Oracle Fusion
Distributed Order Orchestration
Using Order Orchestration

Release 9
## Contents

**Preface**

1. **Manage Open Sales Orders**
   
   - Overview
   - Oracle Fusion Distributed Order Orchestration Business Objects: How They Fit Together
   - Searching for Oracle Fusion Distributed Order Orchestration Objects: Explained
   - Editing Fulfillment Line Attributes: Examples
   - Parallel Processing: Explained
   - Fulfillment Line Tax Information: Explained
   - Error Recovery: Explained
   - Sourcing Items from a Different Warehouse: Points to Consider
   - Resolving Fulfillment Line Exceptions: Points to Consider
   - Holds: Explained
   - Reviewing Exceptions: Examples
   - Manual Scheduling: Explained
   - Order Orchestration Work Area Analytics: Explained
   - Order Orchestration Work Area Analytic Types: Explained
   - Split Fulfillment Lines: Explained
   - Disabled Fulfillment Line Actions: Explained
   - Models: Explained
   - Orchestration Order and Orchestration Order Line Closure: Explained
   - Fulfillment Line Closure: Explained
   - FAQs for Manage Open Sales Orders
2 Process Sales Orders 22

- Overview 22
- Changes: How They Are Processed 22
- Order Capture Services: Explained 22
- Task Layer: Explained 23
- Shipment Task Layer Service: Explained 24
- Schedule Task Layer Service: Explained 25
- Reservation Task Layer Service: Explained 25
- Invoice Task Layer Service: Explained 26
- Pause Task Layer Services: Explained 27
- Holds Task Layer Service: Explained 27
- Activity Task Layer Service: Explained 27
- Return Task Layer Service: Explained 28
- Template Task Layer: Explained 29
- Fulfillment Order Task Layer Services: Explained 30
- Sales Order Transformation to Orchestration Order: Explained 31
- Activities: Explained 32
- Orchestration Process Assignment Failure: Explained 32
- FAQ for Process Sales Orders 32
3 Define Sales Order Fulfillment

Oracle Fusion Distributed Order Orchestration Components: How They Work Together 33
Orchestration Lookups: Explained 34
Orchestration Profile Management: Points to Consider 35
Oracle Fusion Distributed Order Orchestration Extensible Flexfields: Explained 36
Oracle Fusion Distributed Order Orchestration Extensible Flexfield Uses: Explained 36
Oracle Fusion Distributed Order Orchestration Extensible Flexfield Setup: Explained 37
Manage Business Event Trigger Points 40
FAQs for Manage Business Event Trigger Points 42
Manage Hold Codes 43
Manage Orchestration Source Systems 43
FAQs for Orchestration Source Systems 47
Collect Orchestration Reference and Transaction Data 47
Perform Orchestration Data Collections 60
Review Orchestration Collected Data 69
FAQs for Review Planning Collected Data 73
Define Orchestration 74
Define Processing Constraints 140
FAQs for Define Processing Constraints 146
Define Transformation Details 146
Preface

This Preface introduces information sources available to help you use Oracle Applications.

Oracle Applications Help

Use the help icon to access Oracle Applications Help in the application.

Note

If you don’t see any help icons on your page, click the Show Help button in the global area. Not all pages have help icons.

You can also access Oracle Applications Help at https://fusionhelp.oracle.com/.

Oracle Applications Guides

To find other guides for Oracle Applications, go to:

- Oracle Applications Help, and select Documentation Library from the Navigator menu.
- Oracle Help Center at http://docs.oracle.com/

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.

Oracle Enterprise Repository for Oracle Fusion Applications

Oracle Enterprise Repository for Oracle Fusion Applications (http://fusionappsoer.oracle.com) provides details on assets (such as services, integration tables, and composites) to help you manage the lifecycle of your software.

Documentation Accessibility

For information about Oracle’s commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! - Send e-mail to: oracle_fusion_applications_help_ww_grp@oracle.com. - Click your user name in the global area of Oracle Applications Help, and select Send Feedback to Oracle.
1 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
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.

Searching for Oracle Fusion Distributed Order Orchestration Objects: Explained
You can search for Oracle Fusion Distributed Order Orchestration business objects on the Order Orchestration Overview page on the Order Orchestration work area. You can save searches that you want to run frequently and promote saved fulfillment line searches to the Watchlist for quick access.

You can use two different types of searches to locate business objects:

- Basic search: Search for fulfillment lines, orchestration orders, orchestration order lines, orchestration processes, return fulfillment lines, order orchestration messages using a few attributes.
• Advanced search: Contains a greater variety of attributes that you can search on. In addition to the attributes that are displayed in the Advanced search area, you can find more, including extensible flexfields, in the Add Fields list. When you search on an attribute from the Add Fields list the attribute appears as a column in the Search Results region after the search is performed. You can use extensible flexfields when searching for fulfillment lines only. All segments in the context are displayed by default.

Using the Advanced search option, you can also search for closed fulfillment lines. The basic search doesn’t provide this option.

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.

Parallel Processing: Explained

The tasks of fulfillment lines within a group can be processed simultaneously. In parallel processing, the application can process simultaneously tasks of the same fulfillment lines or of different filtered fulfillment lines. You may want to use parallel processing when you have several fulfillment lines within a group that require some different processing tasks. For example, consider the purchase of a laptop computer with a service contract. The laptop is shipped, but the service contract is drafted and sent by e-mail. Order administrators can set up parallel processing on the Manage Orchestration Process Definition pages.

Use parallel processing when you have long-running tasks that you want to run at the same time. For example, you want to send simultaneous requests to two different fulfillment systems that generally take some time to respond. Avoid parallel processing for tasks in which a response to the request is almost immediate.

The following figure shows a partial orchestration process with parallel branches.

The first two steps (Step 100 - Schedule Merchandise and Step 200 - Create Reservation) begin the orchestration process. A parallel node follows and then the two parallel branches begin. Steps 400 and 500 are in one branch, and steps 600 and
700 are in a parallel branch. The Create Invoice and Wait for Billing steps occur after the merge step. These final steps are not shown.

Planning is run after each task is completed. However, consider that some tasks could take several days. Be aware that your planning data will not be up to date as the orchestration process waits for a long-running task to be completed before the branch is replanned. If you need more accurate planning data, then schedule your Plan Orchestration Processes scheduled process, which replans the entire orchestration process, to run more often. For example, you might want to schedule the planning scheduled process to run daily.

**Note**
You can’t split a task that is in a parallel branch. If you or a user of a fulfillment system tries to split a task that is in a parallel branch, then the action will be rejected automatically.
Fulfillment Line Tax Information: Explained

You can view tax information at the fulfillment line level. The tax information from the sales order is listed automatically on the associated fulfillment line when the sales order enters Oracle Fusion Distributed Order Orchestration. During processing, the tax information is transferred to downstream fulfillment systems. Those systems can make updates and send the information back to Distributed Order Orchestration.

You cannot change tax information in Distributed Order Orchestration, but be aware that certain actions in Distributed Order Orchestration could result in a change to the tax information that you see. For example, if you split a line manually or it splits automatically, then a new warehouse and its associated information may be listed automatically. That warehouse may be associated with different tax information than the previous warehouse.

An order administrator can create processing constraints using tax attributes and designate a tax attribute as an order attribute that identifies change (change attribute).

Error Recovery: Explained

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

The following errors might occur during order processing:

- Fulfillment lines in an order are not assigned to an orchestration process.
- An orchestration process did not start, or an orchestration process or task resulted in an error.
- An error occurred during internal processing of an orchestration process step.
- An error occurred when sending a task request to a fulfillment system.

You can use the following action 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. You must correct the root cause of the error before you use this action. This action is available only when an error is associated with an orchestration order. You can simultaneously select one or more orders for recovery. This action attempts to recover any orchestration processes or tasks that are in an error state for the orders that you select. It reruns the order from immediately before the point where the error occurred.
- 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. The action attempts to recover any internal errors or tasks that are in an error state for the selected processes. You can use Recover Process to recover all the tasks that a process contains. It reruns the process from immediately before the point where the error occurred.
- Recover Task: Use this action to correct an error that results from an attempt to send a request to a fulfillment system. The action is available only when an outbound request fails. To simultaneously select multiple tasks for recovery, select multiple orchestration processes or orchestration orders, and then use Recover Process or Recover Order. It reruns the task from immediately before the point where the error occurred.

You must correct some errors outside of Order Orchestration. For example:

- An error that occurs in a source order that fail during decomposition. To correct this error, you must resubmit the order.
- An error that occurs during processing of a reply from a fulfillment system. To correct this error, you must resubmit the reply.
• An error that Oracle Enterprise Manager identifies as recoverable. To correct this error, you must use Oracle Enterprise Manager to recover it.

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.

<table>
<thead>
<tr>
<th>Option</th>
<th>What Does It Do?</th>
<th>Why Use It?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release holds</td>
<td>Releases a hold on a fulfillment line. You can release a hold that was released.</td>
<td>If a fulfillment line has a hold on some part of it, then it may be released.</td>
</tr>
<tr>
<td>Option</td>
<td>What Does It Do?</td>
<td>Why Use It?</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>applied in Oracle Fusion Distributed Order Orchestration only.</td>
<td>unable to proceed through the fulfillment process. A hold that is specific to some part of the orchestration process can allow a fulfillment line to proceed but may arrive at a point at which it will stop and wait for a user to release it.</td>
</tr>
<tr>
<td>Reschedule using Check Availability</td>
<td>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. If the fulfillment line already is scheduled and your selection results in a changed scheduled ship date, then change processing is triggered.</td>
<td>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.</td>
</tr>
<tr>
<td>Override the schedule</td>
<td>Overrides the scheduled ship date or scheduled arrival date assigned by the order promising process. If you change the scheduled ship date after the fulfillment line is scheduled, then change processing is triggered.</td>
<td>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.</td>
</tr>
<tr>
<td>Change the warehouse</td>
<td>Lets you change the warehouse that sources the item on the selected fulfillment lines.</td>
<td>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.</td>
</tr>
<tr>
<td>Change the demand class</td>
<td>Determines priority when scheduling the fulfillment line.</td>
<td>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.</td>
</tr>
<tr>
<td>Option</td>
<td>What Does It Do?</td>
<td>Why Use It?</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Change the shipping method</td>
<td>Lets you change the shipping method for all the selected fulfillment lines to a more expeditious method.</td>
<td>You know the shipping method that you want to use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can change the shipping method of multiple fulfillment lines.</td>
</tr>
<tr>
<td>Clear selected fulfillment line</td>
<td>Gives the order promising process the freedom to choose the best fulfillment option, without regard to a specific attribute.</td>
<td>You want to use Check Availability to search for the best possible fulfillment options, without having to consider some fulfillment requirements.</td>
</tr>
<tr>
<td>attributes</td>
<td></td>
<td>You can clear the attributes of multiple fulfillment lines.</td>
</tr>
<tr>
<td>Substitute the item on the</td>
<td>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.</td>
<td>You see that an allowable substitute will provide you with a better scheduling option.</td>
</tr>
<tr>
<td>fulfillment line</td>
<td></td>
<td>The analytics provide information that may help you determine the item to substitute.</td>
</tr>
<tr>
<td>Split the fulfillment line</td>
<td>Splits the selected fulfillment line into two or more fulfillment lines.</td>
<td>You know that you can fulfill part of the fulfillment line from one warehouse and part from another warehouse.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The analytics provide information that may help you determine the warehouse to source from.</td>
</tr>
<tr>
<td>Unschedule</td>
<td>Releases supply so that it is available for another fulfillment line. It invokes change management which brings the process back to the scheduling step and in Awaiting Scheduling status.</td>
<td>You want to release supply to make it available for a high-priority fulfillment line.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You want to reschedule a fulfillment line from the Order Orchestration work area.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Makes an initial request to the order promising process to schedule a fulfillment line.</td>
<td>The fulfillment line is associated with an orchestration process that has a manual scheduling step, or the fulfillment line was unscheduled.</td>
</tr>
<tr>
<td>Option</td>
<td>What Does It Do?</td>
<td>Why Use It?</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Unreserve</td>
<td>Releases supply that is earmarked for a specific fulfillment line, so that it is available for another fulfillment line.</td>
<td>You want to release supply reserved for a specific line to make it available for a high-priority fulfillment line.</td>
</tr>
<tr>
<td>Reserve</td>
<td>Removes supply from availability and earmarks it for a specific fulfillment line.</td>
<td>You want to ensure that supply is allocated for a particular fulfillment line.</td>
</tr>
</tbody>
</table>

Related Topics

- Changes: How They Are Processed

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:

- Unscheduling a fulfillment line
- Scheduling shipment sets and models

### Unscheduling a Fulfillment Line

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

### Scheduling Shipment Sets and Models

The manual scheduling options apply to configured items, models, and shipment sets. If you select just one fulfillment line in one of these groupings, then the scheduling 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. Analytics about customers use the customer party ID for the query.

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: \( \frac{(\text{system date}) - (\text{ordered date for open orders})}{(\text{last updated date}) - (\text{ordered date for closed orders})} \). 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: \( \frac{(\text{system date}) - (\text{ordered date for open orders})}{(\text{last updated date}) - (\text{ordered date for closed orders})} \). 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.

### Order Orchestration Work Area Analytic Types: Explained

The Order Orchestration work area analytics use several underlying technologies. You might need to know the technology of a report to troubleshoot issues with it. The following table lists the types of reports in the Order Orchestration work area and the underlying technology of each.

<table>
<thead>
<tr>
<th>Report Name</th>
<th>Page</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fulfillment Lines by Exception</td>
<td>Overview</td>
<td>OLTP</td>
</tr>
<tr>
<td>Fulfillment Lines by Status</td>
<td>Overview</td>
<td>OLTP</td>
</tr>
<tr>
<td>On-Time Shipment Percentage by Time</td>
<td>Overview</td>
<td>Oracle Business Intelligence Applications</td>
</tr>
<tr>
<td>Item: Item Availability</td>
<td>Manage Fulfillment Line Exceptions</td>
<td>OLTP</td>
</tr>
<tr>
<td>Item: Order Frequency by Time</td>
<td>Manage Fulfillment Line Exceptions</td>
<td>Oracle Business Intelligence Applications</td>
</tr>
<tr>
<td>Item: Count of Holds for This Item by Hold Name</td>
<td>Manage Fulfillment Line Exceptions</td>
<td>Oracle Transactional Business Intelligence</td>
</tr>
<tr>
<td>Customer: Order Age by Time</td>
<td>Manage Fulfillment Line Exceptions</td>
<td>Oracle Business Intelligence Applications</td>
</tr>
<tr>
<td>Customer: Order Value by Time</td>
<td>Manage Fulfillment Line Exceptions</td>
<td>Oracle Business Intelligence Analytics</td>
</tr>
<tr>
<td>Customer: Open Orchestration Orders for This Customer by Age</td>
<td>Manage Fulfillment Line Exceptions</td>
<td>OLTP</td>
</tr>
<tr>
<td>Customer: Count of Holds for This Customer by Type</td>
<td>Manage Fulfillment Line Exceptions</td>
<td>Oracle Transactional Business Intelligence</td>
</tr>
<tr>
<td>Order: Fulfillment Line Exceptions for This Order</td>
<td>Manage Fulfillment Line Exceptions</td>
<td>OLTP</td>
</tr>
</tbody>
</table>
### 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 the original 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
- Visibility into quantity

**Manual Splits**

In the Order Orchestration work area, you can split fulfillment lines using the Split Fulfillment Line action only on a manual schedule task. Change management is not triggered for manual splits. You can't explicitly split a fulfillment line manually from the Check Availability page. The availability option you select, however, may implicitly split the fulfillment line.

If you want to split a fulfillment line that is not on a manual schedule task, then you can click the **Unschedule** button while an orchestration process is running. This action takes the orchestration process back to the schedule task, which now is manual. Then you can 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 and which have a quantity greater than one 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.

Even if a fulfillment line can be split as indicated previously, you can't split a fulfillment line if it is on a task in a parallel branch.

**Effect on Shipment Sets**

You can't split fulfillment lines within shipment sets. You can split a line within a shipment set only if you remove the line from the shipment set before splitting, 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.

**Visibility into Quantity**

View the original ordered quantity on the orchestration order line. You can see the quantity on each fulfillment line after the split on the Manage Fulfillment Lines page.
### 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.

<table>
<thead>
<tr>
<th>Action</th>
<th>Condition Under Which Action Is Disabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>All fulfillment line actions except view</td>
<td>User request status is processing.</td>
</tr>
<tr>
<td>All fulfillment line actions except view</td>
<td>The fulfillment line is locked.</td>
</tr>
<tr>
<td>All fulfillment line actions except view</td>
<td>The fulfillment line is canceled.</td>
</tr>
<tr>
<td>All fulfillment line actions except view</td>
<td>The fulfillment line is closed.</td>
</tr>
<tr>
<td>All fulfillment line actions except view</td>
<td>The fulfillment line was shipped.</td>
</tr>
<tr>
<td>All fulfillment line actions except view</td>
<td>An orchestration process was not assigned.</td>
</tr>
<tr>
<td>All fulfillment line actions except view</td>
<td>The fulfillment line belongs to a model.</td>
</tr>
<tr>
<td>Unschedule</td>
<td>The fulfillment line is not scheduled. Scheduled ship date is null.</td>
</tr>
<tr>
<td>Reserve</td>
<td>The fulfillment line is not scheduled. Scheduled ship date is null.</td>
</tr>
<tr>
<td>Reserve</td>
<td>The fulfillment line or any associated child fulfillment line already was reserved. Reserved quantity is not null.</td>
</tr>
<tr>
<td>Reserve</td>
<td>A child fulfillment line is reserved.</td>
</tr>
<tr>
<td>Action</td>
<td>Condition Under Which Action Is Disabled</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Unreserve</td>
<td>Neither the fulfillment line nor any associated child fulfillment line was reserved. Reserved quantity is null.</td>
</tr>
<tr>
<td>Substitute Item</td>
<td>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.</td>
</tr>
<tr>
<td>Substitute Item</td>
<td>The fulfillment line does not allow for substitution ( Substitute Allowed = False ), and you do not have the privilege to override the restriction.</td>
</tr>
<tr>
<td>Substitute Item</td>
<td>The fulfillment line is a model.</td>
</tr>
<tr>
<td>Substitute Item</td>
<td>The fulfillment line belongs to a shipment set.</td>
</tr>
<tr>
<td>Release Hold</td>
<td>The fulfillment line is not on hold.</td>
</tr>
<tr>
<td>Split Fulfillment Line</td>
<td>The fulfillment line belongs to a shipment set.</td>
</tr>
<tr>
<td>Split Fulfillment Line</td>
<td>The fulfillment line does not allow for partial shipment ( Split Allowed = False ) and the user does not have the privilege to override.</td>
</tr>
<tr>
<td>Split Fulfillment Line</td>
<td>More than one fulfillment line is selected.</td>
</tr>
</tbody>
</table>

**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.

Orchestration Order and Orchestration Order Line Closure: Explained

Orchestration orders and orchestration order lines remain open after they are fulfilled until they are closed by a scheduled process. The statuses of the orchestration orders and orchestration order lines are updated after the scheduled process closes them. This functionality gives you more control over determining when your organization considers its orders closed.

The scheduled process closes all eligible lines and then closes the order if the order entity is chosen as the parameter.

Creating the Scheduled Process to Close Orders and Lines

You create your closure requests on the Overview page of the Scheduled Processes work area. Click the Schedule New Process tab, select the Update or Close Orchestration Orders process, and create a new request.

The following table explains the parameters for creating the scheduled process:
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity Name</td>
<td>Required field. Enter Header for orchestration order and Line for orchestration order line.</td>
</tr>
<tr>
<td>Entity ID</td>
<td>If you do not enter a start date, end date, or interval hours, then you must enter an entity ID. You can run a query to obtain the entity ID.</td>
</tr>
<tr>
<td>Start Date</td>
<td>Earliest creation date and time of the entity. If you enter interval hours, then do not enter a start date.</td>
</tr>
<tr>
<td>End Date</td>
<td>Latest creation date and time of the entity. If you enter interval hours, then do not enter an end date.</td>
</tr>
<tr>
<td>Interval Hours</td>
<td>Period of time in hours counted backwards from the time when the process is initiated. It is used to determine the earliest and latest creation date and time of the entity.</td>
</tr>
</tbody>
</table>

- The latest creation date and time is the date and the time when the process is initiated. For example, if the process is scheduled to start at 4 PM tomorrow, then the latest creation date and time is 4 PM tomorrow.
- The earliest creation date and time is the latest creation date and time minus the interval hours. If you enter 4 as the interval hours and the process is scheduled to start at 4 PM tomorrow, then the earliest creation date and time is 12 PM tomorrow.

**Fulfillment Line Closure : Explained**

A fulfillment line is considered closed when all steps of the assigned orchestration process have been completed. You may see an indication that fulfillment is complete at one of the steps of the orchestration process, such as shipping complete. However, this indicator simply means that your organization considers fulfillment complete when the process reaches this step. The fulfillment line is not actually closed at this point, though.

Note that orchestration orders and orchestration order lines may appear open even when all associated fulfillment lines are closed, if the Update or Close Orchestration Orders scheduled process wasn’t run since fulfillment line closure. You can set up the scheduled process in the Scheduled Processes work area.
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.
2 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 adjusted. 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 adjusted sometimes are canceled first and then redone.

An event is raised when compensation is completed or if an error occurs during compensation.

Order Capture Services: Explained

Order Capture services receive requests from integrated order capture applications and transform the requests into a structure that is recognized by Oracle Fusion Distributed Order Orchestration. The requests are for information about fulfillment lines.

The Order Capture services perform the following:

- Get order details.
- Get order shipping details.
- Apply a hold.
- Release a hold.
- Check for availability.
The following services request information from order capture applications and transform them into a structure that can be successfully fulfilled through multiple fulfillment systems.

### Get Order Details

This service communicates status information or order details back to the order capture application that requests it. The service can provide information about the entire sales order or a desired set of order lines.

You can get details about an order that originated in any other integrated source system. You can define a source system, which constrains the query to that system. Alternatively, you can choose not to specify a source system, in which case matching order numbers from any source system are returned.

### Get Order Shipping Details

This service communicates shipping details back to the order capture application that requests it. The response in this service is organized by shipments, rather than by fulfillment lines. The shipping details provided by this service are: current schedules, status of each schedule, and schedule details such as warehouse and shipping method. Because each sales order line in order capture may have multiple related fulfillment lines in Distributed Order Orchestration the response to order capture contains information about all the lines. The service can return details of all the sales order lines of a sales order or for a specific set of sales order lines.

### Apply a Hold

This service routes a request for a hold from an order capture system to the Hold task layer. This service allows multiple holds to be requested, each for a single sales order or for multiple sales order lines in that order.

### Release a Hold

This service routes a request to release an existing hold from an order capture system to the Hold task layer.

### Check Availability

This service sends requests to Distributed Order Orchestration inquiring about the availability of product for a given date and source organization. Distributed Order Orchestration responds to these requests by sending the order capture application detailed information about the product, available quantity, and source organization.

### 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
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

Pause Task Layer Services: Explained

The Pause task layer services pause an orchestration process until a specified condition is met. Pause task layer services can be used to create a waiting period between tasks and to define when the pause is released and the next task of the orchestration process is allowed to begin.

You might want to use a Pause task in scenarios such as:

• Layaway Orders: Pause the orchestration process until the customer finishes paying for the item. The release condition, known as the Start-After Condition, is configured to release the pause when payment is complete.

• Pre-Orders: Pause an orchestration process until a specified date before releasing pre-orders. The Start-After Condition is configured to release the pause when the date arrives.

• Shipping System That Cannot Handle Early Requests: Shipping system can handle requests for orders that are scheduled to ship within only 2 days. Pause the orchestration process so that requests are not sent to the shipping system until 2 days before the ship date.

• Conditional Manual Review: If the scheduled date is beyond the requested date, then conditionally wait for manual review and approval.

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 an 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.

For more information about implementing the template task layer, see DOO: Oracle Fusion Distributed Order Orchestration White Papers (1536633.1) on My Oracle Support at https://support.oracle.com.

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 Services: Explained
Fulfillment order task layer services are designed to enable Oracle Fusion Distributed Order Orchestration to integrate with an Enterprise Resource Planning (ERP) system. In most ERP systems, fulfillment tasks are closely coupled with an order management system and open-ended interfaces through which Distributed Order Orchestration can connect are not available. With the fulfillment order task layer service, you can leverage existing ERP systems.

The fulfillment order task layer services primarily support the following functions:

- Sends fulfillment requests to an ERP system.
- Aggregates fulfillment lines from a single order across orchestration groups and sends them in a single request.
- Receives updates from downstream ERP systems as the fulfillment activities are executed.

Send Fulfillment Requests to an ERP System
Fulfillment order task layer services send fulfillment requests to the ERP system. Requests are for: Creating, updating, or canceling an order in the downstream ERP order management system; placing or releasing a hold on an order in the ERP system; and updating the status of an order. The fulfillment order task layer sends updates to the ERP system every time a change order is accepted that affects the fulfillment of the orchestration order.

Aggregate Requests to the ERP System
Fulfillment order task layer services aggregate and store requests that are bound for the ERP system. You express the aggregation criteria as a rule. You can base the criteria on a time limit or the total number of lines that must be aggregated based on implementation-specific conditions. After the aggregator receives some requests, the aggregator publishes the information from each related request as a single aggregated request. Aggregating the requests minimizes issues with the timing of the requests.

The aggregator uses a combination of timeout and estimated number of fulfillment lines within an orchestration order to determine when to send the single aggregated request. The timeout determines how long the aggregator must wait before
it sends whatever it has currently aggregated. When the time expires, the aggregator publishes whatever requests have been accumulated. If an order has several fulfillment lines and the lines complete the task before the timeout occurs, then all requests are sent as soon as the task is completed.

The aggregation is limited to once per system per order. After fulfillment lines are aggregated to a system, no further aggregation is done to the requests that are delivered to the system subsequently. The lines are sent individually.

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

Note that when you use the aggregator for external interface routing rules you can use one or both of the following parameters:

- AGGREGATOR_MAX_FLINES: The external fulfillment system is called only after the number of pending requests exceeds the number specified.
- AGGREGATOR_WAIT_TIME: The external fulfillment system is called only after time elapsed from the first request processed exceeds the wait time.

When you specify both attributes, the external system is called when either of the conditions is met.

**Receive Status Updates from an ERP System**

Fulfillment order task layer services receive interim and final status updates from an ERP system. The responses are not delivered right away. They are delivered as the fulfillment activities are executed.

**Sales Order Transformation to Orchestration Order: Explained**

After an original sales order or change order enters Oracle Fusion Distributed Order Orchestration, the order is decomposed and transformed, using business rules, into an orchestration order. The orchestration order and orchestration order lines that result from the transformation do not necessarily mirror the original sales order and sales order lines. For example, the number of orchestration order lines may exceed sales order lines, or orchestration order lines may appear in a different sequence.

The differences between sales orders and orchestration orders arise because orchestration orders are configured for fulfillment requirements, while sales orders are configured for sales requirements. The following figure shows an orchestration order that has one more line than the sales order. The orchestration order line contains a region-specific AC adapter, which is not relevant to the sales order.

![Diagram of Sales Order Transformation](image)

In some cases, a sales order might not be transformed to an orchestration order. Transformation can 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.

FAQ for Process Sales Orders

Why doesn't the status icon match what I searched for?

You might find that when you search for the results of a mass action by querying by user request number and status that the results are not as you expect because the request might not have completed on some fulfillment lines. For example, you might search for user request number 123 and user request status Completed. The search results will show the fulfillment lines that are completed, but you might see the Processing user request status icon on some of the fulfillment lines because a later user request is processing these fulfillment lines.

How can I release a Pause task?

Pause tasks can be released manually or automatically so that the orchestration process can resume. A scheduled process can release all pauses automatically, so that the orchestration process can resume. To release a Pause task manually, you must have the Order Orchestration Error Recovery Manager role. Click the Release Pause Task button on the Orchestration Process details page.

The Release Pause Tasks scheduled process and Release Pause public service can be set up to release a Pause task automatically.
3 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 create categories and contexts, associate contexts to categories and pages, deploy the flexfields, synchronize the SOA artifacts, and synchronize the extensible flexfields with business rules. You can set up extensible flexfields throughout Distributed Order Orchestration. For example, you can use extensible flexfields for rules and processing constraints, and you can add them to enterprise business objects for integration.
The specific steps to set up extensible flexfields follow:

1. Create predefined categories.
2. Create contexts.
3. Associate contexts to categories.
4. Define pages and associate contexts to pages.
5. Deploy flexfields.
6. Synchronize the SOA artifacts.
7. Publish extensible flexfield artifacts into Oracle Business Rules in Oracle Metadata Services Repository and processing constraints framework in Distributed Order Orchestration.

For details about receiving extensible flexfields from an order capture application, using extensible flexfields in business rules, and sending extensible flexfields to fulfillment applications, see "Oracle Fusion Distributed Order Orchestration: Extensible Flexfield Usage" in Oracle Fusion Distributed Order Orchestration White Papers (Doc ID 1536633.1) on My Oracle Support.

An overview of the extensible flexfield setup steps follows. For more information, see the white paper referenced above.

### Create Predefined Categories

To create predefined categories, you must run the Publish Extensible Flexfields scheduled process using available extensible flexfields. You create the scheduled process in the Scheduled Processes work area. Verify category creation on the Manage Extensible Flexfields pages of the Setup and Maintenance work area.

The following table lists the extensible flexfields that are available.

<table>
<thead>
<tr>
<th>Distributed Order Orchestration Entity Name</th>
<th>Category Name</th>
<th>Category Hierarchy Available?</th>
<th>Preconfigured Context Values?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td>DOO_HEADERS_ADD_INFO</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Lines</td>
<td>DOO_LINES_ADD_INFO</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>FulfillLines</td>
<td>DOO_FULFILL_LINES_ADD_INFO</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Payments</td>
<td>DOO_PAYMENTS_ADD_INFO</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PriceAdjustments</td>
<td>DOO_PRICE_ADJUSTMENTS_ADD_INFO</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Distributed Order Orchestration Entity Name</td>
<td>Category Name</td>
<td>Category Hierarchy Available?</td>
<td>Preconfigured Context Values?</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------</td>
<td>-------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>SalesCredits</td>
<td>DOO_SALES_CREDITS_ADD_INFO</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>OrchestrationTaskActivity</td>
<td>DOO_ACTIVITIES_ADD_INFO</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>FulfillLineDetails</td>
<td>DOO_FULFILL_LINE_DTLS_ADD_INFO</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LotSerialNumber</td>
<td>DOO_LOT_SERIAL_NUM_ADD_INFO</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Categories are created only for the predefined ones listed in the table above. You can’t create categories other than the predefined ones.

Create Contexts
Define contexts on the Edit Extensible Flexfield page. A context determines which attributes are shown based on criteria that you create. You can also think of contexts as part of a hierarchy of attributes. The top level of the hierarchy is the entity and entity category. The next one or more levels are one or more contexts, and the final level consists of the attributes themselves.

Associate Contexts to Categories
Associate contexts with categories on the Categories: Details region of the Edit Extensible Flexfield page.

Define Pages and Associate Contexts to Pages
Associate contexts to pages of the Distributed Order Orchestration user interface using the setups on the Pages tab of the Categories: Details region of the Edit Extensible Flexfield page.

Deploy Flexfields
Deploy each flexfield individually on the Manage Extensible Flexfields page.

Synchronize the SOA Artifacts
Synchronize the SOA infrastructure with the new extensible flexfield artifacts by executing the UpdateSOAMDS composite. Choose the updateDuring operation, and specify the appropriate parameter. This action makes the newly created extensible flexfields available for mapping from the order capture application to Distributed Order Orchestration.

Publish Extensible Flexfield Artifacts into Oracle Business Rules
Publish extensible flexfield artifacts into Oracle Business Rules in Oracle Metadata Services Repository and processing constraints framework in Distributed Order Orchestration. Publishing is not required for orchestration process rules, but it is required for all other rule types. To publish the artifacts, run the Publish Extensible Flexfield Attributes scheduled process after you deploy the flexfields. This scheduled process is the same one that you ran earlier to create the extensible...
Manage Business Event Trigger Points

Oracle Fusion Distributed Order Orchestration Business Events: Explained

Oracle Fusion Distributed Order Orchestration business events notify a subscriber of a business occurrence, such as a change to an orchestration order or a fulfillment line. Business events inform users when:

- A change to an orchestration order attribute occurs
- A hold is applied
- A change to jeopardy priority occurs
- A fulfillment line is split
- A change to an order orchestration header status
- A fulfillment line is closed
- Compensation of an orchestration process is completed
- A fulfillment line achieves a particular status

You can customize all the business events, except for the compensation and fulfillment line status events, on the Manage Business Event Trigger Points page in the Setup and Maintenance work area. You can also configure an event to be raised when a change occurs to an extensible flexfield attribute. When extensible flexfields are configured, they first appear in the “Select and Add: Attributes That Trigger Events” dialog. After you select one or more segments and click OK or Apply, the selected segments appear in the “Orchestration Order Attribute Update: Attributes That Trigger Events” region.

Where you receive and view business events depends on how your organization configures it.

Business Event Trigger Points: Explained

Business event trigger points are the conditions that cause a business event to be raised. An order administrator can configure business event trigger points on the Manage Business Event Trigger Points page in the Setup and Maintenance work area. All the business events on this page are inactive by default, except for the event for fulfillment line closures. Changes that you make to this page take effect immediately. The settings affect one instance only. You cannot add or delete the business events on this page.

Trigger points can affect system performance, so carefully choose the ones that you need.

Hold Business Events: Explained

You can configure a business event to be raised when the order capture application or an Order Orchestration work area user requests a hold to an orchestration order, orchestration order line, or fulfillment line. Set up the trigger point for the hold business event on the Manage Business Event Trigger Points page.
If a hold is applied that is not immediately effective (if it applies to a future task, for example), then a business event is raised when the hold becomes effective, not at the time of the request.

An event is not raised when a hold is released or is applied by change management.

Business Event Routing Rule Setup: Explained

Business events are consumed by applications that are integrated with Oracle Fusion Distributed Order Orchestration, such as order capture and fulfillment applications. To notify the consuming applications when a business event is raised, you must define the appropriate routing rules by editing the mediator file BusinessEventNotificationMediator.mplan in Oracle JDeveloper.

Oracle Mediator is a service component of the Oracle SOA Suite that facilitates integration between events and services, where service invocations and events can be mixed and matched. A mediator provides a lightweight framework to mediate between various components within a composite application, such as business processes and human workflows, using a Web Services Description Language (WSDL) document as the interface. A mediator converts data to facilitate communication between different interfaces exposed by different components that are wired to build an SOA composite application. You can use a mediator service component to consume a business event or receive a service invocation. A mediator service component can evaluate routing rules, perform transformations, validate, and either invoke another service or raise another business event. In this case, it is used to invoke another service.

General Setup Steps

Business event setup consists of the steps below.

1. Import a composite in a project.
2. Create a web service that consumes the business event.
3. Create a partner link to the service that you created in step 2.
4. Specify a mapper file (XSL file) to transform the event payload to the consuming web service that you created in step 2.
5. Create the routing rules by modifying BusinessEventNotificationMediator.mplan.

For more information about importing a composite and specifying a mapper file, see "Oracle Fusion Distributed Order Orchestration White Papers" (1536633.1) on My Oracle Support at https://support.oracle.com.

Creating a Business Event Routing Rule: Worked Example

This example shows how to create a routing rule for the order status update business event in Oracle Fusion Distributed Order Orchestration.

You will modify the BusinessEventNotificationMediator.mplan file in Oracle JDeveloper.

Prerequisites

This example assumes that you:

1. Imported the composite DooDecompositionOrderStatusUpdateComposite SAR in a project.
2. Created a web service called EventConsumptionService.
3. Created a partner link to the web service in DooDecompositionOrderStatusUpdateComposite (composite.xml).
4. Created a transformation file called MediatorInput_To_process.xsl to map the event payload to the consuming web service.

Creating the Business Event Routing Rule

1. Open Oracle JDeveloper in Customized mode.
2. Under SOA Content, open BusinessEventNotificationMediator.mplan.
3. In the Routing Rules region, click the Add icon for orderStatusUpdateEvent.
4. Select Static Routing Rule.
5. When the Target Type dialog box appears, click Service.
6. In the Target Services window, select the target service that you defined earlier.
7. Select the appropriate port.
8. Click OK.
9. In the Static Routing subregion of the Routing Rules region, select the mapper file that you created earlier.

Next Steps
After you create the routing rules, you are ready to:

1. Build the updated composite.
2. Deploy the updated composite.
3. Verify the routing rule. After an event is triggered, log into Enterprise Manager and view the composite instance.

FAQs for Manage Business Event Trigger Points

Where do the attributes that trigger events come from?
The attributes on the Orchestration Order Update: Attributes That Trigger Events region are predefined. You can add attributes by configuring extensible flexfields.

What happens if multiple attributes enabled for the orchestration order attribute update business event are updated at the same time?
Only one notification is sent with consolidated information about all the enabled attributes that were updated.

Where do the status values that trigger the Orchestration Order Header Status Update business event come from?
The status values on the Orchestration Order Header Status Update: Status Values That Trigger Events region are predefined. You cannot add status values to the page.
How is the condition for the Jeopardy business event evaluated?

Upon replanning, the jeopardy score may change. If the change in score results in a change in priority, then an event is triggered, if the new jeopardy priority value has been enabled.

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.

Related Topics

- Holds Task Layer Service: Explained

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.

Related Topics
• Enabling Organizations for ATP Web Service : Points to Consider

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.

Note
You cannot change the version of a source system from Others to Fusion. You must delete the planning source system definition by scheduling the Delete Source Configuration and All Related Data process. The Delete Source Configuration and All Related Data process performs multiple steps. First the process deletes all data previously collected from the source system. After deleting the collected data, the process deletes the planning source system definition and collection parameters. After the Delete Source Configuration and All Related Data process completes, you must redefine the planning source system definition on the Manage Planning Source Systems page.

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>What the Parameter Does</th>
<th>A Typical Value for the Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Database Connections</td>
<td>Defines the maximum number of database connections the source server can create during the collection process. This controls</td>
<td>10</td>
</tr>
</tbody>
</table>
### Define Sales Order Fulfillment

<table>
<thead>
<tr>
<th>Parameter</th>
<th>What the Parameter Does</th>
<th>A Typical Value for the Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Parallel Workers</td>
<td>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.</td>
<td>30</td>
</tr>
<tr>
<td>Number of Rows per Processing Batch</td>
<td>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.</td>
<td>10,000</td>
</tr>
<tr>
<td>Cached Data Entries in Thousands</td>
<td>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.</td>
<td>10,000</td>
</tr>
</tbody>
</table>

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 perform additional processing steps to check for and to cross-reference data during data collections. You must enable cross-referencing for Order Capture source systems.

**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.

Related Topics

- Refreshing the Global Order Promising Engine: Explained

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 customer data managed in the Oracle Fusion Trading Community Model and to product 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 these 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
• 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 categories of data are 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

For a specific collection, the data that is collected is determined by the data collection entities specified for that collection.

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

Data Collected for Order Orchestration
The following reference data is collected and stored to support order orchestration:

• Warehouses (Organization Parameters)
• Units of Measure
• Carriers
• Currencies
• Currency Conversions
• Shipping Methods
• Order Orchestration Reference Objects
  • Payment Term
  • Accounting Rule
  • Invoicing Rule
  • Sales Credit Type
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
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 use 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.

### 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.
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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>What the Parameter Does</th>
<th>A Typical Value for the Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Database Connections</td>
<td>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.</td>
<td>10</td>
</tr>
<tr>
<td>Number of Parallel Workers</td>
<td>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.</td>
<td>30</td>
</tr>
<tr>
<td>Cached Data Entries in Thousands</td>
<td>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.</td>
<td>10,000</td>
</tr>
</tbody>
</table>

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 a global entity.

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.

<table>
<thead>
<tr>
<th>Source System</th>
<th>Collections Entity</th>
<th>Source Value</th>
<th>Target Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System A</td>
<td>Unit of measure</td>
<td>kilogram</td>
<td>kg</td>
</tr>
<tr>
<td>System B</td>
<td>Unit of measure</td>
<td>k.g.</td>
<td>kg</td>
</tr>
</tbody>
</table>

Cross-Reference Implementation

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

1. Decide which business object to enable for cross-referencing.
2. For each object, work with business analyst to decide which values to map to which other values.
3. Update the domain value maps using one of two methods: the Oracle Fusion SOA Composer or the Oracle Fusion Middleware Domain Value Map user interface. The SOA Composer is typically considered easier to use.
4. On the Manage Planning Data Collection Processes page, enable the corresponding entity for cross-referencing.
5. Determine an ongoing procedure for adding new values into the domain value map when new values occur for a business object.

Notes regarding usage of the Oracle Fusion SOA Composer:

- Depending upon how the SOA suite is installed at your location, you should have the SOA Designer role assigned in order to edit the data value maps.
- Choose DVM in MscCollConfigurationXReferenceComposite.
- You must save your changes and commit your changes.
- The APS Weblogic server must be bounced before the changes are visible to the collections engine.
- For documentation regarding using the SOA Composer, refer to the Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite document, Using Oracle SOA Composer with Domain Value Maps section.

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, PerformOdiStagingLoad. 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.
   - 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.

<table>
<thead>
<tr>
<th>Business Entity</th>
<th>Interface ODI Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work-in-Process Requirements</td>
<td>WIP_COMP DEMANDS SCEN</td>
</tr>
<tr>
<td></td>
<td>WIP_OP RESOURCE SCEN</td>
</tr>
<tr>
<td></td>
<td>CALENDAR SCEN</td>
</tr>
<tr>
<td></td>
<td>CALENDAR WORKDAYS SCEN</td>
</tr>
<tr>
<td></td>
<td>CALENDAR DATES SCEN</td>
</tr>
<tr>
<td></td>
<td>CALENDAR_EXCEPTIONS SCEN</td>
</tr>
<tr>
<td></td>
<td>CALENDARSHIFTS SCEN</td>
</tr>
<tr>
<td></td>
<td>CALENDAR PERIODSTARTDAYS SCEN</td>
</tr>
<tr>
<td></td>
<td>CALENDAR WEEKSTARTDAY SCEN</td>
</tr>
<tr>
<td></td>
<td>CALENDAR ASSIGNMENTS SCEN</td>
</tr>
<tr>
<td>Demand Classes</td>
<td>DEMAND CLASS SCEN</td>
</tr>
<tr>
<td>Global Supplier Capacities</td>
<td>GLOBAL SUP CAPACITIES SCEN</td>
</tr>
<tr>
<td>Interorganization Shipment Methods</td>
<td>SHIPMENT METHODS SCEN</td>
</tr>
<tr>
<td><strong>Business Entity</strong></td>
<td><strong>Interface ODI Scenarios</strong></td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Item Cost</td>
<td>ITEM_COST_SCEN</td>
</tr>
<tr>
<td>Item Substitutes</td>
<td>ITEM_SUBSTITUTES_SCEN</td>
</tr>
<tr>
<td>Item Suppliers (Approved Supplier List)</td>
<td>ITEM_SUPPLIERS_SCEN</td>
</tr>
<tr>
<td>On Hand</td>
<td>ONHAND_SCEN</td>
</tr>
<tr>
<td>Organizations</td>
<td>ORGANIZATIONS_SCEN</td>
</tr>
<tr>
<td>Purchase Orders and Requisitions</td>
<td>SUPPLY_INTRANTIT_SCEN</td>
</tr>
<tr>
<td></td>
<td>PO_IN RECEIVING_SCEN</td>
</tr>
<tr>
<td></td>
<td>PO_SCEN</td>
</tr>
<tr>
<td></td>
<td>PR_SCEN</td>
</tr>
<tr>
<td>Planned Order Supplies</td>
<td>PLANNEDORDERSUP_SCEN</td>
</tr>
<tr>
<td>Resources</td>
<td>RESOURCES_SCEN</td>
</tr>
<tr>
<td></td>
<td>RESOURCE_CHANGE_SCEN</td>
</tr>
<tr>
<td></td>
<td>RESOURCE_SHIFTS_SCEN</td>
</tr>
<tr>
<td></td>
<td>RESOURCE_AVAILABILITY_SCEN</td>
</tr>
<tr>
<td>Routings</td>
<td>ROUTING_OPERATION_RESOURCES_SCEN</td>
</tr>
<tr>
<td></td>
<td>ROUTINGS_SCEN</td>
</tr>
<tr>
<td></td>
<td>ROUTING_OPERATIONS_SCEN</td>
</tr>
<tr>
<td>Sourcing Rules</td>
<td>SOURCING_ASSIGNMENTS_SCEN</td>
</tr>
<tr>
<td></td>
<td>SOURCING_RULES_SCEN</td>
</tr>
<tr>
<td></td>
<td>SOURCING_ASSIGNMENTSETS_SCEN</td>
</tr>
<tr>
<td></td>
<td>SOURCING_RECEIPT_ORGS_SCEN</td>
</tr>
<tr>
<td></td>
<td>SOURCING_SOURCE_ORGS_SCEN</td>
</tr>
</tbody>
</table>
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.
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Options and Option Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source System</td>
<td>Select from a list of source systems.</td>
</tr>
<tr>
<td>Collection Type</td>
<td>- <strong>Net change</strong>&lt;br&gt;  Data in the data repository is updated with the data uploaded from the staging tables.&lt;br&gt;  - <strong>Existing records are updated.</strong>&lt;br&gt;  For example, on hand is updated with current quantity.&lt;br&gt;  - <strong>New records are added to the data repository.</strong>&lt;br&gt;  For example, New purchase orders are added to the data repository.&lt;br&gt;  - <strong>Targeted</strong>&lt;br&gt;  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.</td>
</tr>
<tr>
<td>Group Identifier</td>
<td>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.</td>
</tr>
<tr>
<td>Regenerate Calendar Dates</td>
<td>- <strong>Yes</strong>&lt;br&gt;  You loaded calendar patterns into the staging tables so you need the concurrent process to generate and store individual dates to run.&lt;br&gt;  - <strong>No</strong>&lt;br&gt;  You loaded individual dates into the staging tables so you do not need the concurrent process to generate and store individual dates to run.</td>
</tr>
<tr>
<td>Regenerate Resource Availability</td>
<td>- <strong>Yes</strong>&lt;br&gt;  You loaded resource availability patterns into the staging tables so you need the concurrent process to generate and store individual dates to run.&lt;br&gt;  - <strong>No</strong>&lt;br&gt;  You loaded individual dates into the staging tables so you do not need the concurrent process to generate and store individual dates to run.</td>
</tr>
</tbody>
</table>
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:

   ```sql
   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:

   ```sql
   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,
   SELECT a.instance_code, a.instance_id, :b1, 'DONE', 'NORMAL',
   'LOAD_INTERFACE', 'OTHER', 'USER', SYSTIMESTAMP
   FROM msc_apps_instances a
   WHERE a.instance_code= :b2 ;
   ```
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.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source System</td>
<td>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.</td>
</tr>
<tr>
<td>Organization Collection Group</td>
<td>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.</td>
</tr>
</tbody>
</table>

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
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, records without errors or warnings are loaded into the data repository. For records with errors or warnings the following error and warning handling steps occur when using the Web services method:

- **Warnings:** Records with warnings are fixed automatically, then loaded into the data repository and copied into the applicable staging tables with the appropriate warning message.

- **Errors:** Records are loaded to the applicable staging tables instead of the data repository and are marked with the appropriate error message. When there is an error due to missing mandatory fields, in cases where possible, the collections process will retry the record. After several unsuccessful retry attempts, the record will be marked as error. In some cases, retrying is not an option, and the record will be immediately marked as an error.

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

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

- **Errors:** When there is an error due to missing mandatory fields, in cases where possible, the collections process will retry the record. After several unsuccessful retry attempts, the record will be marked as error. In some cases, retrying is not an option, and the record will be immediately marked as an error.

**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.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source System</td>
<td>Select a source system for the list of source systems. All data for the selected system will be deleted from the data repository.</td>
</tr>
<tr>
<td>Purge Global Entities</td>
<td>Yes or No</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>If you select no, data will be deleted from the source-specific entities only.</td>
</tr>
</tbody>
</table>

#### 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.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source System</td>
<td>Select a source system for the list of source systems. Data will be deleted for this source system only.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Parameter Options</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Record Type</td>
<td>The record type specifies which type of records to purge as follows:</td>
</tr>
</tbody>
</table>
|                    | • 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.
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.

External Interface Layer: How It Works with Integrating External Fulfillment Applications

The external interface layer is a module through which fulfillment requests from Oracle Fusion Distributed Order Orchestration are routed to external fulfillment applications. An external fulfillment application is a fulfillment application that is not an Oracle Fusion application.

This figure shows the flow of a sales order from the order capture application to the fulfillment application. The following steps are illustrated:

1. Send the sales order to Distributed Order Orchestration.
2. Transform the sales order to an orchestration order.
3. Assign an orchestration process to the orchestration order.
4. Execute the fulfillment step.
5. Initiate the request to the fulfillment application.
6. Determine the destination fulfillment application. This is done with routing rules.
7. Transform the enterprise business message (EBM) and send it to the connector.
8. Connector--The EBM is transformed to an application business message (ABM) and is sent to the fulfillment application.
9. Fulfillment application sends a response.
10. Transform the response from the fulfillment application.
11. Process the response from the fulfillment application.
12. Execute the next fulfillment step.
Orchestration
The orchestration layer is the module in which fulfillment tasks are executed. Fulfillment requests originate here and must be communicated to integrated fulfillment applications.

Task Layer
The task layer contains the services to request fulfillment tasks in integrated fulfillment applications. Distributed Order Orchestration provides services for some common fulfillment system functions, such as shipping and invoicing. You can add task layer functionality through the Template task layer services. The orchestration layer interacts with the external interface layer through the task layer.

External Interface Layer
The external interface layer uses data and routing rules to determine where to route fulfillment requests, and ultimately sends the requests to the appropriate fulfillment systems. Fulfillment requests are processed as follows:

1. The external interface layer receives a request from the task layer.
2. The external interface layer cross-references the data in the request.
3. The external interface layer transforms the request to enterprise business message format.
4. The external interface layer sends the enterprise business message to the connector. The connector is part of the external interface layer. The connector is the responsibility of the implementer.
5. The connector transforms the enterprise business message to an application business message that the fulfillment application can process.
6. The connector calls the fulfillment application.

In this framework, the external fulfillment application does not communicate directly with the orchestration process. The external interface layer acts as an intermediary through which requests are sent. When you integrate a new fulfillment application, you register the connector web service that integrates the fulfillment application with the external interface layer. Because the fulfillment application and the orchestration process are not connected directly, it is possible to implement multiple service providers for fulfillment applications at run time. This framework precludes the need for modifying the orchestration process every time a new fulfillment application is integrated. After the web services and routing rules are configured, Distributed Order Orchestration immediately can interface with the new external fulfillment applications.

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.
Creating an External Interface Routing Rule: Worked Example
This example demonstrates how to create an external interface routing rule that routes a request to a particular fulfillment system.

Summary of the Tasks
Create the following external interface routing rule: If a request contains server model ABC (Inventory Item ID 100000003357002) and a task type code of invoice, then route the service request to the XYZ fulfillment system.

Navigating to the Manage External Interface Routing Rules Page
1. On your implementation project, expand the Define Sales Order Fulfillment task list.
2. Expand the Define Orchestration task list.
3. Click the Go to Task icon next to Manage External Interface Routing Rules.

Creating a New Rule
1. Click the arrow next to the Add icon, and select Add Rule. A new rule called Rule 1 appears.
2. Replace Rule 1 with Route Invoice Request Rule.
3. Click the Expand icon.
4. Click the Show Advanced Settings arrows.
5. Enter a description, such as: Rule that routes invoice requests for product ABC to XYZ invoice system.
6. Click the Left Value icon.
7. Expand DOOExternalInterfaceLayer.
8. Expand FulfillLineTLVO.
9. Select InventoryItemId.
10. Click OK.
11. In the Operators list, select the Is list item
12. In the field next to the Right Value icon, enter "100000003357002L". L denotes a long value.
13. Click the arrow to display the list, and select Simple Test.
14. In the new row, click the Left Value icon.
15. Expand DOOExternalInterfaceLayer.
16. Expand HeaderTLVO.
17. Select TaskName.
18. Click OK.
19. In the Operators list, select the Is list item.
20. In the field next to the Right Value icon enter "Invoice".
21. In the Then region, click the Add Action icon.
22. Click the Action list to the right of the check box.
23. Select Assert New from the list. DOOExternalInterfaceLayer.Result appears in the field.
24. Click the **Edit Properties** icon to the right of the down arrow. The Properties window appears.
25. In the **Value** field of resultObj, enter a valid connector name.
26. In the **Value** field of resultObjKey, enter "SERVICE_NAME". The Properties window appears.
27. Click **OK**.
28. Click **Save**.

Creating an External Interface Routing Rule for Fulfillment Systems When Customer Cross-Referencing Data Is Not Found in Oracle Fusion: Worked Example

This example demonstrates how to create a routing rule that makes it possible for an order to proceed to an integrated fulfillment system when cross-reference data are not found for that system.

Summary of the Tasks

Create the following external interface routing rule: If the task type code on the fulfillment line is Shipment, then route the service request to the Shipping Fulfillment System A.

The customer data of Shipping Fulfillment System A are not found as cross-references in Oracle Fusion, so you must set the value of parameter RESOLVE_XREF_FOR_CUSTOMERINFO to Yes to indicate to the cross-reference service that Oracle Fusion system keys are returned by this service. If cross-referencing data were available, then target fulfillment system keys would be returned.

**Note**

This rule will send all shipment requests to Shipping Fulfillment System A. If you want to have more than one shipping system, then you must add conditions to differentiate the systems.

Create the If Clause

1. On the Manage External Interface Routing Rules page, click the arrow next to the Add icon and select Add Rule.
2. Enter a rule name, such as the name that was used to register the system connector on the Manage Web Service Details page: "Shipping Fulfillment System A"
3. Near the rule name, click the Expand icon.
4. Next to the rule name, click the Show Advanced Settings icon.
5. Select the Advanced Mode and Tree Mode check boxes.
6. In the Root list, select DOOExternalInterfaceLayer.HeaderTLVO.
7. In the field of the If region, enter: header
8. After "is a" select DOOExternalInterfaceLayer.HeaderTLVO.
9. In the field below Selected Tests, enter "Shipment". You must use quotation marks. You can invert the values in this step and the step in which you entered "header."
10. Select the Equals Ignore Case operator.
11. Click the Right Value icon.
12. Expand DOOExternalInterfaceLayer.
13. Expand HeaderTLVO.
14. Select TaskTypeCode.
15. Click OK.

Create the Then Clause

1. In the Then region, select Assert New.
2. In the list, select DOOExternalInterfaceLayer.Result.
3. Click the Edit Properties icon to the right of the list.
4. Click in the Value field of resultObj.
5. Enter "Shipping Fulfillment System A" (or the connector name you entered on the Manage Web Service Details page). You must use quotation marks.
6. Click in the Value field of ResultObjKey.
7. Enter "SERVICE_NAME". You must use quotation marks.
8. Click OK.
9. Select Assert New. If you are using Fulfillment task layer service, then you do not need to create a new rule but can add this action to the existing routing rule.
10. In the list, select DOOExternalInterfaceLayer.Result.
11. Click the Edit Properties icon to the right of the list.
12. Click in the Value field of resultObj.
13. Enter "Yes". You must use quotation marks.
14. Click in the Value field of ResultObjKey.
15. Enter "RESOLVE_XREF_FOR_CUSTOMERINFO". You must use quotation marks.
16. Click OK.
17. Click Save.

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.

For more information, see Oracle Fusion Distributed Order Orchestration: Implementing TTL in "Oracle Fusion Distributed Order Orchestration White Papers" (1536633.1) on My Oracle Support at https://support.oracle.com.

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.

For more information, see Oracle Fusion Distributed Order Orchestration: Implementing TTL in "Oracle Fusion Distributed Order Orchestration White Papers" (1536633.1) on My Oracle Support at https://support.oracle.com.

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.

For more information, see Oracle Fusion Distributed Order Orchestration Attachments in "Oracle Fusion Distributed Order Orchestration White Papers" (1536633.1) on My Oracle Support at https://support.oracle.com.

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 connector transforms the structure and content of the outbound message to match the inbound interface of the other system.

Distributed Order Orchestration communicates with the fulfillment systems using enterprise business objects. You must create and register the connector for each integrated fulfillment system that supports the enterprise business object and associated operations.

Set up Web services in the following order:

1. Create the transformation style sheet.
2. Create the connector.
3. Use the transformation style sheet.

4. Deploy the connector.

5. Register the connector.


For more information, see Oracle Fusion Distributed Order Orchestration: Non-Fusion Fulfillment System Connector in "Oracle Fusion Distributed Order Orchestration White Papers" (1536633.1) on My Oracle Support at https://support.oracle.com.

Create the Transformation Style Sheet

Create a transformation style sheet for Web service payloads. You can use the Oracle JDeveloper mapper tool or any other tool of choice.

Create the Connector

Define an XSLT transformation file to transform the Distributed Order Orchestration enterprise business object to a Web service-specific message. You can create the connector in any technical stack, but Business Process Execution Language is recommended. You can use the Oracle JDeveloper tool or any other tool.

Define the XSLT transformation file to transform the response from the Web service to a message specific to Distributed Order Orchestration.

Use the Transformation Style Sheet

Reference the transformation style sheet from the connector.

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. You can set up Oracle Business Rules utilizing order, fulfillment line, and process definition attributes to select the fulfillment system connectors. Then the specified connector service is invoked that translates the payload into the external fulfillment system structure.

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 communicate with an external fulfillment system, Oracle Fusion Distributed Order Orchestration invokes user-defined web services through the external interface layer. The external interface layer is a module through which fulfillment requests
from Oracle Fusion Distributed Order Orchestration are routed to external fulfillment applications. The external interface layer is not integrated directly with the orchestration layer, so fulfillment applications can be integrated dynamically without disrupting orchestration processes. To integrate an external fulfillment application, you can create a new connector web service or use an existing service. You log the web service URL on the Manage Web Service Details page of Distributed Order Orchestration. Then you create business rules to select the web service. Because the external interface layer is an intermediary layer that uses open access (mainly HTTP), extra security setup is required. You must ensure that the user credential is valid in the target system and in the security certificate to encrypt and decrypt messages.

For more information, see DOO Connector: Invoking DOO Web Services from Non-Fusion Order Capture System in "Oracle Fusion Distributed Order Orchestration White Papers" (1536633.1) on My Oracle Support at https://support.oracle.com.

User Credential

Obtain a user credential key in Oracle Wallet Manager, 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. Because there is no requirement to match the actual name of the connector you can provide a short name, which can then be used in routing rules. The CSF-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.

Integration with Oracle Fusion Distributed Order Orchestration

Using Public Services: Explained

Oracle Fusion Distributed Order Orchestration provides public services for integration of order capture and fulfillment applications. One such example is the Receive and Transform Sales Order Service, which is used for sales order submission. All Distributed Order Orchestration services are exposed as composite services.

You can find the services in Oracle Enterprise Repository. To locate the services, in the Search region enter the following:

<table>
<thead>
<tr>
<th>Field or List</th>
<th>Value to Enter or Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search String</td>
<td>DOO</td>
</tr>
</tbody>
</table>
Enterprise Business Object Usage in Integrations with Oracle Fusion Distributed Order Orchestration: Explained

When you integrate Oracle Fusion Distributed Order Orchestration with applications outside of Oracle Fusion Applications, you can leverage the Oracle Application Integration Architecture framework to facilitate communication between the disparate applications. Application Integration Architecture provides canonical models called enterprise business objects, each of which is a standardized representation of an independent business concept. An enterprise business object refers to a data model consisting of standard business data object definitions and reusable data components representing a business concept, such as a sales order or invoice. In integrations that are developed using Application Integration Architecture, the enterprise business object data model serves as a common data abstraction across systems. It supports loose coupling of systems in Application Integration Architecture and eliminates the need for one-to-one mappings of the disparate data schemas between each set of systems.

An enterprise business message communicates requests, such as create, update, and read, from Distributed Order Orchestration to an integrated application and vice versa. The enterprise business message passes the enterprise business object as its payload. Distributed Order Orchestration web services for external application integration accept enterprise business messages as input and return enterprise business messages as output, for example, CreateSalesOrder and UpdateSalesOrder.

If you use extensible flexfields, then you must extend the appropriate enterprise business object to include the extensible flexfield. For more information about extensible flexfield usage, see DOO: Oracle Fusion Distributed Order Orchestration White Papers (1536633.1) on My Oracle Support at https://support.oracle.com.

Distributed Order Orchestration leverages the following enterprise business objects:

- FulfillmentOrder
- InventoryReservation
- Invoice
- Process
- ReceiptAdvice
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. You don’t need to release these processes.

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

(Statues 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.

Orchestration Process Step Definition: Points to Consider

Orchestration process definitions contain the steps that are required to process a fulfillment line. Each step contains definitions of the task type, task, and service. A service step is a call to the task layer. Each step contains information about dependencies, planning, and change management.

Steps

Create the steps required to fulfill an order. Each step is a request to a fulfillment system.

• Step type: Function of the step in the orchestration process. You can see the steps in an orchestration process diagram.
  ◦ Conditional: Conditional node of the orchestration process. This is the point where one of two or more orchestration paths is executed, based on the results of a logical condition. You must specify a branching condition on a step that directly follows a conditional step.
  ◦ Merge: Point where two or more orchestration paths reunite.
  ◦ Service: Step where you define a service.
Subprocess: Set of steps that are defined by your organization so that they can be added as a unit to one or more orchestration processes. At runtime, the steps behave as if they were added individually to the process.

- Task type: Grouping of fulfillment tasks. Each task type contains a selection of services that can be used to communicate with a specific type of fulfillment system, for example, a billing system. The predefined task types are: Schedule, Reservation, Shipment, Invoice, Return, Activity, and custom task types. You can create new Activity types. Your task type selection determines the options of task name and service. When you select a task type, the corresponding task type indicator is automatically selected. This icon represents the task type in the Order Orchestration work area.

- Task: A task can have more than one step. For example, the Ship task invokes the Create Shipment service, and the Ship task also invokes the Wait for Shipment service to wait for different status values in the fulfillment system updates.

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

- Service: Fulfillment task service that is called in this step.

- Manual: If you designate a task as manual, then the orchestration process waits until a user completes the task manually in the Order Orchestration work area. You might want to make certain tasks manual, such as the Schedule task, so that an order manager can schedule all fulfillment lines manually at the end of the day, for example.

- Exit criteria: Task status that you select is used to mark the end of a wait step. For example, if the status of a shipment task is Shipped, then exit from the wait step.

  **Note**
  If your task has multiple wait steps, then ensure that the steps do not have the same set of exit criteria values. Also ensure that the last (or only) wait step uses the exit criterion status that indicates the task is complete. You make this designation on the Manage Task Status Conditions page.

- Line selection criteria: Expression used for filtering criteria when all of the orchestration order lines or fulfillment lines are not needed for the fulfillment task service invocation specified for the step.

Dependencies
Determine the dependencies between steps. You need to designate dependencies only if a conditional step is added to the orchestration process.

- Branching conditions: Criteria that must be met to determine whether steps in the node must be processed.

- Evaluation sequence: Order in which branching conditions must be evaluated.

- Otherwise: Required attribute if the step has more than one branch.

Planning
Create step-specific planning parameters. Set user expectations about the duration of each step, task, and orchestration process.

- Planning default branch: If multiple dependencies are present, then the step you mark as the default branch is used for planning until a branch is selected during execution.
Fulfillment completion step: Select the step that marks completion of the task. The chronological last step may not be the last step the customer cares about. For example, the orchestration process is planned with the requested ship date as the completion date for the step identified in the orchestration process as the last step.

Change Management
When you create an orchestration process definition, you have some options for how the orchestration process will respond to changes.

- Hold on wait: Sends a message to the fulfillment system for steps that are active when a change order is received.
- Use transactional item attributes: When this option is selected, the application checks all transactional item attributes when determining the difference between the change order and the previous version of the order.
- Use flexfield attributes: When this option is selected, the application checks all flexfield attributes when determining the difference between the change order and the previous version of the order.
- Compensation pattern: Set of rules that dictate what should be done to each step that was executed prior to receipt of the change order. For example, if a change order arrives requesting a change from carpet to tile, then all of the previous steps need to be canceled. Assume shipping of the carpet was scheduled; now consider what needs to be done to previous steps. The change order is run through decomposition, resulting in different fulfillment lines for tile. A different orchestration process is assigned because tile orders require more time and a different contractor. Most of the previous steps need to be canceled, but the Measure step can be reused because the room dimensions are still accurate.

If you do not specify a compensation pattern, then the affected step may be processed as an update, depending on the context of the operation. Some steps may be redone instead. If compensation is not possible, then the step is bypassed and the next step is compensated.

Parallel Processing Setup: Explained
You may want to set up parallel processing so that you can have more than one task running at the same time. You can filter the fulfillment lines of a group so that the tasks for some lines in the group run at the same time as tasks for the other lines. Alternatively, you can create a parallel process that does not filter lines in a branch, so that more than one task is active for each line.

Setting up an orchestration process with parallel processing is the same as setting up an orchestration process with a conditional branch, except for the following:

- When you create a branch, you select a step type of Parallel.
- You do not need to create branching conditions, set an evaluation sequence, or select Otherwise.

Note
You cannot split a line that is in a parallel branch.

Task Types: Explained
A task type is a group of services that Order Orchestration uses to perform a fulfillment task. It represents a common business function that Order Orchestration requires to process an order from the time that it receives the order from the order capture system, to the time that it runs the order in the fulfillment application. A task type includes the following tasks and services:

- An internal Web service that communicates with the task layer
- A task that represents the services that the task type uses
Order Orchestration comes predefined with the following task types:

- Activity
- Fulfillment Order
- Invoice
- Pause
- Schedule
- Reservation
- Return
- Shipment

These task types are read-only. You cannot modify or delete them. However, you can use the Custom or the Activity task type to create a new task type. Note the following:

- You can modify the name of a task type that you create, but it is not recommended.
- You can edit the service names of an activity and or custom task type that you create.
- You can add a service from the services that are available, but you cannot edit or delete the services that a custom task type references.

You can use a task to represent the services that a task type references. For example, if you create a Ship Goods task that references services from the Shipment task type, then the Order Orchestration work area displays Ship Goods when a Shipment service runs, regardless of whether or not it calls the Create Shipment or the Update Shipment service. The Order Orchestration work area does not display the service. It displays only the task. You can define several tasks for a task type, such as ShipGoods or ShipWidgets.

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

Subprocesses: Explained
A subprocess is a set of sequential steps of an orchestration process definition that is incorporated into a parent orchestration process definition. Using subprocesses can increase administrative efficiency because you can insert a subprocess into an orchestration process definition instead of adding each step individually. You can use the same subprocess in any number of orchestration process definitions.

Subprocesses are created on the Manage Orchestration Process Definitions page in the Setup and Maintenance work area. Create a subprocess just as you would create any orchestration process definition, except you must clear the Parent process check box and you don’t need to select a fulfillment completion step. The Status tab disappears when you clear the Parent process check box, so that you can’t select any status conditions. You must release the subprocess, but no download will be available.

Insert a subprocess into your parent orchestration process definition by creating a new step and selecting the Subprocess step type. Select the name of the subprocess you want to use. Only released subprocesses are available on the list.

Example of a Subprocess
Your organization has many different steps for processing an order, depending on the item. However, your organization bills all orders the same way.
All orders share these billing steps:

- Create invoice
- Wait for invoice
- Send request to activity system that notifies customer of payment receipt
- Send request to activity system that generates coupons.

Therefore, you create a subprocess that consists of these steps. Subsequently, when you create orchestration process definitions, you incorporate the subprocess into the definitions.

Creating an Orchestration Process Definition: Worked Example
This example demonstrates how to create an orchestration process definition.

Summary of Steps
The steps for creating an orchestration process definition without branching can be summarized as follows:

1. Enter the required information in the header.
2. Create the orchestration process definition steps.

Enter the Required Information in the Header
On the Manage Orchestration Process Definitions page, click the Create icon.

1. On the Create Orchestration Process Definition page, enter an orchestration process name without any spaces, such as OrchestrationProcess1.
2. Select the Ship Order process class.
3. Select the Common set ID.

Create the Orchestration Process Definition Steps

Create the Schedule Product Step

1. On the Step Definition tab, click the Add Row icon.
2. Next to Step 100, enter a step name, such as Schedule Product.
3. Select the Service step type.
4. Select the Schedule task type.
5. Select the Schedule task name.
6. Select the Create Scheduling service name.

Create the Reserve Product Step

1. With your cursor on the Schedule Product step (step 100), click the Add Row icon.
2. Next to Step 200, enter a step name, such as Reserve Product.
3. Select the Service step type.
4. Select the Reservation task type.
5. Select the Reserve task name.
6. Select the Create Inventory Reservation service name.

Create the Ship Product Step

1. With your cursor on the Reserve Product step, click the Add Row icon.
2. Next to Step 300, enter a step name, such as Ship Product.
3. Select the Service step type.
4. Select the Shipment task type.
5. Select the Ship task name.
6. Select the Create Shipping service name.

Create the Wait for Product Shipment Step

1. With your cursor on the Ship Product step, click the Add Row icon.
2. Next to Step 400, enter a step name, such as Wait for Product Shipment.
3. Select the Service step type.
4. Select the Shipment task type.
5. Select the Ship task name.
6. Select the Wait for Shipment service name.
7. Set the exit criteria to Shipped.

Create the Create Invoice Step

1. With your cursor on the Wait for Product Shipment step, click the Add Row icon.
2. Next to Step 500, enter a step name, such as Create Invoice.
3. Select the Service step type.
4. Select the Invoice task type.
5. Select the Invoice task name.
6. Select the Create Billing Lines service name.

Create the Wait for Invoice Step

1. With your cursor on the Create Invoice step, click the Add Row icon.
2. Next to Step 600, enter a step name, such as Wait for Invoice.
3. Select the Service step type.
4. Select the Invoice task type.
5. Select the Invoice task name.
6. Select the Wait for Billing service name.
7. Set the exit criteria to Billed.

Creating an Orchestration Process Definition with Conditional Branching: Worked Example

This example demonstrates how to create an orchestration process definition with conditional branching. In this example, a retailer that sells gifts offers gift wrapping using standard yellow gift wrap with the company logo. Alternatively, a customer can select holiday gift wrap. The order administrator needs an orchestration process that models this business flow.

The following figure shows part of the orchestration process that you are creating. Conditional branches begin after the Schedule Merchandise step. The branches merge before the Ship Merchandise step.

Summary of the Steps
The steps for creating an orchestration process definition with branching can be summarized as follows:

1. Enter the required information in the header.
2. Create the standard orchestration process definition for gift wrap: In this series of steps you will create an orchestration process definition assuming the standard gift wrap option. Later you will add the branching logic, and this will be one of the branching options.

3. Create the branching logic, and add the holiday gift wrap option. You will also return to the Standard Gift Wrap step to add some branching options.

This example assumes:

- Enabled extensible flexfield with context Wrap and segment Type, enabled at the fulfillment line level.
- Tasks with the following names: Schedule (in Schedule task type), Reserve (in Reservation task type), Ship (in Shipment task type), Activity.StandardWrap and Activity.HolidayWrap (in the Activity task type).

**Note**

After you complete the steps below, you must create a default status rule set with the statuses Completed and Shipped at minimum. You must set planning fields before you can release an orchestration process. You may also want to set change management options before validating, releasing, and deploying an orchestration process. These actions are not covered in this example.

**Entering the Required Information in the Header**

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

1. On the Create Orchestration Process Definition page, enter an orchestration process name without any spaces, such as OrchestrationProcessGiftWrap.


3. In **Process Class**, select Ship Order Class.

4. In **Set**, select Common Set.

5. Click **Save**.

**Creating the Standard Orchestration Process Definition for Gift Wrap**

**Create the Scheduling Step**

1. On the Step Definition tab, click **Add Row**.

2. Next to Step 100, enter Schedule Merchandise.

3. Select the Service step type.

4. Select the Schedule task type.

5. Select the Schedule task.

6. Select the Create Scheduling service.

**Create the Standard Gift Wrap Step**

1. With your cursor on the Scheduling step (step 100), click **Add Row**.
2. Next to 200, enter Activity Standard Gift Wrap.
3. Select the Service step type.
4. Select the Activity task type.
5. Select the Activity_StandardWrap task.
6. Select the Create Activity service.

Create the Wait Step for the Standard Gift Wrap
2. Next to 300, enter Activity Wait for Standard Gift Wrap.
3. Select the Service step type.
4. Select the Activity step type.
5. Select the Activity_StandardWrap task.
6. Select the Wait for Activity service.
7. Set the exit criteria to Completed.

Create the Shipment Step
1. With your cursor on the Activity Wait for Standard Gift Wrap step (step 300), click Add Row.
2. Next to 400, enter Ship Merchandise.
3. Select the Service step type.
4. Select the Shipment task type.
5. Select the Ship task.
6. Select the Create Shipping service.
7. Select Fulfillment Completion Step.

Create the Wait for Shipment Step
1. With your cursor on the Ship Merchandise step (step 400), click Add Row.
2. Next to 500, enter Wait for Shipment Advice.
3. Select the Service step type.
4. Select the Shipment task type.
5. Select the Ship task.
6. Select the Wait for Shipment service.
7. Set the exit criteria to Shipped.
Create the Invoicing Step

1. With your cursor on the Wait for Shipment Advice step (step 500), click **Add Row**.
2. Next to 600, enter Create Invoice.
3. Select the Service step type.
4. Select the Invoice task type.
5. Select the Invoice task.
6. Select the Create Billing Lines service.

Create the Wait for Invoicing Step

1. With your cursor on the Create Invoice step (step 600), click **Add Row**.
2. Next to 700, enter Wait for Invoicing.
3. Select the Service step type.
4. Select the Invoice task type.
5. Select the Invoice task.
6. Select the Wait for Billing service.
7. Set exit criteria to Billed.

Creating the Special Holiday Processing Steps

Create the Conditional Node

Create a conditional node from which the process will split into two branches. Only one branch is used per process. Whether the process uses one branch or the other depends on a condition that you define.

Note

The numbers that were previously assigned to the steps may change as you add new steps.

1. With your cursor on the Schedule step (step 100), click **Add Row**.
2. Next to 200, enter Conditional Node.
3. Select the Conditional step type.

Set Branching Options on the Standard Gift Wrap Step

1. Return to the Activity Standard Gift Wrap step and select Otherwise.
2. Set the **Evaluation Sequence** to 2.

Create the Holiday Gift Wrap Step

1. With your cursor on the Conditional Node step (step 200), click **Add Row**.
2. Next to 300, enter Activity Holiday Gift Wrap.
3. Select the Service step type.
4. Select the Activity task type.
5. Select the Activity_HolidayWrap task.
6. Select the Create Activity service.
8. In the Branching Condition column, select **Click for Rule**.
9. In the Branching Condition Rule Set window, click the **Add Rule** icon.
10. Click the **Expand** icon.
11. Click the **Show Advanced Settings** downward arrows.
12. Select Advanced Mode and Tree Mode.

### Creating the If Statement

1. In the list of values to the right of Is A, ensure that DooSeededOrchestrationRules.DO0Header is populated.
2. Replace the populated field to the left of Is A with Header.
3. Click the arrow to the right of the **Right Value** icon of the test, which is directly below the Header declaration, and then select Delete Test.
4. On the same line as the Header declaration, click the **Add Pattern** icon.
5. In the list of values to the right of Is A, select Header/childFLines/flexcontexts.
6. Replace the populated field to the left of Is A with: FLineEFF.
7. Directly below the last pattern that you created in the previous step, click the arrow next to **Add Test**, and then select Simple Test.
8. Click the **Left Value** icon.
9. Click the **Expression Builder** icon.
10. Select Functions.
11. Expand FLineEFF, and then expand Context., select EqualsIgnoreCase(String), and then click **Insert Into Expression**.
12. Replace the word String with Wrap.
13. Click **OK** to close the Expression Builder window.
14. Click **OK** to close the Condition Browser window.
15. To the right of your selection, select Isn’t.
16. To the right of Isn’t, enter: null. Do not use uppercase letters.
17. Click the downward arrow to the right of the condition test that you just created, and select Simple Test.
18. Click the **Left Value** icon of the new test.
19. In the Condition Browser, click the **Expression Builder** icon.
20. Select Functions.
21. Expand FLineEFF, select getFlexAttributeValue(String), and click **Insert Into Expression**.
22. Replace the word String with _Type. Make sure you enter the underscore.
23. Click **OK** to close the Expression Builder window.
24. Click **OK** to close the Condition Browser window.
25. To the right of Is, enter "Holiday". Make sure you use quotation marks.

**Creating the Then Statement**

1. Click the arrow next to **Add Action** and select Assert New.
3. Click the **Edit Properties** icon.
4. Click the **Value** icon on the ResultObj row.
5. In the Condition Browser, expand DooSeededOrchestrationRules and Boolean. Then select True.
6. Click **OK** to close the Condition Browser window.
7. In the field next to **ResultObjKey**, enter "1". Make sure you use quotation marks.
8. Click **OK** to close the Properties window.
9. In the Branching Condition Rules window, click **Validate**. If no errors appear, then click **Save**.
10. Click **Save** on the Create Orchestration Process Definition page, where you entered the steps of the orchestration process.

**Create the Wait Step for the Holiday Gift Wrap Activity**

1. With your cursor on the Activity Holiday Gift Wrap step (step 300), click **Add Row**.
2. Next to 400, enter Activity Wait for Holiday Gift Wrap.
3. Select the Service step type.
4. Select the Activity task type.
5. Select the Activity_HolidayWrap task.
6. Select the Wait for Activity service.
7. Set the exit criteria to Completed.

**Create the Merge Step**

1. With your cursor on the Activity Wait for Standard Gift Wrap step (step 600), click **Add Row**.
2. Next to 700, enter Merge.
3. Select the Merge step type.
4. Click Save.

Using Pause Tasks: Examples

A Pause task temporarily stops an orchestration process to wait for a particular condition to be met. When the condition is met, the Pause task is released and the orchestration process proceeds to the next step.

The following scenarios illustrate when you might want to use a Pause task.

Time-Based Pause Task

Let’s say your customer can’t receive ordered goods very early. You use the requested ship date to prevent early delivery, but sometimes this date is overlooked in the warehouse. You can set up a Pause task that pauses the orchestration process until 2 days before the scheduled ship date.

Or perhaps your company offers presales of upcoming video games up to seven days before the release date. Your company allows seven days because that is the estimated time to schedule and process the order prior to shipping. This lead time make it possible for your company to ship the game as soon as it’s available. You can set up a Pause task to pause the orchestration process, before the schedule step, until the game is released. In this case, the rule that releases the Pause task is the release date (an extensible flexfield) minus seven days. When the pause is released, the order is scheduled and the orchestration process continues.

Event-Based Pause Task

Let’s say your company has a layaway program that allows a customer to reserve an item and then pay for it in installments. You can set up a Pause task that pauses the orchestration process. When the customer pays in full, the Release Paused Tasks public service is called or the Release Pause Tasks process is run, and the Pause task is released. Then, the fully paid item is shipped to the customer.

Or perhaps your company has a policy that it releases orders for shipping only when inventory is available at the fulfillment location. You can set up a Pause task that pauses the orchestration process. When inventory is available, the Release Paused Tasks public service is called or the Release Pause Tasks process is run, and the Pause task is released. Then, the items are shipped to the customer.

Pause Task for Lines With Dependencies

Let’s say you have a customer who wants to receive shipment of an entire order of coffee, paper cups, sugar, and creamer at the same time. You can set up a Pause task that pauses each line until it is scheduled. The order is reevaluated periodically until all lines are scheduled. Then all lines are sent to the customer at the same time.

Or perhaps your customer wants to receive invoices of all lines of an order at the same time, regardless of when the items are shipped. You can set up a Pause task that pauses the orchestration process until all items are shipped. Then the invoices of all lines are sent to the customer at the same time.

Creating an Orchestration Process Definition with an Event-Based Pause Task: Worked Example

This example demonstrates how to create a Pause task as part of an orchestration process definition. A Pause task temporarily stops an orchestration process to wait until a particular date or for a particular event (not related to Oracle Fusion Middleware event framework). You might want to use a Pause task to wait until all payments are made before releasing an item from layaway or to wait for a specific date to arrive before releasing preorders.

Summary of the Tasks

Create an orchestration process for pausing preordered items before shipping.

1. Create an orchestration process definition.
2. Create the Pause task.

3. Create the start-after condition.

   The rule used in this example is: If the customer demand class is Gold, then release the pause using the Release Pause Tasks scheduled process that searches for the condition. Note that you will create two rules. One rule pauses the task if the conditions are true. The other rule skips the pause if the conditions of the first rule are false.

   **Note**

   If you plan to use If, Then, and Else within a single rule instead of creating two rules, then you must create the Else before you add an action in the Then statement.

Creating an Orchestration Process Definition

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

2. In the **Process Name** field in the header, enter a process name. The name can’t have spaces.

3. In the **Set** list of values, select a set, such as Common Set.

4. In the **Process Class** list of values, select a class, such as Ship Order Class.

5. On the Step Definition tab, click **Add Row**.


7. Select the Service step type.

8. Select the Schedule task type.

9. Select the Schedule task name.

10. Select the Create Scheduling service name

11. Click **Save**.

12. Using the same procedure, create the following additional steps: Create Reservation, Create Shipment, Wait for Shipment Advice, Create Invoice, Wait for Invoice. Select the Service step type for all. Pick a task type that matches the step, for example, select the Invoice task type for the invoicing steps.

Creating the Pause Step

1. Place your cursor in the Create Reservation step, and click the **Add Row** icon. The new step is added below the step where your cursor is positioned.

2. In the **Step Name** field, enter Pause.

3. In the **Step Type** list of values, select Pause.

4. In the **Task Type** list of values, select Pause.

5. In the **Task Name** list of values, select Pause.

6. In the **Service Name** list of values, select Pause Process.

7. Click **Save**.
Creating the Start-After Condition

First Rule: Create the If Clause

1. In the Start-After Condition column, select Click for Rule.
2. Next to the rule name, click the Expand icon.
3. Click the Collapse Pane icon to expand the visible space.
4. Click the Left Value icon.
5. In the Condition Browser, expand DooSeededOrchestrationRules.
7. Select demandClassCode.
8. Click OK.
9. Select the Is operator.
10. Click the Right Value icon.
11. Enter "Gold". Quotation marks are required.

First Rule: Create the Then Clause
Create the Then clause. The event is a parameter in a scheduled process, but it can also be used in a public service. Other parameters can also be used to release Pause tasks.

1. In the Then region, select Assert New.
3. Click the Edit Properties icon.
4. In the reevaluateFlag field of the Properties window, enter "N". Quotation marks are required.
5. Click the Value icon next to sacType.
7. Expand SacResult.
8. Select SAC_TYPE_EVENT.
9. Click OK.
10. Next to the eventName field, enter "Gold Preorders" into the Value field. Quotation marks are required.
11. Click OK.
12. Click Save.

Second Rule: Create the If Clause

1. In the Start-After Condition column, select Click for Rule.
2. Next to the rule name, click the Expand icon
3. Click the **Collapse Pane** icon to expand the visible space.

4. Click the **Left Value** icon.

5. In the Condition Browser, expand DooSeededOrchestrationRules.


7. Select demandClassCode.

8. Click **OK**.

9. Select the Isn't operator.

10. Click the **Right Value** icon.

11. Enter "Gold". Quotation marks are required.

**Second Rule: Create the Then Clause**

1. In the Then region, select Assert New.


3. Click the **Edit Properties** icon.

4. In the **reevaluateFlag** field of the Properties window, enter "N".

5. Click the **Value** icon next to sacType.


7. Expand SacResult.

8. Select SAC_TYPE_IMMEDIATE.

9. Click **OK**.

10. Click **OK**.

11. Click **Save**.

**Creating an Orchestration Process Definition with a Time-Based Pause Task: Worked Example**

This example demonstrates how to create a Pause task with a time-based rule for resuming the orchestration process.

**Summary of the Tasks**

Create an orchestration process for pausing pre-ordered items before shipping.

1. Create an orchestration process definition.

2. Create the Pause task.

3. Create the start-after condition.

The rule used in this example is: If the source order system is Legacy, then release the pause on ShipDate minus 2. Note that you will create two rules. One rule releases the pause on a certain date if the condition is met. The other rule releases the pause immediately if the condition is not met.
Creating an Orchestration Process Definition

1. On the Manage Orchestration Process Definitions page, click the Create icon.
2. In the Process Name field in the header, enter a process name. The name cannot have spaces.
3. In the Set list, select a set, such as Common Set.
4. In the Process Class list, select a class, such as Ship Order Class.
5. On the Step Definition tab, click Add Row.
7. Select the Service step type.
8. Select the Schedule task type.
9. Select the Schedule task name.
10. Select the Create Scheduling service name.
11. Click Save.
12. Using the same procedure, create the following additional steps: Create Reservation, Create Shipment, Wait for Shipment Advice, Create Invoice, Wait for Invoice. Select the Service step type for all. Pick a task type that matches the step; for example, select the Invoice task type for the invoicing steps. On the wait steps, select Exit Criteria, such as Shipped and Billed.

Creating the Pause Step

1. Place your cursor in the Create Reservation step of the process that you just created, and click the Add Row icon. The new step is added below the step where your cursor is positioned.
2. In the Step Name field, enter Pause.
3. In the Step Type list, select Service.
4. In the Task Type list, select Pause.
5. In the Task Name list, select Pause.
6. In the Service Name list, select Pause Process.
7. Click Save.

Creating the Start-After Condition

First Rule

Creating the If Statement

Create the If statement: If the source order system is Legacy.

1. In the Start-After Condition column of the Pause task, select Click for Rule.
2. Directly above the DOO Customizable Rules tab, which is where errors are displayed, click Collapse Pane to show more of the screen.
3. Directly below the rule set name, click the Add Rule icon.
4. Substitute Rule 1 with a name, such as ResumeShippingMinus2.
5. Next to the rule name, click the Expand arrow and the Show Advanced Settings arrows.
6. Select the **Advanced Mode** check box.
7. Select the **Tree Mode** check box, which maintains the orchestration order and orchestration process hierarchies during rule evaluation.
8. Click the Hide Advanced Setting arrows to view more of the window.
9. Open the Root list and select the DooSeededOrchestrationRules.DOOHeader list item.
10. In the left field that contains "DooSeededOrchestrationRules.DOOHeader," substitute the current text with: Header.
11. To the right of the empty fields below where you entered Header, click the search icon next to the Right Value icon and select Delete Test.
12. Click the Add Pattern icon to the right of DooSeededOrchestrationRules.DOOHeader.
14. In the list to the right of "Is A" select Header/childFLines.
15. Click the arrow of the Add Test icon, and select Simple Test
16. Click the Left Value icon.
17. In the Condition Browser, expand Header and select sourceOrderSystem.
18. Click **OK** to close the Condition Browser.
19. In the operator list select Is.
20. In the field to the right of the operator list, enter "LEG" to represent the legacy system. Enclose the text in quotation marks.

**Creating the Then Statement**

Create the Then statement: Then release the Pause task two days before the scheduled ship date.

1. In the Then region, click the arrow of the Add Action icon, and select Assign New.
2. Click the arrow of the "Select a target" list and select DooSeededOrchestrationRules.Timestamp.
3. In the empty field, enter DateTime. Do not use quotation marks. DateTime is the name you are giving a run time variable for the date calculation.
4. To the right of DateTime, click the Expression Value icon.
5. In the Condition Browser, expand Line and select scheduleShipDate.
6. Click **OK** to close the Condition Browser.
7. Click the arrow next to the Add Action icon, and select Modify.
8. Click the "Select a target" list, and select DateTime.
9. Click the Edit Properties icon.
10. In the **Time** field of the Properties window, click the Value icon.
11. In the Condition Browser, expand DateTime, and select Time.

12. Directly to the right of the text in the field at the top of the condition browser, enter: -(2*24*60*60*1000) You are setting up the calculation (for time in milliseconds) scheduled ship date - 2 days to have the Pause task automatically end two days before the lines are scheduled for shipping.

13. Click **OK** to close the Condition Browser.

14. In the Properties window, click **OK**.

15. Click the arrow next to the Add Action icon, and select Assert New.

16. In the “Select a target” list select DooSeededOrchestrationRules.SacResult.

17. Click the Edit Properties icon.

18. In the **reevaluateFlag Value** field in the Properties window, enter "N". Enclose the text in quotation marks.

19. Click the Value icon.

20. In the Condition Browser, expand DooSeededOrchestrationRules and SacResult.

21. Select **SAC_TYPE_TIMER**

22. Click **OK** to close the Condition Browser.

23. In the waitDateTime row of the Properties window, click the Value icon.

24. In the Condition Browser, select DateTime.

25. Click **OK** to close the Condition Browser.

26. Click **OK** in the Properties window.

**Second Rule**

**Creating the If Statement**

Create the If statement: If the source order system isn’t Legacy.

1. In the same rule set as the first rule, click the Add Rule icon directly below the rule set name. Rule 2 appears below your first rule.

2. Substitute Rule 2 with a name, such as ResumeShippingImmed.

3. Click the Expand icon to the left of the rule name that you just created.

4. Click the Show Advanced Settings arrows.

5. Select the **Advanced Mode** check box.

6. Select the **Tree Mode** check box, which maintains the orchestration order and orchestration process hierarchies during rule evaluation.

7. Click the Hide Advanced Setting arrows to view more of the window.

8. Open the Root list and select the DooSeededOrchestrationRules.DO0Header list item.

9. In the left field that contains "DooSeededOrchestrationRules.DO0Header," substitute the current text with: Header.

10. To the right of the empty fields below where you entered Header, click the search icon next to the Right Value icon and select Delete Test.
11. Click the Add Pattern icon to the right of DooSeededOrchestrationRules.DOOhedera.

12. Enter: Line

13. In the list to the right of "Is A" select Header/childFLines

14. Click the arrow of the Add Test icon and select Simple Test.

15. Click the Left Value icon.

16. In the Condition Browser, expand Header and select sourceOrderSystem.

17. Click OK to close the Condition Browser.

18. In the operator list select Isn't

19. In the field to the right of the operator list, enter "LEG" to represent the legacy system. Enclose the text in quotation marks.

Creating the Then Statement
Create the following then statement: Then release the Pause task immediately.

1. Click the arrow next to the Add Action icon and select Assert New.

2. In the Object list, select DooSeededOrchestrationRules.SacResult.

3. Click the Edit Properties icon.

4. In the reevaluateFlag Value field of the Properties window, enter "N". Enclose the text in quotation marks.

5. The reevaluateFlag is used for a Pause type of Timer to specify whether the rule be evaluated again once the calculated date arrives.

6. In the reevaluateFlag row, click the Value icon.

7. In the Condition Browser, expand DooSeededOrchestrationRules and SacResult.

8. Select SAC_TYPE_IMMEDIATE.

9. Click OK to close the Condition Browser.

10. In the Properties window, click OK.

11. In the Start-After Condition window, click Save.


Creating an Orchestration Process Definition That Sends All Invoices Together: Worked Example
This example demonstrates how to coordinate orchestration processes to ensure that invoicing for all the lines on an order is done at the same time, even if the shipment dates vary.

The instructions below assume that you know how to create an orchestration process definition.

A school district places an order for 600 new history books. The order must be sourced from three different warehouses.

- Fulfillment line 1: 80 books from Atlanta warehouse
• Fulfillment line 2: 400 books from Dallas warehouse
• Fulfillment line 3: 120 books from Chicago warehouse

Each fulfillment line will be processed by the same sequence of steps and the same orchestration process definition (but not the same orchestration process), but the lines may not necessarily be in sync. The school district wants to be invoiced after all the books are shipped.

You need to create an orchestration process that processes the lines but pauses them until all are shipped. Only then will they be allowed to proceed to billing. The steps of the orchestration process are:

1. Schedule
2. Reserve
3. Ship
4. Wait for Shipment
5. Pause
6. Invoice
7. Wait for Billing

Next you need to create the start-after condition for the Pause task.

Creating the Start-After Condition

Create two rules to indicate when to use the Pause task. The first is the default rule. It releases the pause on all lines immediately. The second rule indicates the conditions under which specified lines must be paused. The rules are reevaluated until the conditions no longer apply. The second rule overrides the default rule if the conditions apply.

To create the start-after condition, select Click for Rule under Start After Condition for the Pause task.

1. Create the default rule.
2. Create the rule that dictates the pause conditions.

Creating the Default Rule

1. Click the Add Rule icon.
2. Click the Expand button.
3. Click the Show Advanced Settings button.
4. Select Highest Priority.
5. Select Advanced Mode.
6. In the left field below If, enter an alias, such as Header.
7. In the list of values in the If region, select DooSeededOrchestrationRules.DOOReader.
8. Delete the test that is below the above statement by selecting the arrow next to Right Value icon and select Delete Test. Without the test, the actions in the Then region always apply.
9. In the Then region, click the Add Action arrow and select Assign New.
10. In the **Select a Target** list of values, select DooSeededOrchestrationRules.SacResult.

11. In the middle field, enter: SAC

12. To the right of =, click the **Expression Value** icon.

13. In the Condition Browser, expand DooSeededOrchestrationRules, expand SacResult, and select New.

14. Click **OK** to close the Condition Browser.

15. Click the arrow next to **Add Action** and select Assign.

16. In the **Select a Target** list of values, select Header.sacResult.

17. To the right of =, enter SAC.

18. Click the arrow next to **Add Action** and select Assign.

19. In the **Select a Target** list of values, select Header.sacResult.sacType.

20. To the right of =, click the **Expression Value** icon.

21. In the Condition Browser, expand DooSeededOrchestrationRules, expand SacResult, and select SAC_TYPE_IMMEDIATE.

22. Click **OK** to close the Condition Browser.

23. Click **Validate**, which validates and saves your work but doesn't close the window.

**Creating the Rule That Dictates the Pause Conditions**

Create the following rule: If the actual ship date is not null, then release the pause task. The pause is released when the actual shipped date is not null on all lines.

1. Click the **Add Rule** icon.

2. Click the **Expand** button.

3. Click the **Show Advanced Settings** button.

4. Keep the default priority setting.

5. Select Advanced Mode and Tree Mode.


7. In the If region in the field to the left of Is A, replace the defaulted value with an alias, such as OrderHeader.

8. Delete the test by clicking the down arrow to the right of the **Right Value** icon and selecting Delete Test.

9. Click the **Add Pattern** icon to the right of the If statement.

10. To the right of the empty fields, click the **Surround Pattern with Parenthesis** icon.

11. In the list of values under And, select "There is a Case Where." This selection means that if even one line is still not shipped, then all shipped lines must pause.

12. In the field on the left, enter FLine.

13. To the right of Is A, select Header/allFLinesIntheOrder. This selection gives you the ability to coordinate across multiple orchestration processes.
14. Click the arrow next to the Add Test icon, and select Simple Test. This is the test for the reason to pause.
15. Click the Left Value icon, and expand FLine and then select actualShipDate.
16. Click OK to close the Condition Browser.
17. Ensure that Is is selected.
18. In the field to the right of Is, click the Right Value icon and select null.
19. In the Then region, click the arrow next to Add Action and select Assign.
20. Select Header.sacResult.eventName.
21. Click the Expression Value icon on the same line.
22. In the Condition Browser, expand DooSeededOrchestrationRules and SacResult. Then select SAC_SYSTEM_EVENT_IPC_PAUSE, which forces reevaluation of the associated lines until all are shipped and then releases them.
23. Click OK to close the Condition Browser.
24. Click the arrow next to Add Action and select Assign.
25. Select Header.sacResult.reevaluateFlag.
26. In the field to the right of = enter "Y". Make sure you use quotation marks.
27. Click the arrow next to Add Action and select Assign.
28. Select Header.sacResult.sacType.
29. Click the Expression Value icon on the same line.
30. In the Condition Browser, expand DooSeededOrchestrationRules and SacResult. Then select SAC_TYPE_EVENT.
31. Click OK to close the Condition Browser.
32. Click Save.

Releasing a Pause Task Automatically: Explained
Pause tasks can be released automatically so that an orchestration process can resume. You can configure the following scheduled process or public service:

- Release Pause Tasks scheduled process
- Release Paused Tasks public service

Both types of automated releases release all active Pause task that meet the criteria you select.

Release Pause Tasks Scheduled Process
Ends one or more Pause tasks. You schedule the process on the Scheduled Processes page. You can use any of the following parameters to identify the Pause tasks to release:

- Event Name
- Inventory Item
• Sold-to Customer
• Task
• Order Number
• Fulfillment Line Number
• Warehouse
• From Order Date
• To Order Date
• From Scheduled Ship Date
• To Scheduled Ship Date
• From Scheduled Release Date
• To Scheduled Release Date
• Source Order System
• Source Order Number

Release Paused Tasks Public Service
This is a public service that is called from an external application to release the wait started by the Pause task type. The service parameters are used to identify the set of Pause tasks that is released. The event handler calls the service to end the wait for the combination of parameters that are specified. The service responds with a success message when the release is complete. If an error occurs, then the service responds with an error message to Oracle Fusion Distributed Order Orchestration, as well as to the calling system.

You can use the order capture service that requests order details to get the Distributed Order Orchestration attributes that correspond to the sales order

For more information about this service, see Oracle Enterprise Repository.

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:

• Use transactional item attributes: Indicates whether the product-specific transactional item 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 is 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.

Use the tabs on the Manage Status Values page to create and configure the statuses:

1. Status Codes tab: Enter a status code, and create a display name, which is how the status will appear in the Order Orchestration work area
2. Task Types tab: Assign the code you created to a task type. If the status is from an external system, then assign it to the task that matches the original system. For example, if the status is from a shipping system, then assign it to the Shipping task type.
3. Orchestration Process Classes tab: Assign the status code to fulfillment lines or orchestration processes, wherever you want the status code to be used. For example, when administrators create status conditions for orchestration processes, they can choose only from these status codes. 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.

<table>
<thead>
<tr>
<th>Orchestration Order Line Statuses</th>
<th>Orchestration Order Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>All orchestration order lines are completed.</td>
<td>Closed</td>
</tr>
<tr>
<td>All orchestration order lines are not completed.</td>
<td>Open</td>
</tr>
<tr>
<td>Some, but not all, orchestration order lines are completed.</td>
<td>Partial</td>
</tr>
<tr>
<td>All orchestration order lines are canceled.</td>
<td>Canceled</td>
</tr>
<tr>
<td>Some orchestration order lines are canceled.</td>
<td>Ignore canceled orchestration order lines, and determine status based on the open orchestration order lines.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Fulfillment Line Statuses</th>
<th>Orchestration Order Line Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>All fulfillment lines are completed.</td>
<td>Closed</td>
</tr>
<tr>
<td>All fulfillment lines are not completed.</td>
<td>Open</td>
</tr>
<tr>
<td>Some, but not all, fulfillment lines are completed.</td>
<td>Partial</td>
</tr>
<tr>
<td>All fulfillment lines are canceled.</td>
<td>Canceled</td>
</tr>
<tr>
<td>Some fulfillment lines are canceled.</td>
<td>Ignore canceled fulfillment lines, and determine status based on the open fulfillment lines.</td>
</tr>
</tbody>
</table>

### 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 numeric 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.
5. Process name and task type.
6. Task name.
7. Process name and process version.
For task type.

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 the 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 "Usage of Oracle Business Rules (OBR) in DOO Order Fulfillment" in "Oracle Fusion Distributed Order Orchestration White Papers" (1536633.1).

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.
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**

To fulfill nonshippable items, you must create a separate step in the orchestration process in which you connect to a third-party system using the template task layer.

**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.

**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 "Usage of Oracle Business Rules (OBR) in DOO Order Fulfillment" in "Oracle Fusion Distributed Order Orchestration White Papers" (1536633.1).

**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.
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 in Orchestration Process Flows: 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, task type, and service. 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 "Usage of Oracle Business Rules (OBR) in DOO Order Fulfillment" in "Oracle Fusion Distributed Order Orchestration White Papers" (1536633.1).

**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**.
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, including wait steps and pause steps. Lead time 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.


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 BigDecimal.valueOf(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 BigDecimal.valueOf(4) for BigDecimal.valueOf(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

Order Orchestration does the following types of planning:

- **Process planning.** Sets and helps meet the completion date for each step and task in an orchestration process. It transforms each order that it receives into fulfillment lines, creates an orchestration process, and then assigns these fulfillment lines to this orchestration process. It allows you to create a custom process that determines how Order Orchestration automatically plans and manages each step after it receives an order from the order capture system.

- **Process replanning.** Updates completion dates when a change occurs to an order at any point in the orchestration process. Order Orchestration can replan an orchestration process every time it receives an update from the order capture system.
Settings That Affect Orchestration Process Planning

The following setting affects the entire orchestration process:

- **Replan Instantly.** Does replanning immediately after Order Orchestration completes each step in the orchestration process, and then displays the revised order data according to the replanning work that it does. It is recommended that you use **Replan Instantly** for high priority orders, or with orders that include a jeopardy threshold of less than one day. For performance reasons, it is recommended that you do not select **Replan Instantly** for a step that is long or complex. If you do not select **Replan Instantly**, then Order Orchestration updates the planning data only during the scheduled replanning.

The following settings affect planning for each individual step in an orchestration process:

- **Planning Default Branch.** Identifies the default path that Order Orchestration uses for planning. It uses this setting only if the orchestration process includes one or more conditional branches.

- **Fulfillment Completion Step.** Identifies the step where you consider the fulfillment lines as fulfilled. Order Orchestration uses this setting in planning calculations to meet the request date. It is not necessarily the last step in the orchestration process. The chronological last step might not be the last step that the customer cares about. Order Orchestration uses the customer request date as the completion date for the last step in the orchestration process when it plans the orchestration process.

- **Default Lead Time/Lead Time UOM.** Specifies the duration that Order Orchestration requires to complete a unit of work. If you do not define a lead-time expression for a step, then it uses the default lead time.

- **Lead-Time Expression.** Specifies to use Oracle Business Rules to define the lead time. This feature provides flexibility if you define a complex lead-time expression.

How Order Orchestration Calculates Completion Dates

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.

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 today’s date, then cancel and create 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 "Usage of Oracle Business Rules (OBR) in DOO Order Fulfillment" in "Oracle Fusion Distributed Order Orchestration White Papers" (1536633.1).

This example assumes an orchestration process definition with the following steps:

1. Schedule
2. Reserve
3. Ship Goods
4. Wait for Shipping
5. Invoice
6. Wait for Invoice

Orchestration process definitions are created on the Manage Orchestration Process Definitions pages.

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

1. Create the If statement: If the RequestShipDate is today's date.
2. Create the Then statement: Then cancel and create the ShipGoods task.

Creating the If Statement
On the Ship step, create the If statement: If the RequestShipDate is today's date.

1. In the Compensation Rule column of the Ship step, select Click for Rule.
2. Click the Add Rule icon.
3. Optionally, overwrite Rule 1 with a new rule name.
4. Click the arrows to expand the rule.
5. In the If region, click the Left Value icon to search for the left value.
7. Expand DOOFLine.
8. Expand requestShipDate.
10. Click OK.
11. Select Is.
12. Click the Right Value icon.
13. Expand CurrentDate.
14. Expand Date.
15. Select timeInMillis.
16. Click OK.

Creating the Then Statement
Create the Then statement: Then cancel and create the ShipGoods task.

1. In the Then region, click the arrow next to Add Action and select Assert New.
2. In the Select a Target list of values, select DooSeededOrchestrationRules.Result.
3. Select the **Edit Properties** icon.
4. In the Value column of `resultObjKey`, enter "CANCEL_CREATE". You must surround the text with quotation marks.
5. Click **OK**.
6. Click **Validate**.
7. Click **Save**.

**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.

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.
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.


17. Click Insert Into Expression.

18. On the Operators tab, select =.

19. Click Insert Into Expression.

20. On the Tasks tab, expand Schedule.


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.

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.
17. Click **Insert Into Expression**.
18. On the Operators tab, select =.
19. Click **Insert Into Expression**.
20. On the Tasks tab, expand Schedule.
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. You release and deploy the process definition from the Manage Orchestration Process Definition pages. Don’t modify orchestration process definitions outside of these pages.

**Note**

If you want to use the ShipOrderGenericProcess and ReturnOrderGenericProcess predefined orchestration processes, then you must generate them and deploy them. You don’t need to release these processes.

**Release the Orchestration Process Definition**

You release the orchestration process definition on the Manage Orchestration Process Definitions page.
To release the orchestration process definition:

1. Select the process and click Edit.
2. On the Edit Orchestration Process Definition page, select Release from the Actions menu.

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, then 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 the release process continues. An orchestration process is valid when no error messages are generated. A valid process may have associated warning messages. After validation is complete, the status is updated to Released, and the orchestration process definition becomes read-only. At this point, the orchestration process is in Released status, and the BPEL artifacts that are needed to deploy and run the orchestration process are created and stored.

Deploy the Orchestration Process Definition

You deploy the orchestration process definition on the Edit Orchestration Process Definition page. You can release multiple versions of an orchestration process definition sequentially in a single day. You must reject the previous version before you can release the next version on the same day.

To deploy the orchestration process definition:

1. Select Deploy Process from the Actions menu.
2. In the Deploy Process window, the target server details are prepopulated. Enter your password.

If you want to deploy an orchestration process but don’t have access to the Manage Orchestration Process Definition pages, then follow the instructions for deploying the process using Oracle Fusion Middleware.

Deploying an Orchestration Process Definition Using Oracle Fusion Middleware: 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. Use the following instructions to deploy a process definition using Oracle Fusion Middleware. If you have access to the Manage Orchestration Process Definition pages, then you can deploy the process from there instead.

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

1. Release the orchestration process definition.
2. Download the orchestration process definition.
3. Modify the SOA configuration plan.
4. Deploy the JAR file using the modified configuration plan.

Note

If you want to use the ShipOrderGenericProcess and ReturnOrderGenericProcess predefined orchestration processes, then you must generate them and deploy them. It isn't necessary to release them.

Don't 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, then 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 the release process continues. An orchestration process is valid if no error messages are generated. A valid process may have associated warning messages. After validation is complete, the status is updated to Released, and the orchestration process definition becomes read-only. At this point, the orchestration process is in Released status, and the BPEL artifacts that are 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. Use Oracle Fusion Middleware to deploy artifacts.

To deploy the 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

A SOA 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. 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.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAConfigPlan
  xmlns:jca="http://platform.integration.oracle/blocks/adapter/fw/metadata"
  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"/>
      </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,

You can release multiple versions of an orchestration process definition sequentially in a single day. You must reject the
previous version before you can release the next version on the same day.

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.

Creating an Orchestration Process Assignment Rule: Worked Example
This example demonstrates how to create an orchestration process assignment rule that assigns a specific orchestration
process to large orders.
Summary of the Tasks

The example shows how to write a process assignment rule that assigns an orchestration process based on the ordered quantity.

Rules consist of conditions and actions. If the conditions are true, then the actions are taken. In this example, you will create a rule that says: If ordered quantity is large, then assign the LargeOrderProcess orchestration process to the fulfillment line. You will:

1. Create a decision table.
2. Add a new condition for ordered quantity to the decision table. This is the If clause. In this activity, the If clause is: If ordered quantity is Large...
3. Create a bucket set that contains the quantity ranges for comparison to the runtime value.
4. Use the bucket set in the condition that you created.
5. Select an action. This is the Then clause. In this example, the Then clause is: Then assign the orchestration process "LargeOrderProcess" to this fulfillment line.

Navigating to the Manage Orchestration Process Assignment Rules Page

1. On your implementation project, expand the Define Sales Order Fulfillment task list.
2. Expand the Define Orchestration task list.
3. Click the Go to Task icon for the Manage Orchestration Process Assignment Rules task.

Creating a Decision Table

1. Next to the View list, click the arrow next to the Add icon and select Add Decision Table. This action displays a decision table with one rule (R1). Rules are read vertically. You will be adding information to this decision table.
2. Substitute Decision Table 1 with LargeOrdersRule.
3. Click the Save button.

Adding a New Condition to the Decision Table

1. In the View list, select the decision table that you just created.
2. To the left of the decision table name, click the Show Advanced Settings icon.
3. Select the Advanced Mode check box. If other check boxes are also selected, then you may leave them as they are. You are using advanced mode to create aliases and declare instances for some order orchestration variables. Creating aliases can make it faster to use variables when writing a rule. In this case, these particular variables are also aliased in the predefined decision table AssignProcess. The same aliases have been used in this example, but this is not required because they are local references, not global references.
4. Click Collapse Pane to show more of the screen, if needed.
5. In the field that appears to the left of "is a," enter Group.
6. In the list that appears to the right of "is a," select AssignLaunchRules.ResponseRL. You now have an alias for the AssignLaunchRules.ResponseRL. This means you can refer to AssignLaunchRules.ResponseRL as Group.
7. Click the Add Pattern icon.
8. In the field that appears to the left of "is a," enter: fline.
9. In the list that appears to the right of "is a," select AssignLaunchRules.FulfillLineVO. You are creating an alias for AssignLaunchRules.FulfillLineVO. You can now refer to AssignLaunchRules as fline.

10. Directly above the decision table, click the arrow next to the Add icon and select Add Condition.

11. The Condition Browser appears. Notice that the variables that you declared appear as the aliased names.

12. Expand fline.

13. Select OrderedQty, and click the OK button. You have just created part of the If clause: If ordered quantity is equal to X. Next you will create the quantity buckets that you can select to complete the If clause.

Creating a Bucket Set List

1. Click the cell where you created the condition.

2. In the Select Bucketset list, select the Local List of Ranges list item to create a set of values to be compared to the ordered quantity at runtime.

3. Click the Edit Bucketset icon next to Local List of Ranges.

4. In the Name field of the Bucketset Editor window, enter a valid name, such as Bulk Ranges.

5. In the Description field, enter a valid description, such as the following: Set of ranges to use for bulk fulfillment.

6. In the Data Type list, select the int list item. Only data types that match the previously selected condition attribute are displayed.

7. Click the Add Bucket icon.

8. In the Description field for the end point of 1, enter a valid description, such as Small.

9. Click the Add Bucket icon.

10. In the End Point field, change the value to 100.

11. In the Description field, enter a valid description, such as Medium.

12. Click the Add Bucket icon.

13. In the End Point field, change the value to 1000.

14. In the Description field, enter a valid description, such as Large. For this exercise we created three buckets, but you could create more or fewer.

15. Click the OK button.

Using the Bucket Sets in the Condition

1. Double-click in the rule (R1) next to the condition that you created. The cell contains a question mark. A dropdown list appears.

2. In the list select >=1000. Click elsewhere on the page to close the list.

Configuring the Action to Take if the Condition Is True in the New Rule

1. Click the arrow next to the Add icon, and select Add Action and then Modify. The Action Editor appears.

2. In the Target box, select Group. The attributes associated with the view object appear in the Arguments box.

3. Select the check box in the Parameterized column for processName. That indicates that you want to fill in the value separately for each rule.

4. Click the OK button. You are returned to the decision table.

5. Double-click the cell directly to the right of processName:String.
6. Enter "LargeOrderProcess". Quotation marks are required.
7. Select the check box on the line above the cell from the previous step. Selecting the check box means that you want to use this action for this rule.
8. Click the Save button.

Item Data for Use in Oracle Business Rules: Explained
You might want to use item data from Oracle Fusion Product Model in an Oracle Business Rule. For example, you might want to create a rule that assigns a particular orchestration process to a fulfillment line when it contains a particular item category. To maintain maximum performance, Product Model data is not part of the Oracle Fusion Distributed Order Orchestration data model. However, you can call out to Product Model to access additional item data. To use item data, you must set up the criteria to populate it. The item data is populated at run time and then can be referenced in other rules.

You set up the criteria to populate the item data at the time you write the business rule. You go to the page for the business rule that you want to write, for example, Manage Orchestration Process Assignment Rules. You select the facts that you want to populate when you create the rule that fetches the data. Click the Edit Properties window to enter the values.

Selecting Data from the Available Item Facts
When you write the rule to get the item data (getItemData), you specify the fact that you want to populate. You also select Highest priority so that the rule to populate the fact runs before the rule that needs to use the data from this fact.

In the Arguments window, you enter "Yes" in one or more of the following:

- fetchItemDefinition
- fetchItemCategory
- fetchRelatedItem

Also, add the following information in the Arguments window:

- customKey: Used to perform a duplicate check. Recommended value is FulfillLineId.
- inventoryItemId: Used to fetch the Product Model data.
- inventoryOrganizationId: Used to fetch the Product Model data.
- viewRowImpl: This argument is used for external interface routing rules, orchestration process assignment rules, and product transformation rules. Pass FulfillLineVO.viewRowImpl.
- DOOFline: This argument is used for orchestration process definition rules. Pass DooSeededOrchestrationRules.DOOFLine.

Select Additional Data Using a Custom Template
Only a subset of the Item Definition fact is available. If you need data that aren’t available in one of the provided facts, then you can use the getFusionAppsData template to fetch the attribute that you need.

Below are the arguments. The first five arguments are required. The last five arguments are the attributes that you want to populate.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputAttributeName1</td>
<td>String</td>
<td>&quot;InventoryItemld&quot;</td>
</tr>
</tbody>
</table>
### Creating an Orchestration Process Assignment Rule Using Item Data: Worked Example

This example shows how to create a rule that assigns a specific orchestration process to a group of one or more fulfillment lines associated with a particular customer if the item category on the fulfillment line is 3574.

#### Prerequisites

This example assumes that an order administrator created an orchestration process called ManualShipOrderGeneric123. It also assumes that the organization has product model installed and an item category called 3574 exists.

#### Summary of the Tasks

For this example, you must create two rules. The first rule will get the item data when the fulfillment lines are associated with a particular customer. Select the highest priority to ensure that this rule runs first. The second rule assigns orchestration process ManualShipOrderGeneric123 to a group if the item category ID is 3574.

1. Create a rule that makes item category data available if ShipToCustomerId is 1304523.
2. Create a rule that assigns orchestration process ManualShipOrderGeneric123 to a fulfillment line that has item category 3574.

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputAttributeValue1</td>
<td>String</td>
<td>RuleType. FulfillLineVO. AttributeYouWanttoPopulate</td>
</tr>
<tr>
<td>inputAttributeName2</td>
<td>String</td>
<td>&quot;InventoryOrganizationId&quot;</td>
</tr>
<tr>
<td>inputAttributeValue2</td>
<td>String</td>
<td>RuleType. FulfillLineVO. AttributeYouWanttoPopulate</td>
</tr>
<tr>
<td>viewObjectName</td>
<td>String</td>
<td>&quot;ItemPVO&quot;</td>
</tr>
<tr>
<td>outputAttributeName1</td>
<td>String</td>
<td>&quot;AttributeYouWanttoPopulate&quot;</td>
</tr>
<tr>
<td>outputAttributeName2</td>
<td>String</td>
<td>&quot;AttributeYouWanttoPopulate&quot;</td>
</tr>
<tr>
<td>outputAttributeName3</td>
<td>String</td>
<td>&quot;AttributeYouWanttoPopulate&quot;</td>
</tr>
<tr>
<td>outputAttributeName4</td>
<td>String</td>
<td>&quot;AttributeYouWanttoPopulate&quot;</td>
</tr>
<tr>
<td>outputAttributeName5</td>
<td>String</td>
<td>&quot;AttributeYouWanttoPopulate&quot;</td>
</tr>
</tbody>
</table>

At run time, all data types of your attribute appear in the Condition Browser. When you write your rule, you must provide the correct data type for that attribute. Consult Oracle Enterprise Repository for the data types and additional available columns. Log in as guest, and search for Item SDO.
Creating a Rule That Makes Item Data Available

Creating the If Statement

Create the following If statement: If the ship-to customer ID is 1304523.

1. Click the arrow next to the Add icon and select Add Rule.
2. Replace Rule 1 with a valid name, such as GetItemData.
3. Click the Expand arrow.
4. Click the Show Advanced Settings arrows.
5. Next to Priority, select Highest.
6. Click the Collapse Pane icon to reveal more of the screen, if needed.
7. In the field below If, click the left value icon.
8. In the Condition Browser, expand AssignLaunchRules. Expand FulfillLineVO, and select ShipToCustomerId.
9. Click OK to close the Condition Browser.
10. Select Is from the list of operators.
11. In the field to the right of the operators enter 1304523.

Creating the Then Statement

Create the following Then statement: Then get item data.

1. In the Then region, click the arrow next to Add Action and select Call.
2. In the list, select AssignLaunchRules.getItemData.
3. Click the Edit Properties icon. The Arguments window opens.
4. In the Value column of each of the following names enter:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>inventoryItemId</td>
<td>AssignLaunchRules. FulfillLineVO. inventoryItemId</td>
</tr>
<tr>
<td>inventoryOrganizationId</td>
<td>AssignLaunchRules. FulfillLineVO. inventoryOrganizationId</td>
</tr>
<tr>
<td>fetchItemDefinition</td>
<td>&quot;No&quot;</td>
</tr>
<tr>
<td>fetchItemCategory</td>
<td>&quot;Yes&quot;</td>
</tr>
</tbody>
</table>
5. Click **OK** to close the Arguments window.

Creating a Rule That Uses the Item Category

Creating the If Statement

Create the following If statement: If the item category ID is 3524.

1. Click the arrow next to the Add icon and select Add Rule.
2. Replace Rule 2 with a valid name, such as AssignShipOrderGenericIf3524Category
3. Click the Expand arrow.
4. Click the Show Advanced Settings arrows.
5. Click Collapse Pane to reveal more of the screen, if needed.
6. Select Advanced Mode.
7. In the left value field below If, enter: Group.
8. To the right of "is a", select AssignLaunchRules.ResponseRL.
9. In the test directly below your Group declaration, select the arrow next to the right value icon, and select Delete Test.
10. To the right of "and" in your Group declaration, select Add Pattern.
11. In the field below "and," enter: FLine.
12. To the right of "is a" select AssignLaunchRules.FulfillLineVO.
13. To the right of AssignLaunchRules.FulfillLineVO,
14. In the left value field of the test below the FLine alias that you created, click the Left Value icon.
15. In the Condition Browser, expand FLine and select ShipToCustomerId.
16. Click **OK** to close the Condition Browser.
17. In the list of operators, select Is.
18. In the right value field, enter: 1304523.
19. To the right of the line where you created the alias, click the Add Pattern icon.

20. In the left value field below "And," enter an alias, such as: ItemCat. Quotation marks are not necessary.

21. To the right of "is a", select AssignLaunchRules.ItemCategory.

22. Below the alias, click the Add Test arrow and select Simple Test.

23. Click the left value icon.

24. In the Condition Browser, select ItemCat.

25. Click OK to close the Condition Browser.

26. In the list of operators, select Isn’t.

27. Click the right value icon and select: null.

28. Click the arrow next to right value icon, and select Simple Test.

29. Next to the new left value field, click the left value icon.

30. In the Condition Browser, expand ItemCat and select categoryld.

31. Click OK to close the Condition Browser.

32. Select Is.

33. In the field to the right of the operators, enter: 3524L.

34. Click the arrow next to the right value icon, and select Simple Test.

35. Next to the new left value field, click the left value icon.

36. In the Condition Browser, expand ItemCat and select customKey.

37. Click OK to close the Condition Browser.

38. Select Is.

39. To the right of Is, click the right value icon.

40. In the Condition Browser, expand FLine.

41. Expand FulfillLineId, and then select toString.

42. Click OK to close the Condition Browser.

Creating the Then Statement
Create the following Then statement: Then assign orchestration process ManualShipOrderGeneric123.

1. Click the arrow next to Add Action and select Modify.

2. In the list select Group.

3. Click the Edit Properties icon to open the Properties window.
4. In the Value field next to processName, enter: “ManualShipOrderGeneric123”.

5. Click OK to close the Properties window.

6. Click Save.

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.

Processing Constraint Definition: Explained

When you create a processing constraint you can define what changes are allowed to be made, when, and by whom. You define processing constraints on the Manage Processing Constraints page. You can use predefined record sets, validation rule sets, and constraint entities; or you can create additional ones. You can view the list of existing record sets and validation rule sets and create additional ones on the tabs of the Manage Processing Constraints page. You view existing constraint entities and create additional ones on the Manage Constraint Entities page.

You can create a processing constraint without conditions. However, if you don’t assign conditions, then the constraint is always true. For example, consider the predefined processing constraint that prohibits deletions of an order. It prohibits deletions under all circumstances.

Note

Generate a constraint package on the Manage Processing Constraints page or run the scheduled process anytime you create or modify a record set or a table type validation rule set. You don’t need to generate a constraint package for any other validation type.

Record Sets

A record set is a group of records that are bound by some common attributes values for the purpose of evaluating a constraint. For example, you might want to evaluate all orders for a particular customer. You create a record set by specifying the entity to evaluate: Order header, order line, or order fulfillment line. Then you may refine the record set further by selecting an attribute. For example, to evaluate all orders for a particular customer, select the Order Header entity and then select the Sold-to Customer attribute.

Validation Rule Set

A validation rule set is a grouping of one or more If clauses. For example, let’s say you want to prohibit any changes on an order that is closed. The If-then statement would be something like: If the order is closed, then reject the change. Select the Order is Closed validation rule set, which checks the Open attribute on the order header to see if the value is N. The Order is
Closed validation rule set is one of several predefined validation rule sets. You can create additional validation rule sets, but you can’t change or delete any predefined ones.

**Note**
You can invert a validation rule set, which means that the processing constraint is applied when the condition is not true.

**Constraint Entity**
A constraint entity is a business object that is affected by a processing constraint. For example, you might want to prohibit changes to fulfillment lines under certain circumstances. You can create process task entities, but you can’t create view entities.

**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
A business object that a processing constraint is applied to. During this step, you create a process task entity so that it can be used later to create a processing constraint. A process task entity can be the combination of an orchestration process, a task, and a service.

**Note**
For this example, you are using a process task entity, but predefined view entities are also available. You cannot add new view entities.

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 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 for the purpose of constraint evaluation. 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**.

**Using Extensible Flexfields in Processing Constraints: Explained**

You can create processing constraints using extensible flexfields. Processing constraints control the types of changes that can be made, when they are made, and by whom. For example, you might want to use processing constraints to prohibit an update to an extensible flexfield if a fulfillment line is closed or to require that an extensible flexfield is populated at a particular step of the orchestration process.

In most ways, using extensible flexfields isn’t different from using other business objects when creating processing constraints. You enable extensible flexfields on the Manage Constraint Entities page, so that you can use them on the Manage Processing Constraints page to create a processing constraint. If extensible flexfields are configured, then they are available on these pages. Extensible flexfields aren’t available, however, on the Attributes menu of the Record Sets tab.

Note these additional requirements for using extensible flexfields:

- You must enable the extensible flexfields before using them.
You can’t select different contexts in a validation rule set.

Enabling Flexfields

Before you enable extensible flexfields you must configure them and run the Publish Extensible Flexfield Attributes scheduled process. At this point, extensible flexfields appear on the Manage Constraint Entities page, but they aren’t enabled. You must enable the ones that you intend to use in processing constraints.

Selecting Extensible Flexfields for a Validation Rule Set

You can insert multiple lines in the detail of a validation rule set. All the lines are evaluated together. You can’t select different contexts of an extensible flexfield in a validation rule set. If you select two context and segment attributes, then the attributes must share the same context value; that is, the context is mutually exclusive.

For example, consider the following context:segment combinations:

- Dealer Information: ID
- Dealer Information: Location
- Warranty Information: ID

You can’t select ID for both Dealer Information and Warranty Information contexts together.

Creating a Processing Constraint Using an Extensible Flexfield: Worked Example

This example demonstrates how to create a processing constraint that prevents changes to an extensible flexfield when a fulfillment line is already closed.

Summary of Requirements

These are the main tasks for creating this processing constraint:

1. Enable the extensible flexfield.
2. Create a validation rule set.
3. Create a processing constraint that prohibits changes to the Subcontractor:ID extensible flexfield when the fulfillment line is closed.

Enabling the Extensible Flexfield

Enable an extensible flexfield for use in creating a processing constraint.

1. On the Manage Constraint Entities page, select Entity Type Equals and View Entity.
2. Click Search.
3. In the Search Results region, select Order Fulfillment Line.
4. In the Attribute Details region, locate the Subcontractor:ID extensible flexfield and select Constraint-Enabled.
5. Click **Save and Close**.

**Creating a Validation Rule Set**

1. On the Manage Processing Constraints page, select the Validation Rule Set tab.
2. In **Name**, enter: Fulf Line Closed.
3. In the **Description** field, enter Fulfillment line is closed.
4. In the **Short Name** field, enter FLCLOSE.
5. In the Validation Type list, select Table.
6. In the Entity list, select Order Fulfillment Line.
7. Click **Save**.
8. In the Details region, click the Add Row icon.
9. In the Attribute Name list, select Open.
10. In the Validation Operation list, select: Equal to.
11. In Value String, enter: N
12. Click **Save**.
13. Click **Generate Packages** to start the background process that makes the Validation Rule Set active and available for use in constraints.

**Creating a Processing Constraint**

1. Click the Constraints tab.
2. Click the Add Row icon.
3. In the **Name** field, enter: Constrain Subcontractor ID.
4. In the Constraint Entity list, select Order Fulfillment Line.
5. In the Constrained Operation list, select Update.
6. In the Attribute Name list, select Subcontractor:ID.
7. Click **Save**.
8. In the Conditions tab of the Details region, enter 100 in the **Group Number** field.
9. In the Validation Entity list, select Order Fulfillment Line.
10. In the Validation Rule Set list, select Fulf Line Closed.
11. In the Scope list, select Any.
12. In the Record Set list, select: Fulfillment Line Default Record Set.
13. In the **Message** field, enter: Fulfillment line is closed.

14. Click **Save**.

**FAQs for Define Processing Constraints**

**What's a constraint package?**

A constraint package is a set of triggers that is applied to the appropriate table in the database. Constraint packages activate new and modified table type validation rule sets and new and modified record sets for processing constraints. When you generate a constraint package you are starting a background process that sets the triggers.

Generate a constraint package on the Manage Processing Constraints page or run the Generate Constraint Packages scheduled process at the beginning of any implementation to generate the predefined records. After that, generate constraint packages whenever you create or modify a record set or a table type validation rule set. You do not need to generate a constraint package for any other validation type.

**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.

**Note**
The Sales Order enterprise business object and front-end connector accept only the internal IDs of business objects.

**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, transactional 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.

**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.

---

**Note**
When you define items you must also create item substitution rules. Keep in mind that Oracle Fusion Distributed Order Orchestration does not validate item substitutions. For example, assume that a fulfillment line has a standard item. The order manager checks the availability, and order promising presents several different options. If the item substitution rules are not set up carefully, then the order manager could select an option with an inappropriate substitution, for example, a kit. The application does not forbid this selection.
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
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 a Product Transformation Rule with a Bucket Set: Worked Example
This example shows how to create a product transformation rule with a bucket set in a decision table.

Summary of the Tasks
Create a bucket set that contains the values you will select when creating a product transformation rule that represents the following offer:
Customers who order an printer model T0115 get expedited shipping.

This example includes the following tasks:

1. Navigate to the Manage Product Transformation Rules Page
2. Create a Bucket Set List
3. Create a Decision Table
4. Add a New Condition to the Decision Table
5. Add a New Rule to the Decision Table
6. Configure the Action to Take if the Condition Is True in the New Rule

Navigating to the Manage Product Transformation Rules Page

1. On your implementation project, expand the Define Sales Order Fulfillment task list.
2. Expand the Define Transformation Details task list.
3. Click the Go to Task icon next to Manage Product Transformation Rules.

Creating a Bucket Set List
Oracle Business Rules uses long item IDs from the database. Users might not recognize the long IDs, so create buckets that list the item IDs and aliases.

Note
In this example, you will create two buckets but select only one.

1. Click Bucketsets at the top left of the page.
2. Click the arrow next to the Add Bucketset icon and select List of Values. A new bucket set with the placeholder text Bucket Set 1 appears.
3. Select Bucket Set 1, if it is not selected already. The Edit Bucketset icon appears above the table. If it does not appear, click the Expand Pane icon, or press F5.

4. Click the Edit Bucketset icon. The Bucketset Editor window appears.

5. In the Name field, overwrite the existing text and enter Printer IDs.

6. Enter the following description: List of printer item IDs and aliases for selection in product transformation rules.

7. In the Data Type list, select string.

8. Click the Add Bucket icon. A new Value field appears

9. In the Value field, replace Bucket 1 with 200000003357223L. This is the item ID from the database. L denotes a long value.

10. In the Alias field (which was populated with the same number as the Value field), overwrite the existing number with T0115. This is the item code that is on the customer catalog and is used throughout the company.

11. In the Description field, enter T0115 printer. It is possible to create more buckets, but you need only one for this example.

12. Click OK.

13. Click Save.

Creating a Decision Table

1. Next to the View list, click the arrow next to the Add icon and select Add Decision Table from the list.

2. Replace Decision Table 1 with AddExpeditedShippingforProductT0115.

3. Click Save. The page refreshes.

4. In the View list, select your decision table.

5. Click the Collapse Pane icon, if necessary, to see more of the page.

Adding a New Condition to the Decision Table

1. Scroll down to the table.

2. Directly above the table, click the arrow next to the Add icon, and select Add Condition. The Condition Browser window opens.

3. Expand the OrderTransformationRules list item. The new condition will be based on the item ID.

4. Expand FulfillLineVO.

5. Expand InventoryItemld.

6. Select toString.

7. Click OK.
Adding a New Rule to the Decision Table

1. In your decision table, click the field that shows your condition. A field with a list of values appears.
2. Click the arrow next to Local List of Values.
3. Select the bucket set that you created. Only bucket sets of the matching data type appear in the list.
4. Double-click in the R1 column for the condition. This cell has a question mark. A dropdown list appears.
5. Click the dropdown list arrow.
6. Select T0115. Click elsewhere on the page to close the selection list.

Configuring the Action to Take if the Condition Is True in the New Rule

1. Click the arrow next to the Add icon and select Add Action.
2. Select Modify. The Action Editor opens.
3. In the Target list, select OrderTransformationRules.FulfillLineVO.
4. Press the Page Down key to see more arguments.
5. Click in the Arguments list, and then press the Page Down key to see more arguments.
6. Next to shipClassOfService select the Parameterized check box. You will have to press Page Down about five times to get to shipClassOfService.
7. Click OK.
8. Click the Collapse Pane icon, if necessary, to see more of the page.
9. Double-click the cell to the right of shipClassOfService.
10. Enter "Expedited". Note that quotation marks are required.
11. Click Save.

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.

Summary of the Requirements

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 delete fulfillment line 1 and fulfillment line 2.

Creating an Advanced Transformation Rule


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 InventoryItemld.

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 InventoryItemld.

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 Assert New.

51. In the Target list of values, select OrderTransformationRules.DeleteOrderLine.

52. Click the Edit Properties icon.

53. Click the value icon for fulfillmentLineId.

54. In the Condition Browser, expand OrderTransformationRules and FulfillLineVO.Fulfill.

55. Select FulfillLineId.
56. Click OK to close the Condition Browser.

57. Click the value icon for viewRowImpl. enter OrderTransformationRules.FulfillLineVO.Fulfill.

58. In the Condition Browser expand OrderTransformationRules and FulfillLineVO.Fulfill.

59. Select FulfillLineId.

60. Click OK to close the Condition Browser.

61. Click the Insert Action icon.

62. In the Action list of values, select Retract.

63. In the Target list of values, select FulfillLine2.

64. 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.

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.

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.

model
Fulfillment line that has child fulfillment lines.

OLTP
Abbreviation for online transaction processing, which refers to real-time processing of large amounts of relatively short database transactions.

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.

**start-after condition**
Rules that govern whether the Pause task establishes a pause and how the pause can be released.

**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.