process to all orchestration order lines with tile as the product.

Assigning an Orchestration Process According to Customer

Customer A requires an extra inspection step for all its orders, so you write a process assignment rule that assigns the Customer A Process to all orchestration order lines that have Customer A in the orchestration order header.

Assigning an Orchestration Process According to Ship-to Region

Orders that are bound for countries outside your current location require different handling, such as completion of customs forms. You write a process assignment rule that assigns the International Orders orchestration process to all orchestration order lines that have a foreign country in the ship-to address in the header.

Define Processing Constraints

Define Processing Constraints: Overview

Each company has its own business rules, which must be applied during the orchestration process. The constraint framework allows for the implementation of those specific requirements. Processing constraints are rules that control attempted changes to an order: What can be changed, when, and by whom.

At runtime, processing constraints are checked on changes to orchestration orders, orchestration order lines, and fulfillment lines. Changes that are not permitted by processing constraints are not allowed. A message is returned indicating the reason the change is not permitted.

Processing constraints are used also to validate the required attributes for fulfillment requests.

Some processing constraints are predefined; you cannot change these processing constraints. If you want to make processing constraints more restrictive, then you must create new ones.

Using Processing Constraints: Examples

Consider using processing constraint in scenarios such as the following. In all of these scenarios, the change is submitted, but it is never processed because

the processing constraint rejects it. A message is returned to the order capture application indicating that the change could not be made because of the processing constraint.

Constraint Prohibits Changes at Shipping Stage

An orchestration process gets to the shipping stage. Then a change order is submitted against the orchestration order. The orchestration process is so far along that it is costly and impractical to make changes. To prevent this problem, you create a processing constraint that rejects any changes when an orchestration process is in the shipment step.

Constraint Rejects Orders Without a Required Attribute

Your company has a policy that it does not deliver items to an address that does not have a ship-to contact. Sometimes sales orders that do not have a ship-to contact are submitted. To prevent this problem, you create a processing constraint that rejects sales orders that do not have the required information.

Constraint Prohibits Changes by Unauthorized User

Your company allows customer service representatives to submit certain customer changes without approval from a manager. If the change order has a transaction value over \$100, then the change must be submitted by a manager. You create a processing constraint that rejects changes orders with transaction values over \$100 from anyone with the customer service representative role.

Creating Processing Constraints: Worked Example

This example demonstrates how to create a processing constraint that prevents changes to any orchestration process that is in the shipping phase. An orchestration process gets to the shipping stage. Then a change order is submitted against the orchestration order. The orchestration process is so far along that it is costly and impractical to make changes. To prevent this problem, you create a processing constraint that rejects any changes when an orchestration process is in the SetupShipment step.

Before you create a processing constraint, you must create a constraint entity, a validation rule set, and a record set.

Summary of Requirements

1. Create the Shipping service as a constraint entity.
2. Create a validation rule set.
3. Create a record set.
4. Create a processing constraint that prohibits changes when the orchestration process reaches the shipping stage.

Creating a Constraint Entity

This is the process of creating a process task entity so that it can be used later to create a processing constraint. This is the entity that will be constrained.

1. On the Manage Constraint Entities page, click the **Create** icon.

2. In the **Entity Name** field, enter Shipping Service.
3. In the **Description** field, enter the following: Shipping service entity.
4. From the list of services, select the Create Shipping service; do not select a process name or task name.
5. Click **Save**.

Creating a Validation Rule Set

A validation rule set names a condition and defines the semantics of how to validate that condition for a processing constraint.

1. On the Manage Processing Constraints page, select the Validation Rule Set tab.
2. Click the **Add Row** icon.
3. In the **Name** field, enter Shipment VRS.
4. From the **Validation Type** list, select Process.
5. Select the Create Shipping service
6. Click **Save**.

Creating a Record Set

A record set is a group of records that are bound by common attribute values. You can define conditions and specify a record set to be validated for a given condition as defined by the validation template.

1. On the Manage Processing Constraints page, select the Record Set tab.
2. Click the **Add Row** icon.
3. In the **Name** field, enter Fulfillment Lines That Belong to Same Customer.
4. In the **Description** field, enter the following: A record set created on fulfillment lines that belong to the same customer.
5. In the **Short Name** field, enter FFLCUST.
6. In the **Entity** list, select Order Fulfillment Line.
7. In the Attributes region, click the **Add Row** icon.
8. Select Bill-to customer ID.
9. Click **Save**.
10. Click Generate Packages.

Creating a Processing Constraint

Now that you have created a constraint entity, validation rule set, and record set, you can create the processing constraint.

1. On the Manage Processing Constraints page, select the Constraints tab.
2. Click the **Add Row** icon.
3. In the **Constraint Name** field, enter Shipping Constraint.
4. From the **Constraint Entity** list, select Order Fulfillment Line.

5. From the **Constrained Operation** list, select Update.
6. On the Conditions tab, click the **Add Row** icon.
7. In the Group Number field, enter 10.
8. From the **Validation Entity** list, select Order Fulfillment Line.
9. From the **Validation Rule Set** list, select Shipment VRS.
10. From the **Record Set** list, select Fulfillment Lines That Belong to Same Customer.
11. Create the following message: The fulfillment line could not be updated because it is in the shipping phase.
12. On the Applicable Roles tab, select **All Roles**.
13. Click **Save**.

Define Transformation Details

Sales Orders: How They Are Transformed to Orchestration Orders

When sales orders enter Oracle Fusion Distributed Order Orchestration from disparate order capture applications, they must be transformed into business objects that can be processed by Distributed Order Orchestration. During this process, called decomposition, sales orders are deconstructed and then transformed into Distributed Order Orchestration business objects.

Settings That Affect Sales Order Transformation

Business rules determine how sales orders are transformed. The following business rules are available: Pretransformation defaulting rules, product transformation rules, posttransformation defaulting rules, and process assignment rules.

How Sales Orders Are Transformed

Sales orders are transformed as follows:

1. The sales order is passed from the order capture application.
2. The connector service transforms the sales order from an external order capture system to a canonical business object called the sales order enterprise business object. The sales order enterprise business object structurally transforms the sales order from an external order capture system to an orchestration order in Distributed Order Orchestration. The Receive and Transform service, SalesOrderOrchestrationEBS, looks up the cross-reference values from the customer data hub, Oracle Fusion Trading Community Model, and Oracle Fusion Product Model to determine whether the sales order values must be transformed to common values to populate the sales order enterprise business object. Cross-referencing is required for static data such as country code and currency codes, as well as for dynamic data such as customers and products. The attributes come from different sources: Product Model, Trading Community Model,

and the order orchestration and planning data repository. If the order capture and Distributed Order Orchestration systems use different domain values, then the connector service transforms the structure and values from the order capture system domain to the Distributed Order Orchestration domain. The Receive and Transform service is called in the default predefined process prior to storing the sales order.

3. The connector service calls the decomposition process composite enterprise business function through a decomposition enterprise service. The decomposition process composite is exposed as a WSDL that can be called as a service from the connector service through an enterprise business service.
4. The decomposition service calls the requested operation (create, delete, update, or cancel orchestration order).
5. The decomposition service accepts the sales order enterprise business message as input. The decomposition service returns a sales order enterprise business message as the output.
6. The product is transformed according to the business rules that you write.
7. The Assign and Launch service assigns orchestration processes to line items according to the business rules that you write.

Order capture services are used to communicate updates back to the order capture system. To receive updates, the order capture system must subscribe to the events.

Connector for Sales Order Transformation: Explained

The connector service transforms the sales order business object as understood by an order capture application to an enterprise business object. The connector service then calls the Receive and Transform service.

The connector transforms the structure and content of the sales order.

Structural Transformation

The connector service transforms the sales order from an external order capture system to a canonical business object called the sales order enterprise business object. The sales order enterprise business object structurally transformations the sales order from an external order capture system to an orchestration order in Oracle Fusion Distributed Order Orchestration. The decomposition service accepts a sales order enterprise business message as the input and returns a sales order enterprise business message as the output. You can create the connector according to your organization's requirements, using the sales order enterprise business object attributes that are used by Distributed Order Orchestration.

Cross-Referencing: Explained

You must establish and maintain cross-references to relate business data between different integrated order capture and fulfillment systems and Oracle Fusion Distributed Order Orchestration.

Note the location of and other pertinent information about the following cross-references:

- Customer cross-references
- Item cross-references
- Other cross-references

Customer Cross-References

Customer cross-references are maintained in the Oracle Fusion Trading Community Model. You can use external customer hubs with Distributed Order Orchestration, but you must maintain cross-references in Trading Community Model also, so that Distributed Order Orchestration can resolve the Oracle Fusion customer values and vice versa. You can capture or set up customer cross-references in the Oracle Fusion customer model as part of the customer creation and update process.

During order processing, the order is created in the order capture system and sent to Distributed Order Orchestration, along with customer data. If the customer already exists in the Fusion customer master, then Distributed Order Orchestration uses a cross-reference to obtain the master customer record and the customer ID for the intended order fulfillment system. Then the decomposed order is sent, along with the customer ID and necessary attributes from the master.

Item Cross-References

Item cross-references are maintained in the Oracle Fusion Product Model. The cross-reference is established between the source system item and the item in the master product information repository, which is the Product Model. Two types of relationships are used for the cross-references: Source system item relationship, which captures the relationship between the source item and the Fusion item when a product hub is used; and a named item relationship, which is used to store the cross-reference between the source item and the Fusion item. This type of relationship is used when items are brought from disparate systems into a master product information repository. A hub is not used in the latter scenario.

Other Cross-References

The cross-references of all attributes, except customer and item attributes, are maintained in the order orchestration and product data repository. Use domain value maps for attributes from the order orchestration and planning data repository. Domain value maps are used during the collections process to map the values from one domain to another, which is useful when different domains represent the same data in different ways.

Product Transformation Setup: Explained

Set up product transformation to ensure that the products are converted properly when a sales order is transformed into an orchestration order.

Product transformation is executed using a combination of product relationships, product structures, transaction item attributes, and business rules. Product transformation setup consists of the following steps

1. Define items in Oracle Fusion Product Model.
2. Define rule bucket sets.
3. Set the bucket sets to the facts.
4. Create rules.

Define Items in Product Model

You must set up items and their structures and attributes in Product Model and then map them to fulfillment products.

- Define products used for product transformation rules in the product master.
- If you plan to base transformation rules on product structure, then define product structures.
- If you plan to use attribute-based rules, then define transactional item attributes.
- Define the relationship between sales product and fulfillment product.

Define Rule Bucket Sets

Create bucket sets on the Managing Product Transformation Rules page. Bucket sets contain the options that are available for selection when creating rules. Smaller bucket sets are more likely to be reused.

Create Rules

Create product transformation rules on the Managing Product Transformation Rules page.

Pretransformation Defaulting Rules: Explained

Use pretransformation defaulting rules to automatically populate specific attributes onto an orchestration order before product transformation. You can use the defaulted attribute value in the product transformation rules.

The master inventory organization automatically is defaulted to the orchestration order, so that it is available for product transformation rules.

Creating Pretransformation Defaulting Rules: Examples

Use these scenarios to understand how to use pretransformation defaulting rules.

Automatically Populating an Attribute on the Fulfillment Line

Your company receives sales orders for widgets that have an attribute called Request Date. You want this attribute to appear on all fulfillment lines for

widgets. You write a pretransformation defaulting rule that states that if the product is a widget then populate the Request Date attribute on the fulfillment line.

Automatically Populating an Attribute for Use in a Product Transformation Rule

Your company receives sales orders for widgets. You want to write a product transformation rule that converts the widget size from centimeters to inches, but you must first populate the fulfillment line with the Size attribute. You write a pretransformation defaulting rule that says that if the product is a widget then populate the fulfillment line with the Size attribute.

Product Transformation Rules: Explained

During product transformation, a sales-oriented representation of the products on a sales order is transformed to a fulfillment-oriented representation of the products on an orchestration order. Product transformation is effected using a combination of product relationships, product structures, transactional item attributes, and business rules. You create transactional item attributes and product relationships and structures in Oracle Fusion Product Model. You write rules on the Manage Product Transformation Rules page in Distributed Order Orchestration.

The following types of product transformation are supported:

- Attribute to attribute
- Attribute to product
- Product to attribute
- Product to product
- Context to attribute
- Context to product

Creating Product Transformation Rules: Examples

Use the following examples to understand the types of product transformation rules you can write.

Creating Attribute-to-Attribute Transformation Rules

Your US-based company receives sales orders from its office in Europe. The item size on the sales order line is expressed in centimeters, but you want it to appear in inches on the orchestration order line. You write an attribute-to-attribute transformation rule that transforms the transactional item attribute from the source order line to a different attribute on the orchestration order line.

Creating Attribute-to-Product Transformation Rules

Your company receives sales orders for MP3 players with various transactional item attributes, such as color and storage capacity. You want each combination

of attributes to correspond to a product number, for example, an MP3 player of color silver and storage capacity of 8 megabytes would appear as MA980LL/A on the orchestration order line. You write an attribute-to-product transformation rule transforming the attributes to a product number.

Creating Context-to-Attribute Transformation Rules

Your company manufactures laptop computers. Some are shipped to domestic locations, and others are shipped to international locations. Each type of shipping has different requirements. You write a context-to-attribute transformation rule that transforms the region, or context, on the sales order line into a packing type attribute on the orchestration order line.

Creating Context-to-Product Transformation Rules

Your company receives sales orders for laptop computers from different geographical regions. The geographical region of the order determines which adapter is included with the product. You write a context-to-product transformation rule that transforms a single sales order to an orchestration order with two lines, one of which is reserved for the region-specific adapter:

- Orchestration order line 1: laptop computer
- Orchestration order line 2: 65-watt AC adapter

Creating Product-to-Product Transformation Rules

Your company receives sales orders for camcorders that come with several accessories: Lithium-ion battery, AC adapter, editing software, packing materials. You write a product-to-product transformation rule that creates five orchestration order lines:

- Orchestration order line 1: camcorder
- Orchestration order line 2: lithium-ion battery
- Orchestration order line 3: AC adapter
- Orchestration order line 4: editing software
- Orchestration order line 5: packing materials

Creating Product-to-Attribute Transformation Rules

Sales orders contain attributes for width and height, but you want an attribute for area on the orchestration order. You write a product-to-attribute transformation rule that computes the value for area from the width and height transactional item attributes and places it on the orchestration order.

Creating Advanced Transformation Rules: Worked Example

This example demonstrates how to create an advanced transformation rule. Transformation rules are used at runtime to determine the internal representation of a product on an order line based on the information in the source order. An advanced rule can be used to compare two or more lines in an order.

This example shows you how to create a rule in which two lines are compared. If the first fulfillment line requires that you add an item and the second fulfillment line requires that you delete the same item, then the two actions cancel out one another.

1. Create the If statement: If the change in fulfillment line 1 is Add.
2. Create the If statement: If the change in fulfillment line 2 is Delete.
3. Create the If statement: If the inventory item in fulfillment line 1 is the same as the inventory item in fulfillment line 2.
4. Create the If statement: If the fulfillment line ID of fulfillment line 1 is different from the fulfillment ID of fulfillment line 2.
5. Create the Then statement: Then retract fulfillment line 1 and fulfillment line 2.

Creating an Advanced Transformation Rule

1. On the Manage Product Transformation Rules page, select Product Transformation Rules under Rulesets.
2. Click the New Rule icon.
3. Enter a rule name, such as "Consolidate Add and Delete Actions" in the Rule Name field.
4. In the Description field, type a description of the rule, such as "Rule to remove requests that cancel each other out."
5. Click the Expand icon.
6. Select the Advanced Mode check box.
7. In the If region, type a name, such as FulfillLine, in the Variable field.
8. In the Fact Type list of values, select FulfillLineVO.
9. On the next line, click the Search icon.
10. In the Condition Browser, click the Expand icon next to FulfillLine.
11. Select DeltaType.
12. Click **OK**.
13. In the Operator list of values, select Is.
14. In the Condition Browser, enter "Add."
15. Click **OK**.
16. Click the Add Pattern icon.
17. In the Variable field, enter a name, such as FulfillLine2.
18. In the Fact Type list, select FulfillLineVO.
19. Click Insert Test.
20. Click the Search icon.
21. In the Condition Browser, click the Expand icon next to FulfillLine2.
22. Select DeltaType.
23. Click **OK**.

24. In the Operator list of values, select Is.
25. Click the Search icon.
26. In the Condition Browser, click the Expand icon next to FulfillLine2.
27. In the Condition Browser field, enter "Delete."
28. Click **OK**.
29. Click the Insert Test icon.
30. Click the Search icon.
31. In the Condition Browser, click the Expand icon next to FulfillLine.
32. Select InventoryItemId.
33. Click **OK**.
34. In the Operator list of values, select Is.
35. Click the Search icon.
36. In the Condition Browser, click the Expand icon next to FulfillLine2.
37. Select InventoryItemId.
38. Click **OK**.
39. Click the Insert Test icon.
40. Click the Search icon.
41. In the Condition Browser, click the Expand icon next to FulfillLine.
42. Select FulfillLineId.
43. Click **OK**.
44. In the Operator list of values, select More Than.
45. Click the Search icon.
46. In the Condition Browser, click the Expand icon next to FulfillLine2.
47. Select FulfillLineId.
48. Click **OK**.
49. In the Then region, select Insert Action.
50. In the Action list of values, select Retract.
51. In the Target list of values, select FulfillLine.
52. Click the Insert Action icon.
53. In the Action list of values, select Retract.
54. In the Target list of values, select FulfillLine2.
55. Click **Save**.

Creating Posttransformation Defaulting Rules: Examples

Use posttransformation defaulting rules to automatically populate specific attributes onto an orchestration order based on the product transformation that

is applied to the orchestration order. Use these scenarios to understand how to use posttransformation defaulting rules.

Populating a Newly Added Orchestration Order Line with a Different Warehouse Attribute

Your company receives orders for laptop computers. Your product transformation rule transforms the sales order into an orchestration order with two lines:

- Orchestration order line 1: laptop computer
- Orchestration order line 2: alternating current adapter

You write a posttransformation defaulting rule that populates orchestration order line 2 with a warehouse that is different from the warehouse for the laptop computer.

Populating a New Orchestration Order with a New Attribute

Your company receives orders from that have the requested date as follows: MM/DD/YYYY. Your staff finds it useful to also know the day of the week because delivery options might be limited or cost more on certain days. You write a posttransformation defaulting rule that populates the day of the week onto the new orchestration order.

Glossary

activity

User task that needs to be assigned and completed as part of the order fulfillment process.

ADF

Acronym for Application Developer Framework. A set of programming principles and rules for developing software applications.

backordered

Indicates that an item is not available in sufficient quantity in inventory to satisfy immediate customer demand, but it is recorded for future fulfillment.

BPEL

Business Process Execution Language; a standard language for defining how to send XML messages to remote services, manipulate XML data structures, receive XML messages asynchronously from remote services, manage events and exceptions, define parallel sequences of execution, and undo parts of processes when exceptions occur.

continuous collection

An incremental, entity-specific method of collecting data from the Oracle Fusion source system into the order orchestration and planning data repository.

global entity

An entity for which a single value is stored for all source systems when instances of the entity are collected and stored in the order orchestration and planning data repository.

in jeopardy

Indicates that the level of risk associated with the delay of an orchestration process task is low, medium, or high.

item organization

Item definition where inventory balances are not stored and movement of inventory is not tracked in the applications. Item attributes that carry financial and accounting information are hidden.

JAR

Abbreviation for JAVA archive.

model

Fulfillment line that has child fulfillment lines.

on hold

Indicates that a hold was not released.

orchestration order

Fulfillment-oriented representation of a source sales order. When a sales order enters Oracle Fusion Distributed Order Orchestration it is transformed to an orchestration order.

orchestration process

A sequence of steps that takes one or more fulfillment lines through the fulfillment process.

order orchestration and planning data repository

The set of data collected from source systems and stored for use by order orchestration and order promising processes.

processing constraints

Rule that controls which changes can be made to an orchestration order, by whom, and when.

SOA

Abbreviation for service-oriented architecture.

source system

Any point of origin with integrated, cohesive business application data from which fulfillment data, or planning data, or both, can be extracted into data files.

source-specific entity

An entity for which a record is stored for each instance in each source system when data is collected and stored in the order orchestration and planning data repository. The stored data includes which source system the record was collected from.

staging tables upload

The final step in the extract, store in staging tables, and upload from staging tables, method of collecting data from an external source system into the order orchestration and planning data repository. The upload can be performed as a targeted or net change upload.

step

Building block of an orchestration process definition that specifies which task layer service to call or which subprocess to launch. Each step contains definitions of the task type, task, and service. Branching is also specified in the step. Tasks are made up of one or more steps.

targeted collection

A periodic, full-refresh, entity-specific method of collecting data from an Oracle Fusion source system into the order orchestration and planning data repository. A targeted collection completely replaces all data in the data repository for each entity included in the collection cycle.

task

Representation of a fulfillment task that Oracle Fusion Distributed Order Orchestration requests in a fulfillment system, for example, Invoice or Reservation. A task can have multiple orchestration process steps.

Web service upload

An incremental, entity-specific method of collecting data from an external source system into the order orchestration and planning data repository.