Contents

Preface i

1 Introduction 1
   What’s New in Implementing Order Management: Explained 1
   Order Management Cloud: Overview 5
   Order Management Architecture 9

2 Implementing Order Management Cloud 17
   Implementing Order-to-Cash 17
   Implementing Business-to-Business Order Management 38
   Implementing Drop Ship 47
   More Details About Implementing Order Management 64

3 Integrating Order Management Cloud 69
   Integrating Order Management with External Systems: Overview 69
   Integrating Order Management with Upstream Source Systems 70
   Integrating Order Management with Downstream Fulfillment Systems 74
   Using Integration Cloud Service to Integrate Order Management with External Systems 86
   Integrating Order Management with Other Oracle Applications 102
   Integrating Order Management with Transportation Management 145
   More Details About Integrating Order Management 152

4 Importing and Transforming Source Orders 155
   Importing Source Orders 155
   Transforming Source Orders to Sales Orders 165
## 5 Controlling Order Processing

Controlling Application Behavior .......................................................... 187
Managing Statuses for Sales Orders ....................................................... 192
Constraining Sales Orders .................................................................... 206
Managing Change Orders ..................................................................... 216
Using Jeopardy to Manage Delays ........................................................ 235
Setting Up Fields ................................................................................ 240
Managing Order Types: Procedure ...................................................... 244

## 6 Setting Up Features

Defining Extensions That Modify Order Management Cloud: Overview ........................................................................ 249
Managing Credit, Payment, and Accounting ....................................... 249
Managing Sales Order Approvals ......................................................... 256
Managing Trade Compliance ............................................................... 263
Managing Coverages and Subscriptions .............................................. 276

## 7 Setting Up Orchestration Processes

Defining Orchestration Processes ......................................................... 303
Defining Subprocesses In Orchestration Processes ................................. 321
Assigning Orchestration Processes ....................................................... 333
Pausing Orchestration Processes ........................................................ 339
Setting Up Task Types ........................................................................ 368

## 8 Setting Up Business Rules

Using Business Rules With Order Management: Overview .................. 379
Getting Data from Different Product Models When Using Oracle Business Rules: Procedure ................................... 379
Using Visual Information Builder to Create Business Rules ............... 385

## 9 Performing Other Implementation Work

Administering Messaging and Social Networking .................................. 399
Administering Collaboration Messaging .............................................. 401
Preface

This preface introduces information sources that can help you use the application.

Oracle Applications Help

Use the help icon "?” to access Oracle Applications Help in the application. If you don't see any help icons on your page, click the Show Help icon "?” in the global header. Not all pages have help icons. You can also access Oracle Applications Help at https://fusionhelp.oracle.com.

Using Applications Help

Watch: This video tutorial shows you how to find help and use help features.

Additional Resources

- **Community:** Use Oracle Applications Customer Connect to get information from experts at Oracle, the partner community, and other users.
- **Guides and Videos:** Go to the Oracle Help Center to find guides and videos.
- **Training:** Take courses on Oracle Cloud from Oracle University.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at Oracle Accessibility Program.

Access to Oracle Support

Oracle customers that have purchased support 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.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send e-mail to: oracle_fusion_applications_help_ww_grp@oracle.com.
# Introduction

## What's New in Implementing Order Management: Explained

This topic lists each help topic that is new or significantly revised for each release.

**Release 13 (update 17D)**

The following topics are new or significantly revised. This table lists the new topics first, and then the revised topics.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controlling Decimal Precision: Procedure</td>
<td>New topic. You can use the Quantity Decimal Precision profile to control decimal precision for attributes that store numeric values.</td>
</tr>
<tr>
<td>Specifying Transaction Types to Send from Order Management to Oracle Receivables: Procedure</td>
<td>New topic. You can define a business rule that sets the value for the AR Transaction Type attribute according to details from the sales order, such as business unit, order type, line type, or item.</td>
</tr>
<tr>
<td>The following topics describe how to integrate Order Management with other Oracle applications:</td>
<td>Revised topics. These topics previously described only how to integrate Order Management with Oracle Receivables. They now also describe how to integrate Order Management with Oracle Shipping, Oracle Receiving, and Oracle Procurement.</td>
</tr>
<tr>
<td>• Integrating Order Management with Other Oracle Applications: Overview</td>
<td></td>
</tr>
<tr>
<td>• Integrating Order Management with Other Oracle Applications: Procedure</td>
<td></td>
</tr>
<tr>
<td>• Using Descriptive Flexfields to Integrate Order Management with Other Oracle Applications: Procedure</td>
<td></td>
</tr>
<tr>
<td>• Using Extensible Flexfields to Integrate Order Management with Other Oracle Applications: Procedure</td>
<td></td>
</tr>
<tr>
<td>Order Import Template: Explained</td>
<td>Revised topic. You must make sure you correctly set up the rank for each customer item relationship before you import source orders.</td>
</tr>
<tr>
<td>Pausing Orchestration Processes: Explained</td>
<td>Revised topic. You must make sure to release a paused orchestration process. If you do not, then the orchestration process will remain in a paused state, which might result in failed sales orders, failed order fulfillment, and so on.</td>
</tr>
<tr>
<td>Managing Approval Rules for Sales Orders: How It Works</td>
<td>Revised topic. It is recommended that you define your approval rule so it evaluates to True only one time for the same sales order, and then creates only one approval request.</td>
</tr>
</tbody>
</table>
Managing Change That Occurs During Order Fulfillment: Explained
Revised topic. The compensation pattern for the Shipping task uses the Cancel and Create pattern for item substitution during shipping but does not support the Update pattern. If you require different behavior, then you can write a compensation rule according to an item change.

Release 13 (update 17C)
The following topics are new or significantly revised. This table lists the new topics first, and then the revised topics.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining Extensions That Modify Order Management Cloud: Overview</td>
<td>Use the Manage Extensions page in the Setup and Maintenance work area to write your own Groovy script that modifies your Order Management deployment, and to specify the extension point that determines when to run this script.</td>
</tr>
<tr>
<td>The following topics describe how to integrate Order Management with Oracle Receivables:</td>
<td>Use a descriptive flexfield or an extensible flexfield to communicate details between Order Management Cloud and Oracle Receivables.</td>
</tr>
<tr>
<td>• Integrating Order Management with Oracle Receivables: Overview</td>
<td></td>
</tr>
<tr>
<td>• Integrating Order Management with Oracle Receivables: Procedure</td>
<td></td>
</tr>
<tr>
<td>• Using Descriptive Flexfields to Integrate Order Management with in Oracle Receivables: Procedure</td>
<td></td>
</tr>
<tr>
<td>• Using Extensible Flexfields to Integrate Order Management with in Oracle Receivables: Procedure</td>
<td></td>
</tr>
<tr>
<td>The following topics describe how to set up coverages and subscriptions:</td>
<td>These topics describe how to do the following work:</td>
</tr>
<tr>
<td>• Setting Up Coverage for Sales Orders: Procedure</td>
<td>• Set up and manage coverage items. For example, you can set up a coverage, such as a warranty for laptop support that includes a daily, weekly, monthly or yearly duration.</td>
</tr>
<tr>
<td>• Setting Up Orchestration Processes for Coverage Items: Explained</td>
<td>• If you set up an orchestration process that processes a covered item or a coverage item, then you can use the same orchestration process instance for the covered item and for the coverage item, depending on your business requirements. If you use different instances, then you can coordinate these instances.</td>
</tr>
<tr>
<td>• Setting Up Subscriptions for Sales Orders: Procedure</td>
<td>• Set up subscriptions for sales orders.</td>
</tr>
<tr>
<td>• Setting Up Subscriptions for Covered Items: Procedure</td>
<td>• Set up a subscription for a covered item. In this example, you define a covered item for cloud up time, and then define a subscription that allows the Order Entry Specialist to add this up time.</td>
</tr>
<tr>
<td>• Importing Source Orders That Include Coverage or Subscriptions: Procedure</td>
<td>• Import a source order that includes covered items, coverage items, or subscriptions into Order Management.</td>
</tr>
<tr>
<td>Specifying How to Track Items as Assets: Procedure</td>
<td>Specify how to track an item as an asset, and then allow the Order Entry Specialist to monitor it during order fulfillment.</td>
</tr>
</tbody>
</table>
## Release 13 (update 17B)

The following topics are new or significantly revised. This table lists the new topics first, and then the revised topics.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The following topics describe how to set up trade compliance:</td>
<td>Use Order Management in conjunction with your trade management solution to create and manage trade transactions that comply with import global trade rules, export global trade rules, and other trade regulations that the country or region requires.</td>
</tr>
<tr>
<td>• Managing Trade Compliance: Overview</td>
<td></td>
</tr>
<tr>
<td>• Managing Trade Compliance: How it Works</td>
<td></td>
</tr>
<tr>
<td>• Managing Trade Compliance: Procedure</td>
<td></td>
</tr>
<tr>
<td>• Managing Trade Compliance During Order Fulfillment: Procedure</td>
<td></td>
</tr>
<tr>
<td>• Constraining Compliance Screening: Procedure</td>
<td></td>
</tr>
<tr>
<td>• Managing Trade Control Types: Procedure</td>
<td></td>
</tr>
<tr>
<td>The following topics describe how to integrate Order Management with Transportation Management:</td>
<td>Integrate Order Management with Oracle Transportation Management so that it sends sales orders to a transportation management system for transportation planning, and to receive updates at fulfillment milestones, such as plan completion or proof of delivery.</td>
</tr>
<tr>
<td>• Integrating Order Management with Transportation Management: Overview</td>
<td></td>
</tr>
<tr>
<td>• Integrating Order Management with Transportation Management: How it Works</td>
<td></td>
</tr>
<tr>
<td>• Integrating Order Management with Transportation Management: Procedure</td>
<td></td>
</tr>
<tr>
<td>The following topics describe how to manage approval rules:</td>
<td>Configure Order Management to route each sales order for sales order approval. You can configure sales order approval to accept or reject the sales order according to a variety of approval rules, and on a variety of sales order attributes.</td>
</tr>
<tr>
<td>• Managing Approval Rules for Sales Orders: How it Works</td>
<td></td>
</tr>
<tr>
<td>• Managing Approval Rules for Sales Orders: Procedure</td>
<td></td>
</tr>
<tr>
<td>The following topics describe how to manage credit check:</td>
<td>Use credit check in Order Management to minimize the financial risk that your organization assumes during day-to-day operations. You can use credit check to validate each sales order and verify that your customer has credit available that is sufficient to cover the cost of the purchase, and to allow Order Management to process the sales order through order fulfillment before receiving payment.</td>
</tr>
<tr>
<td>• Managing Credit Check in Order Management: How it Works</td>
<td></td>
</tr>
<tr>
<td>• Managing Credit Check in Order Management: Procedure</td>
<td></td>
</tr>
<tr>
<td>• Importing Source Orders That Include Credit Check Details: Described</td>
<td></td>
</tr>
<tr>
<td>Introduction to Using Business Rules with Order Management Cloud</td>
<td>This white paper provides an introduction to how to create an Oracle Business Rule. It also includes reference details, such as functions, date formats, and so on.</td>
</tr>
<tr>
<td>Topic</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Calling Order Management Cloud from Client Applications</td>
<td>Oracle Fusion Applications exposes its interfaces as web services, and deploys them on a service-oriented architecture (SOA) container, secured through layers of web service policy, authentications, and other protocols. This document describes how to set up a client application so it can call secure web services on Order Management Cloud. For details, search for Document ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at <a href="https://support.oracle.com">https://support.oracle.com</a> and then download the attachment that describes how to call Order Management Cloud from a client application.</td>
</tr>
<tr>
<td>Using Web Services with Order Management Cloud</td>
<td>You can use web services to interact with Order Management. For example, you can use the Order Import web service to create an integration that sends order requests from your upstream system to Order Management. This web service processes the request, and then creates a sales order in Order Management. Various types of upstream systems and applications can use this service, such as channels, legacy systems, quoting systems, contract systems, service request systems, inventory systems, or purchasing systems. For details, search for Document ID 2051640.1 (Using Web Services with Oracle Fusion Order Management Cloud) on My Oracle Support at <a href="https://support.oracle.com">https://support.oracle.com</a>.</td>
</tr>
<tr>
<td>Integrating Order Management Cloud with Oracle eBusiness Suite</td>
<td>Get help if you already own applications that are part of Oracle E-Business Suite, Enterprise Resource Planning (Oracle EBS ERP), and you plan to migrate to Order Management Cloud in phases and will continue to use E-Business Suite to perform some of the fulfillment functionality. For details, search for Document ID 2132744.1 (Integrating Order Management Cloud with Oracle eBusiness Suite) on My Oracle Support at <a href="https://support.oracle.com">https://support.oracle.com</a>.</td>
</tr>
<tr>
<td>Managing Order Types: Procedure</td>
<td>An order type is an optional attribute on the order header that you can administer to arrange sales orders into groups that you define. You can also use it to control order processing.</td>
</tr>
<tr>
<td>Managing Order Management Parameters: Procedure</td>
<td>Revised topic. Includes parameters that you can set for credit check, trade compliance, and order approvals.</td>
</tr>
<tr>
<td>Getting Data from Different Product Models When Using Oracle Business Rules: Procedure</td>
<td>Revised topics. Starting in Release 12, you use a different set of objects in your business rule.</td>
</tr>
<tr>
<td>Routing Requests to Fulfillment Systems: Procedure</td>
<td></td>
</tr>
<tr>
<td>Routing Requests to Fulfillment Systems When Cross-Reference Data Does Not Exist: Procedure</td>
<td></td>
</tr>
<tr>
<td>Creating Transformation Rules: Procedure</td>
<td>Revised topics. These topics include revised procedural instructions and screen captures.</td>
</tr>
<tr>
<td>Creating Advanced Product Transformation Rules: Procedure</td>
<td></td>
</tr>
<tr>
<td>Assigning Orchestration Processes: Procedure</td>
<td></td>
</tr>
</tbody>
</table>
### Order Management Cloud: Overview

Order Management Cloud is a Supply Chain Management application that improves order fulfillment for your business processes. It includes predefined integration, centrally managed orchestration policies, global availability, and fulfillment monitoring that can help increase customer satisfaction and order profitability.

To get the latest details about Order Management, including What’s New in each release and release content documents, see https://cloud.oracle.com/saasreadiness/scm.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connecting Order Management to Fulfillment Systems: Procedure</td>
<td>Revised topic. Set the Invocation Mode to determine how Order Management calls the connector when an orchestration process requires an interface to a system that resides outside of Order Management.</td>
</tr>
<tr>
<td>Implementing Order-to-Cash: Points to Consider</td>
<td>Revised topic. Includes content that describes whether new features in this release come predefined or require setup.</td>
</tr>
<tr>
<td>Implementing Order-to-Cash: Quick Start</td>
<td>Revised topic. Includes revised setup details.</td>
</tr>
<tr>
<td>Assigning Job Roles in Order Management: Explained</td>
<td>Revised topic. Includes details about predefined job roles.</td>
</tr>
<tr>
<td>Using Drop Ship with Order Management: How it Works</td>
<td>Revised topic. Includes a list of setups that are required to implement drop ship.</td>
</tr>
</tbody>
</table>
Capture Orders and Fulfill Them

The following diagram illustrates how you can use Order Management to capture customer demand and fulfill sales orders.

Key

EDI: Electronic Data Interchange
POS: Point of Sale
3PL: Third-Party Logistics
WMS: Warehouse Management System
CM: Contract Manufacturing

Explanation of Callouts
You can use Order Management to capture customer demand and fulfill sales orders in the following ways:

1. **Capture customer demand.** Capture demand from various channels, such as web, mobile, call center, direct sales, and partners. Provide the functionality that your users can use during order capture, such as pricing items, determining availability, getting the order status, and so on.

2. **Orchestrate and monitor fulfillment across channels.** Coordinate with other Oracle Supply Chain Management Cloud applications during fulfillment, such as Purchasing, Manufacturing, Inventory Management, and other solutions.

3. **Use web services.** Integrate Order Management with various systems that reside outside of the Order Management solution. For details, see the Integrating Order Management chapter in the guide titled Oracle SCM Cloud, Implementing Order Management at https://docs.oracle.com/cloud/latest/scmcs_gs/FAIOM/FAIOM2061947.htm#FAIOM2061947.

### Manage Multichannel Orders

Order Management supports order-to-cash processes that are global and that use multiple channels. You can use it as the order hub for your organization to centralize and manage multiple order capture channels, do order promising, orchestrate fulfillment policies, monitor order status, and manage exceptions. You can enter, price, and configure sales orders directly in Order Management. You can receive source orders from external sources, modify them in Order Management, and then process them for fulfillment. You can use predefined integrations with other Oracle cloud services to centrally manage orchestration policies, get global availability, monitor statuses, and manage exceptions:

- **Enter sales orders.** Enter sales orders directly in Order Management. You can enter details for a new sales order, revise a sales order, modify order lines, view change history, place a sales order on hold, or cancel a sales order. You can also create a return order for an existing sales order.

- **Import source orders.** Import source orders from an external capture system and then edit them. For example, you can import source orders from an e-commerce system, and then modify them in Order Management.

- **Use common definitions.** Use a common definition of sales orders across channels so that you can view and search them in a consistent way. For example, a user can search for Item A across sales orders that other users created in Order Management, and across sales orders that an administrator imported from an e-commerce site, call center, or through electronic data interchange (EDI).

- **Use orchestration fulfillment processes.** An orchestration process routes and manages sales orders across multiple fulfillment systems. For example, you can use a schedule, ship, and bill process to route order lines to two different enterprise resource planning (ERP) systems.

- **Get status updates.** Send normalized status updates from multiple fulfillment systems to external capture systems. For example, return a SHIPPED order status to an e-commerce system even if one warehouse system returns a Shipped status for an order line, but another warehouse returns an SHP status for the same line.

- **Get a summary view of statuses and exception orders.** For example, you can view a graph that displays a summary of sales orders that are at risk of missing a promise date.

- **Configure order promising.** Use Oracle Fusion Global Order Promising to collect supply data from multiple sources and to configure rules that select the best fulfillment location for demand from any channel. You can select according to future availability, expected delivery date, and preferred delivery method. Allocate scarce supply according to customer, channel, or to resolve order exceptions.

- **Use heterogeneous deployment.** Use a mixture of cloud and on-premise environments for your capture systems and fulfillment systems. For example, you can import source orders from a cloud capture system, and then fulfill them to an external, on-premise ERP system.

For details about how Order Management integrates these channels with other systems, see Integrating Order Management with External Systems: Overview.
Monitor Order Progress and Manage Exceptions

Order Management provides a single work area that you can use to view the progress of a sales order, order status, and exceptions that might cause problems in fulfilling a sales order. This work area displays a summary of statuses and exceptions according to customer, item, or supplier, and it allows you to drill into the data to view more details. If a sales order is at risk of not meeting the fulfillment dates, then a jeopardy feature makes them more prominent so that you can identify issues in time to take corrective action. Analytic information provides the data that you require to make informed choices. You can use this work area to monitor the progress of each sales order:

- View order statuses and exception conditions.
- Filter sales orders according to customer, item, fulfillment location, supplier, status, or age.
- Use a Gantt chart that includes timelines to monitor processes.
- Fix exceptions one or more lines at a time.
- Use embedded intelligence to resolve exceptions.

Simplify Processing for Change Orders

Change order management allows you to define compensation patterns and roll back steps so that you can consistently control change order processing. You can use change order logic to make sure Order Management processes and revises sales orders consistently across all sales orders. For example, if Order Management receives a quantity change from a source order, and if it has not shipped the item, then it uses change logic that allows the change, and then rolls back the fulfillment process so that it can reschedule and send a new ship request to the external fulfillment system. You can use change order management to do the following work:

- Specify the attributes that affect a change order and that automatically recognize a change order.
- Define rules that specify how to automatically handle changes so that a user is not required to intervene in every change that occurs.
- Coordinate change order tasks with external fulfillment systems.
- Specify the tasks in the process that the change order affects.
- Adjust the fulfillment processing steps that the changes in a sales order affects.
- Cancel orders, add lines to orders, and change order quantity.
- Revise existing sales order in Order Management.

Set Up Order Processing

You can define the orchestration process that Order Management uses to fulfill a sales order. You can define a new orchestration process, or you can modify a fulfillment process that comes predefined with Order Management that already meets most of your business requirements. The process definitions allow you to define the following types of order processing that your organization requires:

- Use common, conditional, parallel, or interrelated flows for order lines.
- Define change order rules that automatically modify order fulfillment.
- Define rules that calculate the order completion date according to the requested date.
• Define common statuses to use across your business process.
• Create draft orders in Order Management that you can later submit for processing.

You can define an orchestration process that plans fulfillment processes, calculates lead times, modifies order compensation steps, and so on. To help you visualize an orchestration process, Order Management displays it as a flowchart that includes steps, lines, decision points, and so on.

**Enrich Source Orders For Fulfillment**

You can transform a source order and add information to it that improves fulfillment. For example, you can add or modify attributes or add more items so that Order Management can efficiently fulfill the sales order. You can use a rule editor to define the rules that specify how to transform a source order, how to modify order attributes, or how to define order lines. You can do the following work to determine how Order Management transforms a source order into a sales order:

• Define item transformation rules according to item structure.
• Use fulfillment item relationships to define item transformation rules.
• Define business rules that transform an item, such as product-to-product, product-to-attribute, transactional attribute-to-product, attribute-to-attribute, context-to-product, or context-to-attribute.

**Manage Problems and Recover from Errors**

You can do the following work to manage problems and recover from errors:

• Use a jeopardy feature to help you monitor each sales order against a fulfillment date, predict whether or not the sales order is on schedule to meet the fulfillment date and, if not, take corrective action before the sales order misses the fulfillment date.
• Use a central work area to recover sales orders that are in an error state.
• Locate the problem, identify the root cause, and then adjust order fulfillment parameters to fix the problem.
• Modify dates and attributes that affect process planning.
• Query and examine a subset of sales orders according to a criteria that you specify.
• Take action on a single sales order, or on multiple sales orders at the same time.
• Schedule a background process that automatically recovers errors according to parameters and filters that you define.

**Order Management Architecture**

**Order-to-Cash: How It Works**

You can use Order Management Cloud to order and fulfill an item, such as a desktop computer, configured item, or a service, such as monthly maintenance for a network of desktop computers. A user can create a new sales order in the Order Management work area. You can also integrate Order Management with one or more upstream order capture systems, and with one or more downstream fulfillment systems.
You can use various fulfillment techniques, such as drop ship, back-to-back shipping, and internal transfers. For details about some of these fulfillment techniques, see Oracle SCM Cloud, Implementing Manufacturing and Supply Chain Materials Management.

The following diagram illustrates the architecture that Order Management uses when a user creates a new sales order in the Order Management work area. To view a variation of this flow, see Drop Ship in Order Management Cloud: How it Works.

**Explanation of Callouts**
The Order Management architecture uses the following steps when it processes a new sales order. It also uses these steps when a user revises a sales order:

1. A user clicks Create Order in the Order Management work area:
   - The user searches for a customer in the Customer field of the order header. Order Management gets information about this customer from the Oracle Customer Model, and then displays it in the order header.
   - The user searches for an item in the Order Lines area of the order. Order Management gets information about this item from the Oracle Product Model, and then displays it on the order line when the user clicks Add.
   - Oracle Pricing Cloud defines and manages prices for the items that the user enters.
   - If the user adds a configured item in an order line, then Oracle Fusion Configurator Runtime allows the user to modify the configure options that configure the item. For details, see the guide titled Oracle SCM Cloud, Configurator Modeling Guide.
   - The user clicks Submit, Order Management validates the order, and then an assignment rule identifies the orchestration process that Order Management must run to fulfill the order. You can configure this rule. For details, see Assigning Orchestration Processes: Explained.

   For information about how to get orders into Order Management from an external source system, see Implementing Order-to-Cash: Points to Consider. For information about how to import orders from a spreadsheet, see Using Files to Import Orders into Order Management: Procedure.

2. Order Orchestration uses an orchestration process that identifies and assigns the set of steps that it will use to fulfill the order. Examples of these steps include source, schedule, fulfill, pick, pack, and ship. The steps that it assigns vary so that they optimize order fulfillment according to the specific needs of the order. It determines process logic, such as determining the statuses to use for the order, how to do forward planning and backward planning, how to compensate for modifications to the order, and so on.

   Order Orchestration also does the following work:
   - References data in the Oracle Product Model to determine the item data to use for the order. You can specify this item model. For details, see Getting Data from Different Product Models When Using Oracle Business Rules: Procedure.
   - References Oracle Business Rules that specify how to orchestrate each order. You can define a variety of rules. For example, you can define a rule that uses a different lead time depending on whether or not the inventory organization resides in Denver or in San Francisco, or a rule that makes sure Order Management does not attempt to ship a nonshippable item, such as a video that your customers can only stream from the cloud. For details, see Defining Lead-Times for Orchestration Process Steps: Procedure.
   - References data that resides in the Order Management and Planning Data Repository to cross-reference and validate data. To collect this data, you define data collection parameters, enable data for collections, and so on. For details, see Managing Planning Source Systems for Data Collections: Explained.
   - Sequences the fulfillment steps that process the order.
   - Calls the task layer services. You can determine how an orchestration process calls a task layer service. For details, see Defining Orchestration Processes: Examples.

   For details about how to define an orchestration process that meets your business needs, see Defining Orchestration Processes: Explained.

3. Task layer services run the orchestration process steps and manage fulfillment tasks. The Schedule task layer service communicates with Global Order Promising so that it can cross-reference items before it sends the message to downstream fulfillment systems through the External Interface Layer. Task layer services interpret replies and updates from these systems. For example, to send a shipment request to the shipping system, an orchestration process can call the Shipment Task Layer service to create a shipment request. For details, see Task Layer: Explained.
4. An orchestration process interacts with Oracle Global Order Promising Cloud to determine availability and to promise the order. Global Order Promising Cloud does the following work:

- Uses supply and demand data from the Order Management and Planning Data Repository to determine availability in the supply chain, such as inventory levels in warehouses.
- Schedules the best fulfillment options for each fulfillment line. A user can also manually use the Order Management work area to determine and choose these options.
- Communicates with Oracle Shipping to ship the order. For details, see Shipping: How it Works.
- Updates the fulfillment status.
- Sends order information to order billing.
- If the order must reserve the item, then Order Management interacts with Oracle Inventory Management.

Each orchestration process that comes predefined with Order Management includes a scheduling step that uses Global Order Promising. However, you can define an orchestration process that does not include a scheduling step and that does not use Global Order Promising. For example, if your organization sells downloadable software or publications that do not require a warehouse or shipping.

5. The External Interface Layer communicates order details between Order Management and each fulfillment system. It primarily routes the fulfillment request and converts the data so that each fulfillment system can correctly use this data.

6. Oracle fulfillment systems do the following work:

- Oracle Materials Management Cloud manages logistics and inventory for the order, including schedule, reserve, receive, and ship each item.
- Accounts Receivables processes the billing information, including one-time charges and recurring charges, and then sends this information to Oracle Financials Cloud.
- Oracle Financial Cloud performs the financial transactions for the order. It creates an invoice, manages accounts receivable, processes payments, and manages revenue.

7. The External Interface Layer sends a request to external fulfillment systems to fulfill the order. These external systems process the request and send completion updates to the External Interface Layer so the orchestration process can move to the next step. The user can use the Order Management work area to specify fulfillment options that affect these fulfillment systems.

8. Order Management Cloud and Global Order Promising Cloud collect data, and then sends it to the Order Management and Planning repository. For example, Global Order Promising collects the following types of data:

- **Static.** For example, structures, routing rules, suppliers, transit times, and so on.
- **Dynamic.** For example, on-hand inventory, purchase orders, and data from other Cloud solutions.

9. An end user uses the Order Management work area to manage orders during the fulfillment process. For example, to monitor order line status, modify fulfillment options, revise orders, and so on.

**Order-to-Cash with Order Capture Systems: How it Works**

You can configure an order capture system that resides outside of Order Management Cloud, such as Oracle Configure, Price and Quote Cloud, to send a source order to Order Management that includes a nonconfigured item, pick-to-order configured item, or kit. This topic describes the order-to-cash flow that Order Management Cloud uses when it receives a source order from a source system that resides upstream of Order Management.
The following diagram illustrates how Order Management can transform a source order to a sales order, reply with a confirmation, and then orchestrate fulfillment tasks for this sales order, such as schedule, reserve, ship, and bill. It also updates the fulfillment line status while it does this work, and then sends it periodically to the order capture system.

Explanation of Callouts

Order Management uses the same flow with an external order capture system that it uses when the user creates a sales order directly in Order Management, with the following important differences:

1. An order capture system that resides outside of Order Management captures a source order. For example, a user might use a legacy application that your company owns to enter a source order. This capture system then
uses a connector to send the source order to Order Management. You must add this connector. For details, see Connecting Order Management to Source Systems: Procedure.

2. Data Collection collects data from your order capture system. Order Management uses this data to create cross-references for various items, such as warehouses, units of measure, carriers, currencies, shipping methods, payment terms, accounting rules, invoicing rules, service levels, tax classification codes, and so on. For details about how Data Collection uses this data, see Data Collections, Order Orchestration, and Order Promising: How They Fit Together.

The data that you collect into the Order Management and Planning repository references data that the Oracle Customer Model and the Oracle Product Model use. For details about how to set up data collection, see the guide titled Oracle SCM Cloud, Implementing Supply Chain Planning.

3. Order Orchestration uses connector services and references rules in the Oracle Business Rules repository to transform and orchestrate each order:
   - Performs pretransformation.
   - Uses transformation rules to transform the source order into a sales order.
   - Performs posttransformation cleanup.

For details, see Transforming Source Orders to Sales Orders: How It Works.

4. Global Order Promising uses data in the Order Management and Planning repository that you populated from your order capture system to do promising.

5. Order Management uses the order-to-cash flow to continue, with the difference occurring primarily during order entry and order transformation. No transformation occurs when the user enters a sales order directly in the Order Management work area. For details, see Order-to-Cash: How it Works.

Order Management communicates order status and invoice status to your order capture system throughout processing.

A single source order might contain order lines that include one-time charges, and other order lines that include recurring charges. If Order Management receives a source order that includes a recurring charge, then Oracle Receivables creates a recurring billing invoice for these lines.

Order import typically uses the same flow that Order Management uses when it gets source orders from an external source system, except you import source orders from a spreadsheet. For details, see Using Files to Import Orders into Order Management: Procedure.

**Related Topics**
- Using Files to Import Orders into Order Management: Procedure

**Architectural Layers in Order Management: How They Work**

Order Management Cloud uses several architectural layers to fulfill each source order that it receives from a source system.
Order Management uses the following flow.

Order Management performs the following work. The flow for a sales order that the Order Entry Specialist enters directly in Order Management is similar, except it starts primarily in the Orchestration Layer at step 3:

1. An external source system sends a source order to Order Management.
2. The transformation layer transforms the source order to a sales order, and then assigns an orchestration process to this sales order.
3. The orchestration layer runs the fulfillment step, and then sends it to the task layer. Each fulfillment request starts here.
4. The task layer sends the request to the external interface layer. This layer includes services that can send a request to perform a fulfillment task to a fulfillment system. Order Management provides services that can perform some of the more typical fulfillment system work, such as shipping or invoicing. You can use the Template task layer services to add functionality. For details, see Task Layer: Explained.
5. The external interface layer does the following work:
   - Cross-references the data that the request contains.
   - Uses routing rules to identify the fulfillment system to use to fulfill each fulfillment line. For details about these rules and how to create them, see Routing Requests to Fulfillment Systems: Explained.
   - A connector sends the request to the fulfillment system.
6. The fulfillment system accepts the request, and then sends a reply.
7. The external interface layer converts the reply from the fulfillment system.
8. The task layer processes the reply from the fulfillment system.
9. The orchestration layer runs the next fulfillment step.
External Interface Layer

The external interface layer manages communication between Order Management and the fulfillment system. Note the following points:

- The fulfillment system does not communicate directly with the orchestration process. Instead, the external interface layer provides an intermediary that Order Management uses to route requests.
- Order Management uses a web service to route each request from the fulfillment task layer to the fulfillment system, and then from the fulfillment system to the fulfillment task layer.
- The connector modifies the structure and content of the outbound message so that it matches the inbound interface that the fulfillment system uses.
- You can set up each web service and routing rule, and then register the connector web service. This work integrates the fulfillment system with the external interface layer.

The external interface layer provides the following benefits:

- Keeps the external system separate from the orchestration process to minimize the modifications that you must make when you add a new source system or fulfillment system. This way, it is not necessary to modify the orchestration process every time you integrate a source system or fulfillment system.
- Provides a flexible integration that you can use in a Service Oriented Architecture (SOA) with an external system or application.
- Provides a complete, open, and integrated solution that lowers cost of ownership.

Related Topics

- Task Layer: Explained
- Routing Requests to Fulfillment Systems: Explained
- Connecting Order Management to Fulfillment Systems: Procedure
Implementing Order Management Cloud

Implementing Order-to-Cash

Implementing Order-to-Cash: Points to Consider

It is recommended that you identify the features that you must implement to support the order-to-cash flow in your business environment, and then estimate the effort required to set them up.

Consider Some of the Features You Must Set Up

Perform the preparation work described in Implementing Order-to-Cash: Roadmap. This work will provide a solid understanding of the features that you need to implement, and the work involved in setting them up.

Estimate Effort Required to Set Up Features

Use the following details to create a rough estimate of the amount of effort that your implementation will require. For example, if your implementation will require the configurator and order import, then include the setup effort for them in your project plans, and consider whether or not you must administer the job roles that support them.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Comes Predefined</th>
<th>Setup Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dashboard</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Create Order</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Revise Order</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Return Order</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Search and View Orders</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Configurator</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Order Import</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Order Transformation</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Set Up Order Status Behavior</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Set Up Constraint Behavior</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Set Up Change Order Behavior</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Set Up Notifications</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Requirement</td>
<td>Comes Predefined</td>
<td>Setup Required</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Set Up Jeopardy</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Set Up Fields</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Set Up Orchestration Process Behavior</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Receive and Process Orders from Trading Partners</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Send Advanced Shipment Notice</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Collaboration Messaging Framework</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Approve Sales Orders</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Screen Orders for Trade Compliance</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Integrate Order Management with Transportation Management</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Manage Credit Check</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Manage Credit Cards</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Manage Internal Sales Orders</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Manage Attachments</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Determine to Modify the Predefined Integration or Set Up a New One

You can modify the predefined integration or set up a new one.

Modifying the Predefined Implementation

Order Management comes predefined with most of the setup tasks already completed for you. You can modify this predefined implementation.

To get started, do the following work:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Notice the functional areas that display above the Orders functional area, such as Initial Users, Enterprise Profile, and so on. These functional areas are common areas. For details about how to set them up, see the guide titled Oracle SCM Cloud, Implementing Common Features for Oracle SCM Cloud.
4. In the Functional Areas list, click **Orders**.
5. In the Orders area, click **Required Tasks**, and then click **All Tasks**.
6. In the Orders list, drill into and complete each task as necessary, depending on your business requirements.
If you find that modifying the predefined implementation does not meet your business requirements, then create a new implementation.

**Setting Up a New Implementation**

It is recommended that you modify the predefined implementation. However, if the predefined implementation does not meet your business requirements for some reason, then you can set up a new implementation.

To get started, do the following work:

1. In the **Navigator**, click **Setup and Maintenance**.
2. On the Setup page, click **Tasks**, and then click **Edit Implementation Projects**.
3. On the Implementation Projects page, click **Actions**, and then click **Create**.
4. On the Create Implementation Project page, modify the field values, as necessary, and then click **Next**.
5. On the Select Offerings to Implement page, in the Name list, expand **Order Management**.
6. Add a check mark in the Include column for the Order Management row and for the Pricing row, and then click **Save and Open Project**.

You can create one or more implementation projects for the offerings and options that you must implement. Each Oracle application creates the task list that you must complete for each project that you create. An Application Implementation Manager can set up these task lists, and can assign and track each task that these lists contain.

7. On the Implementation Project page, in the Task list, expand **Order Management**, and then perform the tasks that your business flow requires.

**Consider Which Scheduled Processes You Must Run**

You might need to set up or run one or more of the following scheduled processes when you set up Order Management.

<table>
<thead>
<tr>
<th>Scheduled Process</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform Data Load from Staging Tables</td>
<td>You collect organization parameters from your source system and load them into staging tables. You use Perform Data Load from Staging Tables to process data in these staging tables. For details, see the Collect Data section in Implementing Order-to-Cash: Quick Start.</td>
</tr>
<tr>
<td>Refresh and Start the Order Promising Server</td>
<td>Updates the Global Order Promising data. For details, see the Configure Global Order Promising section in Implementing Order-to-Cash: Quick Start.</td>
</tr>
<tr>
<td>Load Interface File for Import Import Sales Order</td>
<td>You use these scheduled processes when you import source orders. For details, see Using Files to Import Orders into Order Management: Procedure</td>
</tr>
<tr>
<td>Delete Orders From Interface Tables</td>
<td></td>
</tr>
<tr>
<td>Transfer Invoice Details to Supply Chain Financial Flow Orchestration</td>
<td>Send details about validated invoices, canceled invoices, and corrected invoices to Financial Orchestration.</td>
</tr>
<tr>
<td>Transfer Ownership Change Events to Receiving</td>
<td>Send details about the AP Invoice Match from Financial Orchestration to the receiving process. For details, see Specifying Events That Indicate an Ownership Change: Explained.</td>
</tr>
<tr>
<td>Update or Close Sales Orders</td>
<td>Order Management might display sales orders and fulfillment lines as open even if it closed all the fulfillment lines that these orders and lines reference. You can use Update or Close Sales Orders to fix this problem. For details, see Closing Fulfillment Lines That Remain Open: Explained.</td>
</tr>
<tr>
<td>Generate Constraint Packages</td>
<td>Create a constraint package. For details, see Managing Processing Constraints: Explained.</td>
</tr>
</tbody>
</table>
### Scheduled Process

<table>
<thead>
<tr>
<th>Scheduled Process</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publish Extensible Flexfield Attributes</td>
<td>Publish and deploy an extensible flexfield. For details, see Deploying Extensible Flexfields in Order Management: Procedure</td>
</tr>
<tr>
<td>Plan Orchestration Processes</td>
<td>Update an orchestration process plan at regular intervals according to the frequency that your deployment requires. For details, see Defining Orchestration Processes: Explained.</td>
</tr>
<tr>
<td>Release Pause Tasks</td>
<td>Release a pause task so processing can continue. For details, see Pausing Orchestration Processes Until an Event Occurs: Procedure and Automatically Resuming Paused Orchestration Processes: Explained.</td>
</tr>
<tr>
<td>Generate Bucket Sets</td>
<td>Automatically keep bucket sets up to date with reference data and transactional data. For details, search for Document ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at <a href="https://supportoracle.com">https://supportoracle.com</a> and then download the attachment that describes how to use business rules.</td>
</tr>
<tr>
<td>Import AutoInvoice</td>
<td>Import and validate transaction data from Order Management or non-Oracle financial systems to create invoices, debit memos, credit memos, and account credits in Oracle Fusion Receivables. For details, see AutoInvoice Import: How Data Is Processed at <a href="https://fusionhelporacle.com:443/helpPortal/faces/AtkHelpPortalMainTopicIdA7E6EC0B3F86C3D8E040D30A68811AA8">https://fusionhelporacle.com:443/helpPortal/faces/AtkHelpPortalMainTopicIdA7E6EC0B3F86C3D8E040D30A68811AA8</a>.</td>
</tr>
</tbody>
</table>

## Implementing Order-to-Cash: Roadmap

This road map provides a high-level procedure that you can use to implement order-to-cash. You perform the common setup tasks, set up integration, set up parameters, and so on.

Note the following:

- It is recommended that you perform the tasks that this topic references in the same sequence that this topic displays them.
- Steps after step 3 in this topic primarily reference tasks that reside in the Orders functional area. Complete each of these tasks, as necessary, depending on your business requirements.
- To perform a task, click each task in the Orders area. If the Orders area does not display a task, then search for it in the Search Tasks field.
- The work you must perform depends on your business requirements. For example, a typical implementation requires that you perform most or all of the tasks that integrate Order Management. However, whether or not you perform other tasks depend on your business requirements, such as a task that controls order status, constrains changes, or that modifies an orchestration process.
- The Topic column references topics that reside in the guide titled Oracle SCM Cloud, Implementing Order Management. It is recommended that you use these topics when you perform the task that they describe.
- The Prepare column describes work that you can do to prepare for the implementation. It is recommended that you perform this work before you administer order-to-cash. Doing so will help to avoid interruption and downtime during administration, and will help to make sure it goes smoothly. You can also use the Prepare column to check off preparation work as you complete it.
- To help monitor your progress, add a check mark to the Done column when you finish each functional area or task.
- To implement drop ship, see the Implementing Drop Ship section in the Implementing Order Management guide.
To implement order-to-cash, do the following work:

1. Read the Implementing Order-to-Cash: Points to Consider topic. It contains details that help to make the implementation go smoothly.

2. In the Navigator, click **Setup and Maintenance**.

3. On the Setup page, click **Order Management**.
   This topic assumes that you will modify the predefined implementation that is available through Setup. For details about setting up a new implementation, see Implementing Order-to-Cash: Points to Consider.

4. In the Functional Areas list, complete the following functional areas that display above the Orders functional area. These areas reference common tasks. A typical implementation requires that you complete most or all common tasks. For details about how to set them up, see the guide titled Oracle SCM Cloud, Implementing Common Features for Oracle SCM Cloud.

<table>
<thead>
<tr>
<th>Functional Area</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Users</td>
<td></td>
</tr>
<tr>
<td>Enterprise Profile</td>
<td></td>
</tr>
<tr>
<td>Organization Structures</td>
<td></td>
</tr>
<tr>
<td>Users and Security</td>
<td></td>
</tr>
<tr>
<td>Items</td>
<td></td>
</tr>
<tr>
<td>Catalogs</td>
<td></td>
</tr>
<tr>
<td>Customers</td>
<td></td>
</tr>
</tbody>
</table>

5. In the Functional Areas list, click **Orders**.

6. In the Orders area, click **Required Tasks**, and then click **All Tasks**.

7. Integrate Order Management. Complete the following tasks. For details, see the Integrating Order Management chapter in the Implementing Order Management guide.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
<th>Prepare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Web Service Details</td>
<td>Creating User Credential Keys: Procedure</td>
<td>Gained access to the administrator privilege and administrator role so that we can use Oracle Wallet Manager.</td>
</tr>
<tr>
<td>Manage Trading Community Source Systems</td>
<td>Connecting Order Management to Source Systems: Procedure</td>
<td>Located the time zone where the server that the source system uses is located.</td>
</tr>
<tr>
<td>Manage Upstream and Fulfillment Source Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage External Interface Web Service Details</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>Topic</td>
<td>Prepare</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>Manage External Interface Web Service Details</td>
<td>Connecting Order Management to Fulfillment Systems: Procedure</td>
<td>- ___Identified an integrated development environment we can use to create a transformation style sheet. &lt;br&gt; - ___Acquired the user credentials that the service provider requires when calling their web service. &lt;br&gt; - ___Acquired contact details for the IT administrator who works for the service provider. &lt;br&gt; - ___Identified the URL that locates the web service that resides on the fulfillment system. &lt;br&gt; - ___Acquired the security certificate from the service provider.</td>
</tr>
<tr>
<td>Manage External Interface Routing Rules</td>
<td>Routing Requests to Fulfillment Systems: Procedure</td>
<td>___Created a list that includes the names of the items that we must route to fulfillment systems, the unique identifier for each item, and the name of each fulfillment system where Order Management must route the request.</td>
</tr>
</tbody>
</table>

8. Control application behavior. Complete the following tasks. For details, see the Controlling Application Behavior chapter in the Implementing Order Management guide.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
<th>Prepare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Order Management Parameters</td>
<td>Managing Order Management Parameters: Procedure</td>
<td>___Determined whether or not our business requirements will allow or disallow the Configurator to choose items in a configuration and to modify a configuration after the user adds a configured item. &lt;br&gt; ___Determined the date that Order Management must use when it determines the configure options that it displays for a configured item. &lt;br&gt; ___Determined how Order Management will display the Ship-to Address and the Bill-to Location for each customer during order entry. &lt;br&gt; ___Determined whether or not the Configurator stops processing when it encounters an error. &lt;br&gt; ___Identified the item validation organization that Order Management must use to validate and display items. &lt;br&gt; ___Identified an Order Management user that the buyer can contact to resolve a problem that occurs with a drop ship.</td>
</tr>
</tbody>
</table>
## Task | Topic | Prepare
---|---|---
| Completed whether or not the Configurator validates each order that includes a configured item during order import. | | 

### Manage Order Profiles

**Managing Order Profiles: Procedure**

- Identified the value to use when converting a currency.
- Identified the currency to display in the Order Management work area.
- Identified the customer to use when filtering status data on the Overview page.
- Determined whether or not to use the source order number as the order number during transformation.
- Determined the number of seconds to wait after an action finishes.

### Administer pricing. For details, see the guide titled Oracle SCM Cloud, Administering Pricing.

### Import source orders. If your implementation includes a source system, then you must import source orders.

Complete the following tasks. For details, see the Importing Orders section in the Implementing Order Management guide.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
<th>Prepare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not applicable</td>
<td>Using Files to Import Orders into Order Management Cloud: Procedure</td>
<td>Acquired access to our source order data.</td>
</tr>
<tr>
<td>Create Source System</td>
<td>Using Web Services to Import Source Orders: Explained</td>
<td>Determined whether or not our implementation will use the product model or the product hub.</td>
</tr>
<tr>
<td>Manage Upstream and Fulfillment Source Systems</td>
<td></td>
<td>Identified the attributes that we will use in the Request Payload.</td>
</tr>
<tr>
<td>Manage Standard Lookups</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage Item Relationships</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Transform source orders. If your implementation includes a source system, then you must transform source orders.

Complete the following tasks. For details, see the Transforming Source Orders to Sales Orders section in the Implementing Order Management guide.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
<th>Prepare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Product Transformation Rules</td>
<td>Setting Up Transformation: Procedure</td>
<td>Acquired login user and password to the Oracle Product Model.</td>
</tr>
</tbody>
</table>
12. Set up statuses. For details, see the Controlling Order Statuses section in the Implementing Order Management guide.

13. Set up processing constraints. For details, see the Constraining Orders section in the Implementing Order Management guide.
14. Control change orders. For details, see the Managing Change Orders section in the Implementing Order Management guide.

15. Notify systems when orders change. For details, see the Notifying Systems When Orders Change section in the Implementing Order Management guide.

16. Use jeopardy to manage delays. For details, see the Using Jeopardy to Manage Delays section in the Implementing Order Management guide.

17. Manage task types. For details, see the Setting Up Task Types section in the Implementing Order Management guide.
<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
<th>Prepare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Task Types</td>
<td>Setting Up Task Types: Procedure</td>
<td>___ Described business requirements regarding setting up task types that our implementation might require.</td>
</tr>
</tbody>
</table>

**18.** Set up fields. For details, see the Setting Up Fields section in the Implementing Order Management guide.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
<th>Prepare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Order Extensible Flexfields</td>
<td>Deploying Extensible Flexfields in Order Management: Procedure</td>
<td>___ Described business requirements regarding what details we must display in fields, including the data and location in the user interface.</td>
</tr>
</tbody>
</table>

**19.** Set up orchestration processes. For details, see the Setting Up Orchestration Processes chapter in the Implementing Order Management guide.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
<th>Prepare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Orchestration Process Definitions</td>
<td>Defining Orchestration Processes: Points to Consider</td>
<td>___ Confirmed that the predefined orchestration processes do not meet our business requirements.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>___ Identified configuration details for each orchestration process that we must create, such as task type to use, change logic, process planning, jeopardy, statuses, and so on.</td>
</tr>
<tr>
<td>Manage Orchestration Process Definitions</td>
<td>Defining Orchestration Processes: Procedure</td>
<td>___ Described the IF/THEN rules that our business requires for each orchestration process.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>___ Described orchestration planning and replanning that our business requires for each orchestration process.</td>
</tr>
<tr>
<td>Manage Orchestration Process Definitions</td>
<td>Deploying Orchestration Processes: Procedure</td>
<td>Not applicable.</td>
</tr>
<tr>
<td>Manage Orchestration Process Definitions</td>
<td>Defining Lead-Times for Orchestration Process Steps: Procedure</td>
<td>___ Described the IF/THEN rules and conditions that determine the lead-time to use for each orchestration process step.</td>
</tr>
<tr>
<td>Manage Orchestration Process Definitions</td>
<td>Selecting the Fulfillment Lines That Run for Orchestration Process Steps: Procedure</td>
<td>___ Described the IF/THEN rules that select the fulfillment line to run for each orchestration process step.</td>
</tr>
<tr>
<td>Task</td>
<td>Topic</td>
<td>Prepare</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>--------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Manage Orchestration Process Definitions</td>
<td>Adding Branches to Orchestration Processes: Procedure</td>
<td>___ Specified subprocesses and parallel processing that our orchestration process requires.</td>
</tr>
<tr>
<td>Manage Orchestration Process Definitions</td>
<td>Pausing Orchestration Processes: Explained</td>
<td>___ Specified the IF/Then rules that Order Management must use when it pauses each of our orchestration processes and each orchestration process step.</td>
</tr>
<tr>
<td>Manage Orchestration Process Assignment Rules</td>
<td>Assigning Orchestration Processes: Procedure</td>
<td>___ Specified the IF/Then rules that Order Management must use when it assigns each of our orchestration processes and each orchestration process step.</td>
</tr>
<tr>
<td>Not applicable</td>
<td>Automatically Resuming Paused Orchestration Processes: Explained</td>
<td>___ Identified the pause tasks that we will release to resume a paused orchestration process.</td>
</tr>
</tbody>
</table>

**Related Topics**
- Order Management Cloud: Overview

**Implementing Order-to-Cash: Quick Start**

This topic describes the minimum steps you must perform to implement Order Management Cloud if you do not use all the tasks that are available in the Order Management offering. You can use it to set up a test instance of Order Management.

⚠️ **Caution:** This topic does not include all the steps and security tasks that are required to fully implement Order Management. It includes only the minimum implementation steps you must perform so your implementation can receive and fulfill test orders.

For important background details, see the following documentation:

- **How to use the predefined implementation.** See Implementing Order-to-Cash: Points to Consider.

**Summary of the Work**

To get started with implementing Order Management, do the following work:

1. Prepare to start your implementation.
2. Configure common components.
3. Set up item organizations and product models, and import items.
4. Connect source systems and set up customers.
5. Collect data.
6. Configure Global Order Promising.
7. Set up Order Management and Pricing, and test your work.

This topic references several guides, such as the Oracle Fusion Product Information Management Implementation guide. To access each guide, navigate to the Technology Library for Oracle Cloud Applications at http://www.oracle.com/technetwork/documentation/default-1577155.html.

Prepare to Start Your Implementation

Prepare to start your implementation:

1. Consider the data you will use in your test environment.

   For example:
   - Items you will add to a sales order
   - Customers who will order the items
   - Item Organization you will use as the source to fulfill each item
   - Units of measure (UOM) you will use
   - Currency you will use in each sales order
   - Users who will create sales orders

   As part of planning your test, it is recommended that you create a list for each of these data, such as a list of the items, a list of the customers, and so on. You can then use these details later during set up.

2. Get the URLs, User IDs, and passwords you need to access Order Management Cloud and other applications, such as Oracle Identity Manager, Oracle Authorization Policy Manager, Security Console, and Oracle Cloud Applications.

   You can get these details from the Oracle provisioning team.

3. Get the location of the Oracle home directory in the Oracle Identity Management environment.

   This location uses the following format:

   `IDM_ORACLE_HOME`

   For example, the URL for Oracle Cloud Applications is:

   `http://host/homePage/faces/AtkHomePageWelcome`

   where:

   `host` is the name of the host location, such as abc.oracleoutsourcing.com.

   For example:

   `http://abc.oracleoutsourcing.com/homePage/faces/AtkHomePageWelcome`

   If you do not know the URLs, contact the person who installed the systems at your company. The summary page displays these URLs when the provisioning process that Oracle Cloud Applications uses finishes. The default file name of the file that stores these URLs is Provisioning Summary. The person who installs the software can choose a different file name. This file resides in the same folder where the provisioning plan resides.

   Your URLs will resemble the following formats.
Oracle SCM Cloud
Implementing Order Management
Chapter 2
Implementing Order Management Cloud

Authorization Policy Manager | Oracle Identity Manager

https://host/apm | https://host/oim

4. Get the user name and password for each of the following users. Contact the person who installed the systems to get the user names and passwords they used or specified when they installed and provisioned the application.

<table>
<thead>
<tr>
<th>User</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super user for Oracle Cloud Applications</td>
<td>The default user name is FAADMIN.</td>
</tr>
<tr>
<td>System administrator for the Security Console</td>
<td>Not applicable.</td>
</tr>
<tr>
<td>System administrator for Oracle Identity Manager</td>
<td>The default system administrator is XELSYSADM.</td>
</tr>
</tbody>
</table>

5. Determine the following for your test orders:
   - Identify the items that the test orders will contain and the customers who will order them.
   - Identify the item organizations that you will associate with these items.
   - Identify the units of measure (UOMs) and currencies that these test orders will use.

Security Tasks You Might Need to Perform

You might need to perform one or more of the following security tasks.

<table>
<thead>
<tr>
<th>Authorization Policy Manager</th>
<th>Oracle Identity Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform one or more of the following tasks:</td>
<td>Perform one or more of the following tasks:</td>
</tr>
<tr>
<td>• Manage Duties</td>
<td>• Import Users</td>
</tr>
<tr>
<td>• Manage Data Security Policies</td>
<td>• Create Implementation Users</td>
</tr>
<tr>
<td>• Manage Role Templates</td>
<td>• Provision Roles to Implementation Users</td>
</tr>
<tr>
<td>• Manage job to duty role hierarchies</td>
<td>• Manage Job Roles</td>
</tr>
<tr>
<td>•</td>
<td>• Manage User Principal</td>
</tr>
</tbody>
</table>

When you acquire the super user, you can use the following tasks from the Setup and Maintenance work area:

- Manage Role Templates
- Manage Job Roles

Configure Common Components

Perform the implementation steps that are common to an Oracle application. For details about how to perform each of the following steps, search for Document ID 1387777.1 (Getting Started with Oracle Fusion Applications, Common Implementation) on My Oracle Support at https://support.oracle.com, and then see the table that the step mentions. For
example, for step 1, see Table 1 in Getting Started with Oracle Fusion Applications. For step 2, see Table 2 in Getting Started with Oracle Fusion Applications:

Configure common components:

1. Prepare the super user for user management and configuration.
2. Prepare the IT Security Manager Role.
3. Create the setup task list for the Order Management offering.

This offering includes the tasks that you perform.

A large task list displays when you create the list for the Order Management offering. To proceed with the implementation, make sure you perform step 5, define the enterprise structures, perform step 3, Set up Item Organizations, through step 10, Set up Pricing, and then create or import orders.

4. Define implementation users.

Perform the work described in Document ID 1387777.1. Also, assign the following roles to users.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing Administrator, All Business Units, which is QP_PRICING_ADMINISTRATOR_ALL_BUSINESS_UNITS_DATA</td>
<td>This role allows you to perform pricing administration tasks. Add it when you perform the Set up Pricing task.</td>
</tr>
<tr>
<td>Product Manager, which is EGP_PRODUCT_MANAGER_JOB</td>
<td>This role allows you to set up organizations and items so you can add items to your sales orders.</td>
</tr>
</tbody>
</table>

You might need to use Authorization Policy Manager to modify the data security for these roles. For details, see the following topics:

- Data Security Privileges for Creating Items: Explained
- Adding Data Security Policies to Duty Roles in Authorization Policy Manager: Procedure

5. Set up the enterprise structure.

Note the following:

- You can use the default enterprise structure for a pilot implementation.
- The Order Management work area displays sales orders for the business units that the current user can access. You must create a separate business unit for each business unit that will receive sales orders.
- A set is a collection of business units. Order Management uses sets to restrict access to holds and orchestration processes. You must specify a default set when you create a business unit. You can use the predefined Common Set for the default set.

6. Define the users who perform functional testing.

An implementation user can access a wide range of privileges. To test with users who have fewer privileges, you can define users with the following roles that are specific to Order Management.

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job</td>
<td>Order Entry Specialist</td>
</tr>
<tr>
<td>Role</td>
<td>Name</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>Job</td>
<td>Order Manager</td>
</tr>
<tr>
<td>Job</td>
<td>Order Administrator</td>
</tr>
<tr>
<td>Job</td>
<td>Supply Chain Application Administrator</td>
</tr>
<tr>
<td>Abstract</td>
<td>Error Recovery</td>
</tr>
</tbody>
</table>

To examine how these roles implement security, you can create at least one user for each of the following roles:

- Only the Order Manager role
- The Order Manager role and the Error Recovery role
- The Order Administrator role and the Supply Chain Application Administrator role

The Supply Chain Application Administrator role provides the Order Administrator role.

For details about job roles, see Assigning Job Roles in Order Management: Explained.

**Set Up Organizations and Product Models, and Import Items**

Set up item organizations, the product model, and import items:

1. Set up item organizations:
   - In the Navigator, click **Setup and Maintenance**.
   - On the Setup page, click **Order Management**.
   - Click **Organization Structures**, search for, and then use the Manage Item Organizations page to configure your item organizations so the Oracle Product Model can use them.

   Note the following points:
   - You will need at least one organization that represents a warehouse where your implementation can collect the supply data that it uses to fulfill each order.
   - You must configure each warehouse from a fulfillment system as an item organization in Oracle.
   - You must not associate an item organization with a business unit.

   For details, see the following topics:
   - Item Organization: Explained
   - Item Master Organization: Explained

2. Set up the product model:
   - In the Setup and Maintenance work area, search for, and then use the Manage Units of Measure page to configure your Units of Measure so the Oracle Fusion Product Model can use them.
   - On the Setup page, click **Items**, search for, and then use the Manage Units of Measure page to configure your item organizations so the Oracle Fusion Product Model can use them.
For details, see the Manage Units of Measure section in the Oracle Fusion, Product Information Management Implementation guide.

- Set up the Oracle Fusion Product Model.

The Oracle Fusion Product Model must contain the items that each test order references. For details, see the Define Product Information Management section in the Oracle Fusion, Product Information Management Implementation guide.

3. Define item classes, items, and catalogs according to your test requirements.

4. Optional. Use an order import template to import the items that your test orders will reference. Transform orders, as necessary.

For details, see the following references:

- The chapter that describes Importing and Transforming Source Orders in the Implementing Order Management Guide.
- Item Imports: Explained
- The Define Product Information Management section of the Define Basic Items chapter in the Oracle Fusion Product Information Management Implementation Guide.

## Connect Source Systems and Set Up Customers

Connect source systems and set up customers:

1. Optional. Set up and connect the source system.

You must set up and connect the source system in the following situations:

- You plan to use a source system. For details, see Connecting Order Management to Source Systems: Procedure.
- You plan to import customer data from a source system.
- You plan to use cross references for referential data. For details, search for Document ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then download the attachment that describes how to use cross-references.

If you will use only the Order Management work area to create sales orders, then you do not need to set up a source system.

2. Optional. Import customers:

- On the Setup page, click Customers, search for, and then open the Import Person and Organization page.
- On the Manage Data Import Batches page, click Actions, and then click Create to create an import batch.
- Use an Extract, Transform, and Load tool to load your data into the interface tables.

You can use this import process to import a batch from the interface tables into the Trading Community Model registry. The batch you import must include the customers that your orders will reference. For details, see Defining the Import Process for Customers and Consumers: Points to Consider, and Defining the Import Process for Customers and Consumers: Worked Example.

### Getting Customer Details when Not Integrating or Importing
If you will not integrate with a source system or fulfillment system, or if you will not import your customers, then you can create customers in one of the following ways:

- **Create them in Order Management.** In the Navigator, click Setup and Maintenance. On the Setup page, search for, and then open Create Customer. For details, turn on embedded help, and then click the help icon on the Create Organization Customer page. Also, see Oracle Trading Community Architecture User Guide.

- **Synchronize them on the customer model.** If the customer and customer entities that Order Management requires to fulfill the sales order do not exist, then the customer model synchronizes the customer master in the customer model with the customer details from the sales order. The important details include sold-to party, ship-to party, and bill-to account.

**Collect Data**

Order Management references units of measure, currency, and currency conversions from the Order Orchestration and Planning Data repository. You must perform the tasks in this table to collect data for these entities so your implementation can receive source orders. Global Order Promising sourcing rules reference the Organization Parameter entity. You must collect Organization Parameters before you define sourcing rules.

Collect data:

1. Manage planning source systems:
   - In the Navigator, click Setup and Maintenance.
   - On the Setup page, click Order Management.
   - Search for, and then open Manage Planning Source Systems.
   - On the Manage Planning Source Systems page, in the Source Systems list, locate your Oracle system, and then set the following values for it.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Fusion</td>
</tr>
<tr>
<td>Collections Allowed</td>
<td>Contains a check mark</td>
</tr>
<tr>
<td>Order Orchestration Type</td>
<td>Order Orchestration</td>
</tr>
<tr>
<td>Enable Data Cross-Reference</td>
<td>Contains a check mark</td>
</tr>
</tbody>
</table>

- Set the following values for each of the systems that your implementation must integrate, such as an order capture system or fulfillment system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Others</td>
</tr>
<tr>
<td>Collections Allowed</td>
<td>Contains a check mark</td>
</tr>
<tr>
<td>Enable Data Cross-Reference</td>
<td>Contains a check mark</td>
</tr>
</tbody>
</table>
2. Allow the currencies and units of measure to cross-reference data. On the Setup page, search for, and then open the Manage Planning Source Systems page.

For details, see Collecting Planning Data: Explained.

3. Collect the organization parameters from your external source system.

To identify your organization, the collected data concatenates the source system to the organization ID. Other entities, such as Sourcing Rules, also use this identification. To collect this data, you load it into staging tables, and then use a scheduled process named Perform Data Load from Staging Tables.

4. Collect units of measure, currencies, and currency conversions from your order capture system.

Note the following points:

- Order Management validates the units measures and currencies against data that it already collected from your order capture system when it receives source orders from this system.
- You must collect calendars from your fulfillment system so Order Management can use them during scheduling. A calendar specifies when a facility, such as a warehouse, is open or closed.
- To collect this data, you load it into staging tables, and then use a scheduled process named Perform Data Load from Staging Tables.
- Use the review pages to verify the data you collect.
- A pilot implementation expects your test orders to use the same values for the unit of measure, currency, and currency conversion you collect into the data repository. During a full implementation, if multiple order capture systems use different values for these entities, then you must perform more implementation steps. For example, if one order capture system uses Ea for each item, and if another order capture system uses Each for each item, then you must perform more steps to implement the data cross-referencing that these order capture systems require.
- Your implementation can collect reference data from multiple systems. However, a reference data entity is a global object, so Order Management and Global Order Promising uses the most recent data that your implementation collects. It is recommended that you identify the source system that contains the master data list, and then collect data from this source system after you collect data from all other source systems. For example, if source system B contains the master list of currencies, then collect currencies from source system A so Order Management can cross-reference currencies to system A, and then collect currencies from system B.
- If Order Promising must consider the transit time that occurs during shipping between the warehouse and a destination, then you must collect shipping methods. If you do not collect them, then Order Management uses a transit time of zero days.

For details, see Data Collections, Order Orchestration, and Order Promising: How They Fit Together.

Configure Global Order Promising

Oracle Fusion Global Order Promising uses sourcing rules, assignment sets, and ATP Rules when it determines availability. You must enable any new organizations and items you create for this implementation for collection.

For details about Global Order Promising, see the following topics:

- Order-to-Cash: How It Works
- Data Collections, Order Orchestration and Order Promising: How They Fit Together
- Refreshing the Global Order Promising Engine: Explained
Configure Global Order Promising:

1. In the Navigator, click **Order Promising**, click **Tasks**, click **Manage Sourcing Rules**, and then create a global sourcing rule that specifies the warehouse that Order Management must use when it fulfills a sales order.

   For details, see Sourcing Rules and Bills of Distribution: Explained.

2. On the Manage Sourcing Rules page, click **Tasks**, click **Manage Assignment Sets**, and then create an assignment set that assigns the sourcing rule that you created in step 1 to the Global assignment level.

   For details, see Sourcing Assignment Levels: Explained.

3. Set the administrator profile value:
   - Navigate to the Manage Administrator Profile Values page, and then, in the Application field, select **Global Order Promising**, and then click **Search**.
   - In the Search results list, set the following value for the MSP_DEFAULT_ASSIGNMENT_SET profile.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Value</td>
<td>Choose the sourcing rule that you created in step 1.</td>
</tr>
</tbody>
</table>

   If you already defined a default assignment set, and if you are writing rules for new organizations and items, then add the assignment rules for these new organizations and items to the default assignment set.

4. Create an ATP rule:
   - In the Navigator, click **Order Promising**, click **Tasks**, click **Manage ATP Rules**, and then create an ATP rule that uses the Infinite-Availability-Based promising mode.
   - Assign the ATP rule to each of the organizations that you created for your fulfillment warehouses.

   For details, see ATP Rule Promising Modes: Explained.

5. Update the Global Order Promising data:
   - In the Navigator, click **Scheduled Processes**.
   - On the Scheduled Processes page, click **Actions**, and then click **Schedule New Process**.
   - In the Schedule New Process dialog, set the following field, and then click **OK**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Refresh and Start the Order Promising Server</td>
</tr>
</tbody>
</table>

   - In the Process Details dialog, add a check mark to every parameter, and then click **Close**. Internal Transfer Order is an example of a parameter. You must add a check mark to every parameter even if you have not collected data for them.

   For details, see Refreshing the Global Order Promising Engine: Explained.

**Set up Order Management and Pricing, and Test Your Work**

You must set up connections to any order capture systems and fulfillment systems that your implementation uses, and then deploy the predefined data that specifies how Order Management fulfills each order.
Set up Order Management, and then test your work:

1. Set up order management parameters according to your testing requirements.
   For details, see Managing Order Management Parameters: Procedure.

2. Perform the minimum setup tasks that a test environment requires:
   - In the Navigator, click **Setup and Maintenance**.
   - On the Setup page, click **Order Management**.
   - Perform the following setup tasks. For a complete list of tasks you can perform, see Implementing Order-to-Cash: Roadmap.

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Order Profiles</td>
<td>Define values for the following profile options:</td>
</tr>
<tr>
<td></td>
<td>• Display Currency</td>
</tr>
<tr>
<td></td>
<td>• Currency Conversion Type</td>
</tr>
<tr>
<td>For details, see Setting Orchestration Profile Options: Procedure.</td>
<td></td>
</tr>
<tr>
<td>Manage Orchestration Process Definitions</td>
<td>Define and deploy the following predefined orchestration processes:</td>
</tr>
<tr>
<td></td>
<td>• ShipOrderGenericProcess</td>
</tr>
<tr>
<td></td>
<td>• ReturnOrderGenericProcess</td>
</tr>
<tr>
<td>For details, see Defining Orchestration Processes: Procedure.</td>
<td></td>
</tr>
<tr>
<td>Manage Processing Constraints</td>
<td>On the Manage Processing Constraints page, click <strong>Validation Rule Sets</strong>, click <strong>Generate Packages</strong>, wait a moment, and then click <strong>OK</strong> in the confirmation dialog.</td>
</tr>
<tr>
<td>For details, see Managing Processing Constraints: Procedure.</td>
<td></td>
</tr>
</tbody>
</table>

3. Optional. Define the orchestration that your test environment requires:
   - Define and deploy orchestration processes.
   - Define status codes.
   - Manage change orders.
   - Set up jeopardy and planning.
   - Release and deploy orchestration processes.
   - Create processing constraints.
   For details, see the chapters that describe controlling order processing and setting up orchestration processes in the Implementing Order Management guide.

4. Set up pricing.
   If you use Order Management to create sales orders, and do not import them or integrate to a source system that contains price details, then you must set up pricing. Order Management requires details about pricing objects, such as the pricing strategy, to determine the price of each item that you add to the sales order. You cannot submit a sales order in Order Management without the price. For details about how to set up pricing, see Managing Oracle Fusion Pricing: Roadmap in the guide titled Oracle SCM Cloud, Administering Pricing.
In addition, it might be necessary to perform the following work:

- Set up the Item Validation Organization order management parameter.
- Get access to pricing_mgr_operations, which is the job role you need to administer pricing segments, pricing strategies, price lists, and so on.

5. Use a tool of your choice, such as SOAP (Simple Object Access Protocol), to simulate order processing.

6. Test your work:

- Use a tool of your choice, such as SOAP, to simulate sending a test order. Note the order number.
- In the Order Management work area, use the source order number to search for the order.
- Confirm that Order Management received the order and is processing it.

### Setting Up Other Features

You can set up the following features, depending on your testing requirements.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Drop Shipment</strong></td>
<td>The setup you must perform to implement drop ship depends on your business requirements. For details, see the Implementing Drop Ship section in the Implementing Order Management Guide.</td>
</tr>
<tr>
<td><strong>Internal Material Transfers</strong></td>
<td>Internal Material Transfers comes predefined as available. The predefined job roles include this feature. No setup is required except that an option in Supply Chain Orchestration determines whether or not inventory routes the transfer order to Order Management. For details, see Supply Chain Orchestration: Overview.</td>
</tr>
</tbody>
</table>
| **Back-to-Back Shipping**        | Back-to-back shipping can create supply only after Order Management successfully submits the sales order to order fulfillment. Back-to-back shipping can then create supply in the following ways: 
  - Procure supply from an external supplier.
  - Produce or assemble supply at an in-house manufacturing location.
  - Transfer material from another warehouse.
  - Reserve on-hand supply. 

The fulfillment warehouse receives the supply, and then back-to-back shipping ships it to the customer. 

For details, see Back-to-Back Fulfillment: Overview. |
| **Configure to Order**           | You can use Configure to Order to efficiently fulfill each configured item. You can set the following order management parameters to control configurator behavior: 
  - Allow Changes Through Configurator Validation
  - Configuration Effective Date
  - Halt Configurator Validation on First Error
  - Use Configurator for Order Import Validation

You can also use Configure to Order to set up a kit. A kit is a configured item that includes one or more configure options, but the Order Management work area does not allow the Order Entry Specialist to modify the configure options of a kit. Note the following points: 

  - You set the subtype for a kit to KIT-SMC when you set up the configuration model. |
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| • Order Management sends the shippable lines of a kit as a shipment set to Global Order Promising.  
• The Order Management work area does not display values for On Hand and Item Availability for kits because Global Order Promising does not support the quick availability check feature for kits. |

For details about the set ups you must perform for configured items and kits, see the following resources:

• Configurator Modeling Guide
• Managing Order Management Parameters: Procedure

Integrating Order Management with Upstream Source Systems

For details, see the Integrating Order Management with Upstream Source Systems section in the Implementing Order Management Guide.

You can use the following ways to import source orders from a source system:

• **File.** For details, see Using Files to Import Orders into Order Management: Procedure.

Integrating Order Management with Downstream Fulfillment Systems

You can use a web service to allow Order Management Cloud to communicate with a fulfillment system through the external interface layer. You can use a predefined web service or create a new one. For details, see the Integrating Order Management with Downstream Fulfillment Systems section of the Implementing Order Management Guide.

**Related Topics**

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

• Security Console: Overview

**Implementing Business-to-Business Order Management**

**Business-to-Business Order Management: Overview**

Order Management Cloud provides a central order hub in a multichannel environment that you can use to improve order capture and order fulfillment in the order-to-cash process when communicating sales orders between businesses. Business-to-business messaging uses Collaboration Messaging Framework to automate message flow so that Order Management can receive and process source orders from trading partners, and then reply with an advance shipment notice after shipping successfully finishes.

Web services use Open Applications Group Integration Specification (OAGIS) messages in the payload that it uses to handle the interactions that occur between Oracle Cloud Applications and trading partners. You can use your existing Electronic Data Interchange (EDI) infrastructure with OAGIS. This configuration provides you the flexibility to receive each source order and allow a trading partner and supplier to process it through order fulfillment.
Business-to-business order management allows you to achieve the following results:

- Process purchase order
- Change purchase order
- Cancel purchase order
- Acknowledge purchase order
- Acknowledge a change in the purchase order

Business-to-business order management provides the following benefits:

- Reduce cost
- Increase processing speed and accuracy
- Improve relationships between business partners
- Simplify setup and management

**Business-to-Business Order Management: How it Works**

The Collaboration Messaging Framework uses a web service to communicate sales order details between the trading partner and Order Management Cloud. It communicates these details in XML payloads.
The following diagram illustrates how Order Management uses Collaboration Messaging Framework to communicate sales orders between businesses.

**Explanation of Callouts**

Order Management performs the following steps to communicate sales orders between businesses:

1. A trading partner uses a web service endpoint that they have set up and enabled on their server to send an XML document that contains source orders to the CollaborationMessage web service on the Collaboration Messaging Framework.
2. Collaboration Messaging Framework converts the source orders to a CSV format (comma separated value) that Order Management supports.
3. Collaboration Messaging Framework uses a web service to upload these CSV files to a Universal Content Management folder.

4. You use the order import template and a scheduled process to convert the CSV files into sales orders.

5. Collaboration Messaging Framework subscribes to a business event that the scheduled process raises when it finishes. Order Management recognizes this event, and then sends an order acknowledgment in an OAGIS message through Collaboration Messaging Framework to the trading partner.

6. Order Management processes the sales order, and then sends it to shipping for order fulfillment.

7. Oracle Fusion Shipping creates, and then processes the shipment. When shipping finishes delivery, it sends the ship confirm and advanced shipment notice in an OAGIS message, through Collaboration Messaging Framework, to the trading partner.

### Types of Messages That Collaboration Messaging Can Send

Collaboration Messaging can receive the following types of messages from a trading partner:

<table>
<thead>
<tr>
<th>Collaboration Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAGIS_10.1CANCEL_PO COLLAB_MSG_IN</td>
<td>Cancel purchase orders received from the trading partner.</td>
</tr>
<tr>
<td>OAGIS_10.1CHANGE_PO COLLAB_MSG_IN</td>
<td>Change purchase orders received from the trading partner.</td>
</tr>
<tr>
<td>OAGIS_10.1PROCESS_PO COLLAB_MSG_IN</td>
<td>Process purchase orders received from the trading partner.</td>
</tr>
</tbody>
</table>

Collaboration Messaging can send the following types of messages to a trading partner:

<table>
<thead>
<tr>
<th>Collaboration Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAGIS_10.1ACK_PO COLLAB_MSG_OUT</td>
<td>Send acknowledgment to the trading partner that Collaboration Messaging received the purchase order from.</td>
</tr>
<tr>
<td>OAGIS_10.1ACK_CHANGE_PO COLLAB_MSG_OUT</td>
<td>Send acknowledgment to the trading partner that Collaboration Messaging received the purchase order change from.</td>
</tr>
<tr>
<td>OAGIS_10.1PROCESS_SHIPMENT COLLAB_MSG_OUT</td>
<td>Send details to the trading partner about shipments.</td>
</tr>
<tr>
<td>OAGIS_10.1PROCESS_RCV_DEL COLLAB_MSG_OUT</td>
<td>Send details to the trading partner about purchase order deletions.</td>
</tr>
</tbody>
</table>

### Administering Business-to-Business Order Management: Procedure

You can use the Collaboration Messaging work area to set up business-to-business order management, and the Order Management and the Shipping work areas to test your set up.
In this example, assume you must create a relationship between a customer named Computer Service and Rentals who resides in Oracle Cloud Applications, and a trading partner named Computer Associates. You create an integration that achieves the following results:

- Allow Computer Service and Rentals to receive sales orders, updates, and cancellations from Computer Associates
- Allow Computer Service and Rentals to send acknowledgments and shipments to Computer Associates

**Summary of the Work**

To administer business-to-business order management, do the following work:

1. Identify the trading partner and the messages to send and receive.
2. Create a relationship between the trading partner and the customer.
4. Import source orders.
5. Process the sales order in shipping.
6. Examine the results in collaboration messaging.

This topic includes example values. You might use different values, depending on your business requirements.

**Identify the Trading Partner and the Messages to Send and Receive**

Identify the trading partner and the messages to send and receive:

1. Log into Oracle Cloud Applications with a job role that includes the B2B Messaging Administration Duty role.

   You can use one of the following job roles:

   - Order Entry Specialist
   - Order Manager
   - Order Administrator
   - Supply Chain Application Administrator
   - Warehouse Manager

2. In the Navigator, click **Collaboration Messaging**.
3. Click **Tasks**, and then click **Manage Trading Partners**.
4. On the Manage Trading Partners page, click **Actions**, and then click **Create**.

   You can use Manage Trading Partners to communicate with a trading partner directly or through a service provider. For example, you can communicate through a service provider to implement a solution where the trading partner must use EDI (Electronic Data Interchange), a modified XML format, a proprietary format, and so on. In this example, you set up the configuration to communicate directly with the trading partner.

5. In the Create Trading Partner dialog, set the following values, and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Provider</td>
<td>None</td>
</tr>
<tr>
<td>Trading Partner ID</td>
<td>ComputerAssociates</td>
</tr>
</tbody>
</table>

   Computer Service and Rentals will communicate directly with Computer Associates instead of going through a service provider, so you set this attribute to None.
### Oracle SCM Cloud

Implementing Order Management

#### Chapter 2

Implementing Order Management Cloud

---

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partner ID Type</td>
<td>Generic</td>
</tr>
</tbody>
</table>

6. In the Delivery Methods tab, click **Actions**, click **Add Row**, set the following values, and then click **Save and Close**. The Delivery Methods tab allows you to specify how to send messages to the trading partner.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>CMKDelivery00</td>
</tr>
<tr>
<td>Delivery Method</td>
<td>Web Service</td>
</tr>
<tr>
<td>Service Name</td>
<td>CollaborationMessage. Process</td>
</tr>
<tr>
<td></td>
<td>The integration uses the Process method of the CollaborationMessage web service. You must make sure that the trading partner has deployed this web service on their server.</td>
</tr>
<tr>
<td>Security Policy</td>
<td>None</td>
</tr>
<tr>
<td>Endpoint</td>
<td>Enter the URL that locates the server and port that the trading partner uses at their location as the end point for their web services. For example: <a href="http://ComputerAssociatescom7012">http://ComputerAssociatescom7012</a></td>
</tr>
<tr>
<td>User Name</td>
<td>Enter the user name that the trading partner server requires to access the endpoint.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password that the trading partner server requires to access the endpoint.</td>
</tr>
</tbody>
</table>

7. Click **Outbound Collaboration Messages**, add the following rows, and then click **Save**.

<table>
<thead>
<tr>
<th>Name</th>
<th>Collaboration Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMKORDERACK001</td>
<td>OAGIS_10.1_ACK_PO_COLLAB_MSG_OUT</td>
</tr>
<tr>
<td>CMKORDERACK002</td>
<td>OAGIS_10.1_ACK_CHANGE_PO_COLLAB_MSG_OUT</td>
</tr>
<tr>
<td>CMKSHIPCONFIRM001</td>
<td>OAGIS_10.1_PROCESS_SHIPMENT_COLLAB_MSG_OUT</td>
</tr>
</tbody>
</table>

Note the following points:
- The Outbound Collaboration Messages tab allows you to specify the messages that Collaboration Messaging sends to the trading partner.
- Set the Status to Active for each message.
- You can enter any text for the name. It is recommended that you enter text that describes the message contents.

8. Click **Inbound Collaboration Messages**, add the following messages, and then click **Save and Close**.
Note the following points:

- The Inbound Collaboration Messages tab allows you to specify the messages that Collaboration Messaging receives from the trading partner.
- Set the Status to Active for each message.

### Create a Relationship Between the Trading Partner and the Customer

In this section, you create a relationship between Computer Service and Rentals, who is a customer that you have already defined in Oracle Cloud Applications, and the trading partner, Computer Associates. You also identify the documents that you must enable for this partner.

Create a relationship between the trading partner and the customer:

1. On the Overview page, click Tasks, and then click **Manage Customer Collaboration Configuration**.
2. On the Manage Customer Collaboration Configuration page, search for the customer who will receive communications from the trading partner.

   For this example, enter **Computer Service and Rentals**. Assume that you already set up Computer Service and Rentals as a customer. If you have not set up the customer, then you must do so now. For details, see Customer Data Management Offering: Overview.

3. In the search results, if more than one row exists for this customer, then click the row that includes a check mark in the Ship to Party and in the Collaboration Configured options, and then click **Edit Collaboration Configuration**.
4. On the Edit Customer Collaboration Configuration page, in the Associated Service Providers area, click **Actions**, click **Add Row**, and then set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Provider</td>
<td>None</td>
</tr>
<tr>
<td>Trading Partner ID</td>
<td>ComputerAssociates</td>
</tr>
<tr>
<td></td>
<td>Note that you specified this ID earlier in this topic when you identified the trading partner.</td>
</tr>
<tr>
<td>Order Processing Business Unit</td>
<td>Specify the business unit that processes each sales order that Order Management Cloud receives. For this example, Computer Service and Rentals resides in the Vision Operations business unit, so choose <strong>Vision Operations</strong>.</td>
</tr>
<tr>
<td>Application Partner Code</td>
<td>Accept the default value.</td>
</tr>
</tbody>
</table>
5. In the Collaboration Documents for Service Provider area, add the following documents, click **Save and Close**, and then click **Done**:  
   - PROCESS_PO_IN  
   - CHANGE_PO_IN  
   - CANCEL_PO_IN  
   - ACKNOWLEDGE_PO_OUT  
   - ACKNOWLEDGE_CHANGE_PO_OUT  
   - PROCESS_SHIPMENT_OUT  

Note the following points:  
   - You use this area to specify the documents that this customer will communicate with the trading partner.  
   - Set the Association Status to Enabled for each document.  
   - The customer can communicate with the trading partner after you finish this step.  

### Simulate Collaboration Messaging  

Simulate collaboration messaging:  

1. Click **Tasks**, and then click **Validate Inbound Collaboration Messaging Setup**.  
2. On the Validate Inbound Collaboration Messaging Setup page, set the following values.  

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Provider</td>
<td>None</td>
</tr>
<tr>
<td>From Partner ID</td>
<td>ComputerAssociates</td>
</tr>
<tr>
<td>External Message ID</td>
<td>Enter the number that identifies the message you are testing, such as 08192016_001.</td>
</tr>
<tr>
<td>External Message Name</td>
<td>OAGIS_10.1_PROCESS_PO_COLLAB_MSG_IN</td>
</tr>
<tr>
<td>Processing Service</td>
<td>CollaborationMessage, Process</td>
</tr>
</tbody>
</table>

3. Click **Create Message Payload**.  

Notice that the Message Payload area uses the settings that you specified in step 2 to create, and then display the XML that this test will use to communicate the message. You can modify this XML, if necessary.  

4. Click **Process**.  

A separate web service in Collaboration Messaging simulates the communication from the trading partner. It receives the XML document, validates it, and then displays a Processing Confirmation dialog that includes the XML result of the test. If the test is successful, then this XML will include the following code:  

```xml
<ProcessingResultCode><Success/></ProcessingResultCode>
```
5. Make a note of the line that includes the message ID. It will resemble the following:

 `<BODID>IN_8001</BODID>`

In this example, IN_8001 is the message ID.

Import Source Orders

The simulation that you performed earlier in this topic created a simulated source order. In this section, you import this source order into Order Management, and then view it. Note that you must also import source orders in a production environment.

Import source orders:

1. Use a scheduled process to import the source orders. For details, see Using Files to Import Orders into Order Management: Procedure.

   Note the following points:
   - When you run the scheduled process named Load Interface File for Import, set the Data File to the file that the Messaging Framework created when you simulated collaboration messaging earlier in this topic. The data file will include a concatenation of the document name plus the message ID. For example, `PROCESS_PO_IN-1.0-IN_8001`.
   - When you run the Import Sales Order scheduled process, set the Source System to ORA_ELECTRONIC_DOCUMENTS. Order Management uses this predefined source system for each source order that it receives from the trading partner.
   - As an option, you can specify the unique ID for the message that you received, such as 08192016_001, in the Batch Name parameter.
   - When the Import Sales Order scheduled process finishes, Order Management can process the sales orders through order fulfillment.

2. In the Navigator, click Order Management.

3. On the Overview page, search for the simulated order, such as Demo_Order.

4. In the Search Results, note that the value in the Source Order attribute includes the order number that you simulated, such as Demo_Order_0819_001.

5. In the Order column, click the order number.

6. On the Order page, notice that the order status is Processing, and the fulfillment line status is Awaiting Shipping, which indicates that Order Management released the sales order to order fulfillment.

   The messaging framework creates an acknowledgment message, and then publishes it to the trading partner.

Process the Sales Order in Shipping

Process the sales order in shipping:

1. In the Navigator, click Shipments.

2. On the Overview page, search for the sales order that you noted in the Source Order attribute earlier in this topic, such as Demo_Order_0819_001.


4. On the Manage Shipment Lines page, click Actions, click Pick Release, and then click Save and Close.

5. On the Overview page, search Demo_Order_0819_001.

6. On the Edit Shipment Line page, notice that the Line Status is Staged, and then click the link next to Shipment.

7. On the Edit Shipment page, click Ship Confirm, and then click Save and Close.

   Shipping creates an event. Collaboration Messaging subscribes to this event.

8. On the Manage Shipments page, in the row for your shipment, notice that the ASN Status is Sent.
This status indicates that Shipping sent the ASN status to Collaboration Messaging.

**Examine the Results in Collaboration Messaging**

Examine the results in Collaboration Messaging:

1. In the Navigator, click **Collaboration Messaging**.
2. Click Tasks, and then click **Manage Collaboration Messaging History**.
3. In the Search area, set the following attribute, and then click **Search**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>PROCESS_PO_IN</td>
</tr>
</tbody>
</table>

4. In the Messages area, in the External Message ID column, click the row that references the message that you simulated earlier in this topic, such as 08192016_001.

5. Click **Actions**, and then click one of the following choices.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Source Document</td>
<td>View the input XML that Collaboration Messaging received from the trading partner.</td>
</tr>
<tr>
<td>View Output Document</td>
<td>View the output XML that Collaboration Messaging converted from the input XML, and then sent to Order Management.</td>
</tr>
</tbody>
</table>

6. Click **Done**.
7. Click **Tasks**, and then click **Manage Collaboration Messaging History**.
8. In the Search area, set the following attribute, and then click **Search**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>ACKNOWLEDGE_PO_OUT</td>
</tr>
</tbody>
</table>

9. Examine the results of the acknowledgment.
10. In the Search area, set the following attribute, and then click **Search**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>PROCESS_SHIPMENT_OUT</td>
</tr>
</tbody>
</table>

11. Examine the results of the shipment.

**Related Topics**

- Customer Data Management Offering: Overview
- Using Files to Import Orders into Order Management: Procedure
Implementing Drop Ship

Using Drop Ship with Order Management: Overview

Drop ship is a supply chain management technique where the seller relies on a supplier or contract manufacturer to build, store, and ship each sales order to the customer. The retailer does not maintain an inventory of the goods that each sales order references. You can use Order Management Cloud to automate this process.

The following flow illustrates an example drop ship flow.

Assume your customer places an order with you that requires a drop ship. Order Management sends a purchase request to Procurement, which places a purchase order with your supplier, and then your supplier ships directly to your customer. You provide a purchase order for the item and instructions that describe how to ship directly to the customer. The supplier or contract manufacturer ships the item, and your company earns a profit.

Drop ship allows you to do the following:

- Forecast and plan for future demand.
- Allow your customer to place an order with you, and for you to promise a ship date.
- Automatically place an order with your supplier.
- Allow your supplier to ship directly to your customer.
- Receive notification from your supplier when you when shipment has occurred.
- Manage order changes.

Note that Order Management only supports return to the warehouse even for a drop ship order. Order Management does not support a return from the customer directly to the supplier.
Sending Shipping Instructions and Packing Instructions in Drop Shipments

If your fulfillment uses the predefined drop ship flow, then note the following points:

- Order Management sends shipping instructions and packing instructions in one or more attachments on the fulfillment line because Oracle Fusion Purchasing only accepts these instructions in an attachment.
- Oracle Fusion Purchasing only processes attachments that include an attachment category of MISC on the fulfillment line. It ignores an attachment that contains any other value.
- An Order Entry Specialist who uses the Create Order page in the Order Management work area can set the Category attribute in the Attachments dialog to any value that this attribute displays when adding an attachment to an order line. However, Order Management sends a fixed value of MISC (Miscellaneous) to Oracle Fusion Purchasing regardless of the value of the Category attribute.
- The user can enter text in the Shipping Instructions and Packing Instructions attributes in the Shipping Details on the Create Order page. However, Order Management does not send this text to Oracle Fusion Purchasing because Oracle Fusion Purchasing only accepts them in an attachment.
- You must set the attachment category to MISC on the order line of each source order that you import.

For details about attachment categories and how to set them, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then download the attachment that describes how to use attachments.

Using Drop Ship with Order Management: How it Works

You can use Order Management Cloud to manage the flow of information through the systems and business functions that it uses to run the drop ship flow. This topic describes the systems and functions involved.

The following diagram illustrates how Order Management runs a drop ship flow. Each horizontal row in this diagram represents a system or business function that this flow uses. The drop ship flow is a variation of the order-to-cash flow. For details, see Order-to-Cash: How It Works.

You can modify how Order Management performs some of these steps. For example, you can use Manage Sourcing Rules to modify how Global Order Promising considers sourcing rules, you can use Manage Drop Ship Financial Flows to modify
how Supply Chain Financial Orchestration runs the financial flow, and so on. For details, see Implementing Drop Ship in Order Management: Procedure.

**Explanation of Callouts**

Order Management does the following work when it runs a drop ship flow:

1. An order capture system captures a source order. Although Order Management comes predefined to capture sales orders directly in Order Management, you can also capture source orders in an external order capture system. For details, see Order-to-Cash with Order Capture Systems: How it Works.
2. The Order Orchestration component in Order Management assigns an orchestration process to the sales order, uses the Schedule Order task to start the scheduling work for this sales order, and then sends a scheduling request to Global Order Promising Cloud.

3. Global Order Promising considers sourcing rules, supplier calendar, capacity, and so on to identify the supplier and supplier site that can fulfill the order in the most efficient way. If the sales order specifies a supplier, then Global Order Promising uses this supplier.

4. Order Orchestration gets the requisition organization from Supply Chain Financial Orchestration so that it can create a purchase request. You can use the Manage Drop Ship Financial Flows task in Supply Chain Financial Orchestration to configure a relationship between the selling business unit and the requisition organization.

5. Order Management sends a purchase order request to Oracle Procurement Cloud.

6. Procurement creates a purchase requisition and a purchase order, and then sends the purchase order to the supplier. If a blanket purchase agreement exists, then Procurement might source the requisition from this agreement.

7. The supplier examines the purchase order and uses your shipping fulfillment system to ship the goods to the customer. The enterprise supplier also communicates with the Receiving component of Oracle Inventory Management Cloud to cost the shipment.

8. The supplier uses a supplier portal to enter an advance shipment notice (ASN). Order Management sends this notice to interested parties and Financial Orchestration. For details about the supplier portal, see Supplier Portal Overview: Explained.

9. The Receiving component of Oracle Inventory Management Cloud creates a logical receipt in the receiving organization. It does not create a physical receipt because Order Management ships goods directly to the customer.

10. Financial Orchestration runs a financial flow that comes predefined to handle drop ships, and that specifies how to handle the flow that runs from the supplier to the customer.


12. Order Management waits to receive the advance shipment notice from receiving that indicates that the goods have been shipped, and that the customer acknowledged receipt.

13. Order Management communicates with Oracle Receivables to create an invoice and process payments.

**Setups That Are Required to Implement Drop Ship**

The following setups are required to implement drop ship. Each feature comes predefined as already available except for Fulfill a Customer Order Through Drop Shipment and Handle Landed Cost Charges for Drop Ship Receipts. A Predefined Job Role includes each feature.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Setup Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fulfill a Customer Order Through Drop Shipment</td>
<td>Yes</td>
</tr>
<tr>
<td>Automatic Change Management on Drop Ship Orders</td>
<td>No</td>
</tr>
<tr>
<td>Change Drop Ship Orders from Check Availability Page</td>
<td>No</td>
</tr>
<tr>
<td>Partial Shipment on Drop Ship Orders</td>
<td>No</td>
</tr>
<tr>
<td>Analytics with Supplier Source</td>
<td>No</td>
</tr>
<tr>
<td>Create Relationship between Selling Business Unit and Requisition Business Unit</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Implementing Order Management Cloud

This topic describes how to set up Order Management Cloud so that it supports the drop ship flow.

**Summary of the Work**

To implement drop ship in Order Management, do the following work:

1. Set up Oracle Applications.
2. Enable the features that drop ship requires.
3. Manage items.
5. Manage financial flows for drop ship.
6. Manage order management parameters.
7. Manage suppliers and supplier sites.
8. Manage agreements.
9. Manage orchestration process definitions.
10. Test your work.

Each of these steps are required except for Manage Agreements and Manage Orchestration Process Definitions, which are optional.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Setup Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automate Financial Flow</td>
<td>Yes</td>
</tr>
<tr>
<td>Order Promising for Drop Ship</td>
<td>Yes</td>
</tr>
<tr>
<td>Create ASN for Drop Ship Receipts</td>
<td>Yes</td>
</tr>
<tr>
<td>Create Accounts Payable Invoices for Drop Ship Receipts</td>
<td>Yes</td>
</tr>
<tr>
<td>Process Accounting for Drop Ship Transactions</td>
<td>No</td>
</tr>
<tr>
<td>Handle Landed Cost Charges for Drop Ship Receipts</td>
<td>Yes</td>
</tr>
<tr>
<td>Report Gross Margins for Drop Ship Orders</td>
<td>No</td>
</tr>
<tr>
<td>Report In-transit Inventory for Drop Shipments</td>
<td>No</td>
</tr>
</tbody>
</table>
Set Up Oracle Applications

You must set up various Oracle applications so that they support drop ship. For example, the Order Management offering references the Global Order Promising application to collect supply data from multiple sources. To start your implementation work for drop ship, it is recommended that you perform the following tasks:

1. Perform the Define Blanket Agreement and ASL task in Procurement. Requires a user login that can set up Procurement.
2. Perform the Create Drop-Ship Validation Org task in Inventory Management. Requires a user login that can set up Inventory Management.
3. Perform the following tasks in Global Order Promising. Requires a user login that can set up Global Order Promising:
   - Manage Data Collections.
   - Define Item Processing Lead Times.
   - Manage Sourcing Rules and Manage ATP Rules. Set up sourcing rules and ATP rules so that they support the drop ship flow. For details, see ATP Rules, Allocation Rules, and Sourcing Rules: How They Work Together.
4. Perform the Manage Trade Operations task in the Manufacturing and Supply Chain Materials Management offering.
5. Perform the following tasks:
   - Define Shipping Network
   - Define Transit Lead Times
   - Manage Assignment Sets
   - Maintain Supply Network Model

For details about how to perform some of these tasks, see the guide titled Oracle SCM Cloud, Implementing Common Features for Oracle SCM Cloud.

Enable the Features That Drop Ship Requires

Enable the features that drop ship requires:

1. Implement the Order Management offering.
   For details, see the Implementing Order Management Cloud chapter in the guide titled Oracle SCM Cloud, Implementing Order Management.
2. Log into Order Management with administrative privileges.
3. In the Navigator, click Setup and Maintenance.
5. Click Actions, and then click Change Feature Selection.
6. On the Edit Features page, in the Drop Ship row, add a check mark to the Enable option, and then click Done.
   This feature allows you to access the setup tasks that you must perform in Oracle Supply Chain Financial Orchestration so that you can implement the drop ship flow.
7. Click Actions, and then click Go to Offerings.
8. On the Setup page, click Procurement.
9. Click Actions, and then click Change Feature Selection.
10. On the Edit Features page, in the Customer Sales Order Fulfillment row, add a check mark to the Enable option, and then click Done.
    This feature allows Oracle Procurement to accept purchase requests for sales orders that a drop ship supplier fulfills, and to display some of the attributes that you use to set up drop ship.
Manage Items

Allow items to participate in drop shipments:

1. In the Navigator, click **Product Information Management**.
2. On the Overview page, click **Tasks**, and then click **Manage Items**.
3. On the Manage Items page, search for an item that your supplier must ship directly to your customer, such as Sentinel Deluxe Desktop.
4. In the Search Results list, click the **link** in the Item column for the item that you must manage.

   For example, in the Sentinel Deluxe Desktop row, click **AS18947**.
5. On the Edit Item page, click **Specifications**, and then click **Purchasing**.
6. In the Item Organization: Purchasing area, set the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purchasable</td>
<td>Set to Yes. Allows this item to participate in a drop shipment.</td>
</tr>
</tbody>
</table>

7. If a blanket purchase agreement does not exist for this item, then, in the Pricing area, enter a number in the **List Price** attribute.
8. Click **Save**.
9. Repeat steps 3 through 8 for each item that you must make purchasable.

   You must specify each item that your supplier ships directly to your customers. You specify this item as purchasable in the organization that is responsible for purchasing each item. The Product Information Management work area comes predefined to set the Purchasable attribute to Yes, so you must modify it only if you have set it No.

   For details about how to make a large number of items purchasable, see the guide titled Oracle SCM Cloud, Implementing Product Management.

Manage Sourcing Rules

You can specify a sourcing rule that already includes details for the supplier and the supplier site. Global Order Promising Cloud evaluates this sourcing rule and considers the supplier calendar, supplier capacity, and supplier lead times when it promises the order. For details, see ATP Rules, Allocation Rules, and Sourcing Rules: How They Work Together.

Manage the sourcing rule:

1. In the Navigator, click **Order Promising**.
2. On the Overview page, click **Tasks**, and then click **Manage Sourcing Rules**.
3. On the Manage Sourcing Rules page, search for the sourcing rule that you must modify, such as DOO-DS-Rule.
4. In the Search Results list, click the **sourcing rule** that you must edit, click **Actions**, and then click **Edit**.
5. On the Edit Sourcing Rule page, verify the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization Assignment Type</td>
<td>Make sure the type is Global. If it is not Global, then you cannot use this sourcing rule.</td>
</tr>
</tbody>
</table>

6. In the Effective Start Date area, to add a sourcing rule for your drop ship flow, click **Actions**, click **Add**, and then set the following values.
Attribute | Description
--- | ---
Organization | Leave empty. The Organization is typically the warehouse that stores inventory for a flow that does not include a drop ship.

Type | Choose Buy From. Global Order Promising interprets Buy From to indicate that this value must come from a drop ship supplier.

Supplier | Choose the supplier who will drop ship the item.

Supplier Site | Choose the site that the supplier uses to store the item that this supplier will drop ship.

7. Click **Save and Close**.
8. Click **View Assignment Sets**.
9. In the Assignment Sets dialog, choose an **assignment set**, such as AYY-OP-ASET, and then click **Done**.
10. On the Edit Assignment Set page, in the Sourcing Assignments area, click **Actions**, click **Add Row**, and then assign an item to this sourcing rule. Use the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment Level</td>
<td>Set to Item. This setting assigns the sourcing rule to the item.</td>
</tr>
<tr>
<td>Item</td>
<td>Choose an item that the supplier that you specified in step 6 supplies. For example, if your supplier drop ships video cards, then choose CM22111.</td>
</tr>
<tr>
<td>Sourcing Type</td>
<td>Set to Sourcing Rule. This value associates the sourcing rule with the assignment set so that Order Management Cloud can use the rule to assign the item.</td>
</tr>
<tr>
<td>Sourcing Rule or Bill of Distribution</td>
<td>Set to Drop Ship.</td>
</tr>
</tbody>
</table>

11. Repeat step 10 for each item that the supplier you specified in step 6 supplies through a drop ship.

Note that you can also specify the following items for the supplier.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Supplier Calendar</strong></td>
<td>Specify the working days for the supplier. Note the following:</td>
</tr>
<tr>
<td></td>
<td>• You can modify the calendar for each supplier site.</td>
</tr>
<tr>
<td></td>
<td>• The calendar that you specify can be different from the calendar that the Supplier Site uses.</td>
</tr>
<tr>
<td></td>
<td>• Global Order Promising considers only working days when it calculates and incorporates lead times.</td>
</tr>
<tr>
<td><strong>Supplier Capacity</strong></td>
<td>Specify the supplier capacity according to item, supplier, and supplier site. Order Promising measures the supplier capacity that exists on the arrival date.</td>
</tr>
<tr>
<td><strong>Supplier Lead Times</strong></td>
<td>You specify and collect the lead times for item processing on the item master in Oracle Fusion Product Model. You can specify a lead time for each supplier in Oracle Fusion Global Order Promising.</td>
</tr>
</tbody>
</table>
Manage Financial Flows for Drop Ship

Specify the receiving trade organization for each selling business unit that your enterprise contains. For details about how the settings that you make in this topic affects the financial flow, see Managing Drop Ship Financial Flows: Explained:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, search for, click **Order Management**.
3. Search for, and then open Manage Drop Ship Financial Flows.
4. On the Manage Drop Ship Financial Flows page, in the Search Results area, click **Actions**, and then click **Create**.
5. Set the Name and Priority, set a value for the following attribute, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplier Ownership Change Event</td>
<td>Choose one of the following values. Start the ownership change when the financial flow for this drop ship order receives:</td>
</tr>
<tr>
<td></td>
<td>◦ <strong>ASN From Supplier</strong>. Receives an advance shipment notice (ASN) from the supplier.</td>
</tr>
<tr>
<td></td>
<td>◦ <strong>AP Invoice Match</strong>. Receives an invoice from the supplier.</td>
</tr>
</tbody>
</table>

For details about how to set this attribute, see Specifying Events That Indicate an Ownership Change: Explained.

6. In the Selling BU to Receiving BU Relationships area, click **Actions**, click **Add Row**, and then specify the relationship.
   For details about how to specify this relationship, see Specifying Business Units for Financial Flows That Drop Ship Uses: Explained.
7. Repeat steps 3 through 4 for each selling business unit that your enterprise contains.
8. Click **Save and Close**.

Manage Order Management Parameters

Specify an Order Management user who the buyer can contact to help resolve a problem that might occur with a fulfillment line that involves a drop ship supplier:

1. On the Setup page, search for, and then open Manage Order Management Parameters.
2. On the Manage Order Management Parameters page, click **Preparer for Procurement**.
3. In the Preparer for Procurement list, add one or more preparer, and then click **Save and Close**.

You must specify at least one value so that Order Management Cloud can successfully send a purchase request to Oracle Procurement Cloud. The default value applies across all business units. You can add a separate preparer for each business unit. For details, see Managing Order Management Parameters: Procedure.

Manage Suppliers and Supplier Sites

Specify the client business unit that Procurement will use to requisition and process invoices for the supplier site. For details, see Supplier Sites and Supplier Site Assignments: Explained:

1. Log in with a user role that allows you to edit suppliers.
2. In the Navigator, click **Suppliers**.
3. On the Overview page, click **Tasks**, and then click **Manage Suppliers**.
4. On the Manage Suppliers page, search for the supplier that you must manage, such as GVR_DS_SUPPLIER1.
5. In the Search Results area, click the **supplier** you must manage, such as GVR_DS_SUPPLIER1.
6. On the Edit Supplier page, click **Sites**.
7. In the Site column, click a **site**, such as GVR_SUP1_SITE1.
The supplier site that you select must reference a procurement business unit. This business unit authorizes procurement services for the site.

8. On the Edit Site page, make sure the Purchasing option contains a check mark.

This option specifies that this supplier site fulfills a purchasing role for the client business unit.

9. Click **Site Assignments**, and then set the following value. If necessary, click **Actions**, and then click **Create** to add a client business unit.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client BU</td>
<td>Choose the client business unit that will requisition and process invoices for the supplier site.</td>
</tr>
</tbody>
</table>

10. Repeat step 9 for each requisition business unit that resides at the site that might request to purchase a drop ship item.

11. Click **Save and Close**.

12. Repeat steps 4 through 10 for each supplier that you must manage.

### Manage Agreements

Oracle Fusion Purchasing allows your buyer to create a blanket purchase agreement for the items that it will drop ship from a supplier. You must define an agreement for each supplier and supplier site, and associate one or more items with this agreement. You do this work when you set up purchasing. This topic describes how to modify this setup so that it supports drop ship. Purchasing uses the prices that the agreement specifies to set default values in the purchase documents. For details, see Blanket Purchase Agreement Lines: Points to Consider.

You manage agreements differently for a drop ship that includes a configured item. For details, see Setting Up Drop Ship for Configured Items: Procedure.

**Manage agreements:**

1. Log in with a user role that allows you to create an agreement.
2. In the Navigator, click **Purchasing**.
3. Click **Tasks**, and then click **Manage Agreements**.
4. On the Manage Agreements page, choose a value in the Procurement BU attribute, such as Vision Operations, and then click **Search**.
5. In the Search Results list, in a row that includes an agreement with a supplier who will drop ship your item, click the link in the Agreement column.
6. On the Agreement page, click **Actions**, and then click **Edit**.
7. On the Edit Document page, click **Controls**, and then set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically Generate Orders</td>
<td>Set to one of the following values:</td>
</tr>
<tr>
<td></td>
<td>Contains a check mark. Purchasing will automatically convert each requisition that it sources from this blanket purchase agreement. It will convert each requisition to a purchase order.</td>
</tr>
<tr>
<td></td>
<td>Does not contain a check mark. Your buyer must do this conversion manually in the Purchasing work area.</td>
</tr>
</tbody>
</table>
### Attribute

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Purchasing examines eligibility according to the requisition to purchase order, even if Automatically Generate Orders does not contain a check mark. For example, it makes sure each of the following items is valid:</td>
</tr>
</tbody>
</table>
|                                  | ◦ Buyer  
|                                  | ◦ Supplier  
|                                  | ◦ Supplier site  
|                                  | ◦ Source agreement  
|                                  | ◦ Item is purchasable in the inventory organization of the procurement business unit  |
| Automatically Submit for Approval| It is recommended that you make sure this option contains a check mark. If it does not, then Order Management will create the order with an incomplete status, and the buyer must manually submit it for approval. |
| Use Customer Sales Order         | Set to one of the following values:  
|                                  | ◦ Contains a check mark. Purchasing will group requisition lines that reference the same sales order number. It will group these lines on a single purchase order.  
|                                  | ◦ Does not contain a check mark. Purchasing will not do this grouping.  |

8. Click **Save**, and then click **Save and Close**.
9. Repeat steps 5 through 7 for each supplier who participates in your drop ship flow.

### Manage Orchestration Process Definitions

Order Management comes predefined with the following orchestration process:

- **DOO_OrderFulfillmentGenericProcess**

It contains branches that run under the following conditions:

- If the shipment is a drop shipment, then run the Create Shipment Request branch.
- If the item in the inventory organization is enabled for back-to-back shipping, then run the back-to-back branch.
- If the first two conditions are false, then run the Create Reservation branch.

This branch gets the shipment from on-hand quantity in your warehouse. You can use this predefined orchestration process as the default process assignment in your Assign and Launch rule. You can also set up an orchestration process to meet your business requirements. For details, see Defining Orchestration Processes: Explained.

Examine the predefined orchestration process:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Orchestration Process Definitions.
5. Examine the steps and branches.
6. If necessary, make a copy of this process, and then modify it so that it meets your business requirements, or create a new orchestration process and use **DOO_OrderFulfillmentGenericProcess** as a starting point.

### Test Your Work

Create a sales order that uses drop ship. For details, see Using Drop Shipments: Procedure.
Managing Drop Ship Financial Flows: Explained

Financial Orchestration Cloud controls the change in ownership of an item that it processes during a drop ship flow. For example, to transfer ownership from the selling business unit to the requisition business unit. The financial flow allows you to create cost accounting distributions that track the costs and ownership liability each time a transfer occurs between parties, including the supplier, one or more internal organizations, and the customer. It also creates an intercompany invoice for each internal ownership transfer, when necessary.

You can specify multiple requisition business units to manage and own multiple transactions that request the item. You can also specify the selling business unit that resides in the selling legal entity that sells the item.

The following diagram provides a high-level illustration of the process that Financial Orchestration performs during a drop ship flow.

**Explanation of Callouts**

Financial Orchestration performs the following work during a drop ship flow:

1. Receives events. Financial Orchestration captures the physical supply chain event each time one occurs in the drop shipment flow. For example, when the supplier sends the Advance Shipment Notice to indicate that shipment of goods occurred.
2. Identifies the financial flow to run. Uses the following details to identify the financial flow to use:
   - Purchase order for the drop ship
o Sales order details that it gets from source documents
o Internal selling and buying business units
o Financial orchestration qualifiers
o Priority of the financial orchestration flow

3. Creates tasks.

The flows that Financial Orchestration runs vary depending on the number of internal organizations that are involved:

- **One organization.** The selling business unit and the requisition business unit are the same unit, and Financial Orchestration uses only one financial flow for the drop ship.

- **Multiple organizations.** Financial Orchestration might run through multiple business units that involve procurement financial flows and shipment financial flows.

For details about the settings that you can make that affect this flow, see the Manage Drop Ship Financial Flows subtopic in Implementing Drop Ship in Order Management: Procedure. For details about financial orchestration, see Financial Orchestration Flows: Explained.
Specifying Business Units for Financial Flows That Include Drop Shipments: Explained

The following diagram illustrates the relationship between the selling business unit, requisition business unit, suppliers, and customers in a financial flow that includes a drop shipment.

The Manage Drop Ship Financial Flows page sets the Receiving BU and the Receiving Legal Entity according to the value that you set in the Receiving Trade Organization field. For example, assume you set Receiving Trade Organization to Vision Operations, and that you set up the application to create a requisition in the Vision Operations inventory organization for any drop ship sales order that the Vision Operations business unit creates. In this situation, each of the following fields will contain a value of Vision Operations:

- Selling BU
- Selling Legal Entity
- Receiving Trade Organization
• Receiving BU
• Receiving Legal Entity
• Requisition BU

Order Management Cloud gets the value for the requisition organization field and the Requisition BU field from the purchase requisition that it creates for each drop ship order that involves an external supplier, and that requires a requisition organization.

Specifying the Receiving Trade Organization
Choose the organization that does the following work for the drop ship flow when you specify the Receiving Trade Organization field on the Create Drop Ship Financial Flow page:
• Places the requisition for the goods
• Receives the drop ship purchase order
• Performs the receipt accounting and shipment accounting for the drop shipment

Setting the Selling Business Unit and the Receiving Business Unit to Different Values
Financial Orchestration Cloud allows you to set up and run the financial flow according to ownership transfers that occur between the parties that are involved in the drop ship flow. You can configure the relationship that exists between the selling business unit and the receiving business unit. Financial Orchestration uses this relationship when it creates a purchase requisition for a drop shipment.

If you set the value for the Selling BU field to a value that is different from the value that the Receiving BU field contains, then Financial Orchestration determines whether or not it must perform more financial and accounting transactions according to the procurement and shipping flows that the application defines.

Defining Different Requisition Trade Organizations
If you set the value for the Selling BU field to a value that is different from the value that the Receiving BU field contains, then a shipment financial flow must exist between the receiving business unit and the selling business unit. Financial Orchestration determines whether or not this flow exists when you create this relationship. If it does not, then Financial Orchestration displays an error message.

If a procurement financial flow exists in Financial Orchestration, and if Financial Orchestration can use it for the purchase order that the drop ship references, then Financial Orchestration uses this procurement financial flow when it does financial orchestration for the drop ship. For example, assume you must create a requisition that references China for each item that resides in the Metal category, and create a requisition that references the United States for each item that resides in the Plastic category. In this situation, you can create drop ship financial flows that use the following configuration.

<table>
<thead>
<tr>
<th>Drop Ship Financial Flow</th>
<th>Inventory Organization</th>
<th>Financial Orchestration Qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow 1</td>
<td>Resides in China as the receiving trade organization.</td>
<td>Uses Metal as the item category.</td>
</tr>
<tr>
<td>Flow 2</td>
<td>Resides in the United States as the receiving trade organization.</td>
<td>Uses Plastic as the item category.</td>
</tr>
</tbody>
</table>

For details about how to prioritize a drop ship financial flow, see the Implementing Drop Ship section in the guide titled Oracle Supply Chain Management Cloud, Implementing Order Management, on Oracle Help Center at https://docs.oracle.com/cloud/latest/scmcs_gs/FAIOM/toc.htm.
Specifying Events That Indicate an Ownership Change: Explained

To specify the event that indicates the ownership change, you can specify to use the receipt of the advanced shipment notice (ASN) from the supplier, or to use the arrival and entry of the accounts payable invoice. You specify these values in the Supplier Ownership Change Event field on the Manage Drop Ship Financial Flows page.

Using ASN From Supplier

If the supplier site sends an advanced shipment notice (ASN), then it is recommended that you set the Supplier Ownership Change Event field to ASN From Supplier. The arrival of the advanced shipment notice starts the downstream processes that cost and bill the shipment, including creating the drop ship receipts, starting the billing processes, and starting the financial orchestration process for the drop ship order. Order Management Cloud loads the advanced shipment notice into the Receiving component, which creates a type of drop ship flow known as receipt and delivery that records an event. Receiving sends this event to Financial Orchestration and Order Management. Financial Orchestration uses the status information and details that this event contains to start financial orchestration processes. These processes perform trade transactions in costing, process intercompany receivable invoices, process intercompany payable invoices, and so on. Order Management uses these details to start billing processes and order orchestration processes.

The supplier can use one of the following features on the supplier portal to create the advanced shipment notice:

- Create ASN
- Create ASBN
- Upload ASN
- Upload ASBN

If the supplier uses informal communication, such as email, to report the shipment of a purchase order that references a drop shipment, then the warehouse manager can also use these features on the Receipts page of the Warehouse Operations work area to manually enter the advanced shipment notice. You can also upload advanced shipment notices electronically through XML or EDI. For details, see Supplier Portal Overview: Explained, and Creating ASNs and ASBNs: Overview.

Using AP Invoice Match

If the supplier site does not send an advanced shipment notice, then it is recommended that you set the Supplier Ownership Change Event field to AP Invoice Match. Financial Orchestration receives the events that are related to the accounts payable invoice from Accounts Payable, and then uses one of these events to start the financial orchestration processes and to inform the receiving process. The arrival and entry of the accounts payable invoice starts the downstream processes that cost and bill the shipment. The receiving process works the same way that it does for ASN From Supplier, except the process never creates an advanced shipment notice. Order Management creates the receipt and delivery drop ship when it creates the accounts payable invoice for the purchase order that the drop ship references.

The following scheduled processes handle processing:

- **Transfer Invoice Details to Supply Chain Financial Flow Orchestration.** Sends details about validated invoices, canceled invoices, and corrected invoices to Financial Orchestration.
- **Transfer Ownership Change Events to Receiving.** Sends details about the AP Invoice Match from Financial Orchestration to the receiving process.
Setting Up Drop Ship for Configured Items: Procedure

To allow your users to drop ship a configured item, you must define an agreement for the configuration model that specifies the configured item and for the configure options that might be part of the configure-to-order shipment.

A hybrid configuration model is a type of configuration model that includes an assemble-to-order configuration model as the child and a purchase-to-order configuration model as the parent. Order Management Cloud uses one blanket purchase agreement to source the assemble-to-order options, and a different blanket purchase agreement to source the purchase-to-order options. So, Order Management issues separate purchase orders to the supplier.

Set up drop ship for configured items:

1. Log in with a user role that allows you to edit values in the Procurement work area.
2. In the Navigator, click Purchasing.
3. Click Tasks, and then click Manage Agreements.
4. Click Actions, and then click Create.
5. In the Create Agreement dialog, set the following values, and then click Create.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Style</td>
<td>Select the following value:</td>
</tr>
<tr>
<td></td>
<td>◦ Configure to Order Blanket Purchase Agreement</td>
</tr>
<tr>
<td>Supplier</td>
<td>Select the supplier who will drop ship the configured item.</td>
</tr>
</tbody>
</table>

6. On the Edit Document page, in the Lines area, add at least one line, and then click Submit.

For details about how to import multiple agreements, see Understanding File-Based Data Import and Export on Oracle Help Center at http://docs.oracle.com/cloud/latest/salescs_gs/docs.htm.

More Details About Implementing Order Management

Assigning Job Roles in Order Management: Explained

A duty role allows Order Management Cloud to access the web services that it uses to communicate with an external system according to the user who is currently logged in. You must make sure that any job roles you create can reference the duty roles that Order Management requires.

For example, if a user releases a sales order for shipping, then the job role that this user uses must reference the duty roles that allow Order Management to call the shipping system and to receive details about the shipment from this shipping system. In this example, you must assign a job role that references the following duty roles:

- Shipment Request Processing Web Service Duty
- Orchestration Order Shipping Web Service Duty
Predefined Job Roles

The following job roles come predefined with Order Management.

<table>
<thead>
<tr>
<th>Job Role</th>
<th>Description</th>
</tr>
</thead>
</table>
| Order Entry Specialist        | This job role includes the following responsibilities:  
  • Create new sales orders, update existing sales orders, and create return sales order.  
  • Create sales orders in Order Management or modify sales orders that Order Management imports from a source system.  
  • Monitor order fulfillment.  
  • Work directly with customers who purchase items.  
  This job role includes the following duty roles:  
  • FSCM Load Interface Administration Duty  
  • Item Inquiry Duty                                                                                                                                                                                                                                                                                                                                                             |
| Order Manager                 | This job role includes the following responsibilities:  
  • Manage sales orders created in Order Management or imported from a source system. Makes sure Order Management books sales orders so that they can proceed to order fulfillment.  
  • Approve sales orders that require approval.  
  • Work with other professionals in the organization, such as pricing administrator, customer contract manager, credit manager, and accounts receivable manager, to help determine configuration requirements and troubleshoot problems.  
  This job role includes the following duty roles:  
  • Orchestration Order Monitoring Duty  
  • Orchestration Order Management Duty  
  • Orchestration Order Scheduling Duty                                                                                                                                                                                                                                                                                                                                                     |
| Order Administrator           | This job role includes the following responsibilities:  
  • Set up and maintain Order Management Cloud so that it supports order entry and order fulfillment.  
  • Set up rules, policies, constraints, and so on. For example, set up defaulting rules, order entry preferences, order entry privileges, orchestration processes, change order rules, process planning, jeopardy conditions, order statuses, and hold definitions.  
  • Set up of order orchestration.  
  Each of these job roles include the following duty roles:  
  • Orchestration Order Administration Duty  
  • Orchestration Infrastructure Administration Duty                                                                                                                                                                                                                                                                                                                                                   |
<p>| Supply Chain Application Administrator |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |</p>
<table>
<thead>
<tr>
<th>Duty Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Orchestration Decomposition Web Service Duty</td>
<td>Communicate with source systems so that Order Management can separate source orders during transformation.</td>
</tr>
<tr>
<td>Orchestration Order Activity Management Web Service Duty</td>
<td>Communicate with a fulfillment system that can process an activity.</td>
</tr>
<tr>
<td>Orchestration Order Billing Web Service Duty</td>
<td>Communicate with a system that processes the billing for each sales order.</td>
</tr>
<tr>
<td>Orchestration Order External Integration Web Service Duty</td>
<td>Allow an external system to call Order Management.</td>
</tr>
<tr>
<td>Orchestration Order Fulfillment Web Service Duty</td>
<td>Allow a fulfillment system to send status updates to Order Management through the fulfillment task layer.</td>
</tr>
<tr>
<td>Orchestration Order Inquiry Web Service Duty</td>
<td>Extract order details from a sales order.</td>
</tr>
<tr>
<td>Orchestration Order Receiving Web Service Duty</td>
<td>Receive a return for a sales order.</td>
</tr>
<tr>
<td>Orchestration Order Shipping Web Service Duty</td>
<td>Process the shipment of a sales order.</td>
</tr>
<tr>
<td>Orchestration Order Template Web Service Duty</td>
<td>Allow a fulfillment system to send status updates for fulfillment tasks through the template task layer.</td>
</tr>
</tbody>
</table>


Enabling Offerings: Explained

Offerings and their functional areas are presented in an expandable and collapsible hierarchy to facilitate progressive decision making regarding whether or not you want to implement them. An offering or its functional areas can either be opted into or not opted into for implementation. Implementation managers decide which offerings to enable for implementation. Although all of the functional areas that represent core functionality of an offering are automatically enabled for implementation when a parent offering is enabled for implementation, you can select which of the optional functional areas are enabled. You can identify which functionality is already opted into by looking at the check box in the Enable column.
Configuring Offerings: Procedure

Enable offerings to modify functionality so that it matches the services you plan to implement.

Enable Offerings

To enable offerings, follow these steps:

1. Select Navigator > My Enterprise > Offerings work area to open the Offerings page. You need the Configure Oracle Fusion Applications Offering privilege.
2. Select the offering you want to implement.
3. Click Opt In Features button to go to the Opt In page.
4. In the Opt In page, select the Enable check box for the offering.
5. Review functional area hierarchy. Select the Enable check box to opt into functional areas as applicable to your business operations.
6. Click the Edit icon in the Features column for the functional area you enabled to opt into and enable applicable features.
   - Depending on the feature type, a check box for Yes or No features or an Edit icon for single and multiple choice features is displayed in the Enable column.
   - To enable a feature, select the check box for Yes or No types or click Edit and select the appropriate choices for single and multiple choice features.
7. Click Done when you’re finished to return to the Opt In page.
8. Click Done to return to the Offerings page.

Repeat the same steps for each offering you want to implement or if you must change the opt-in configuration of any functional areas or features of an enabled offering.

Related Topics
- Configuring Offerings

Defining Extensions That Modify Order Management Cloud: Overview

You can use the Manage Extensions page in the Setup and Maintenance work area to write your own Groovy script that modifies your Order Management Cloud deployment, and to specify the extension point that determines when to run this script.

An order management extension is a Groovy script that runs at an extension point during order creation. An extension point is an event that you can specify that starts the extension. You can use a flexfield to modify the data model, or Page Composer to modify the user interface, but you cannot use them to programmatically modify logic or to modify your order management implementation. Instead, you can use an order management extension.

For details, see Defining Extensions That Modify Order Management Cloud (Doc ID 2252683.1) on My Oracle Support at https://support.oracle.com.

Related Topics
- Defining Extensions That Modify Order Management Cloud (Doc ID 2252683.1)
3 Integrating Order Management Cloud

Integrating Order Management with External Systems: Overview

Order Management Cloud comes predefined with integrations to other Oracle cloud services so that you can use fulfillment processes that require minimal set up to get them up and running in your environment. For example, Order Management and Oracle Fusion Inventory Management work together to fulfill each sales order that an Order Entry Specialist can enter directly in Order Management or that you import from a source system.

Order Management also provides web services that you can use to integrate with some other Oracle application, a third-party cloud application, or an on-premise application that your supply chain requires to complete the order-to-cash process:

- **Integrate cloud processes.** Integrate cloud processes, including order-to-cash, drop-ship, back-to-back, configure-to-order, or internal orders. The integration includes predefined processes and simplified set up.

- **Integrate applications.** Order Management comes integrated with the following Oracle cloud services:
  
  - Oracle Inventory Cloud
  - Oracle Cost Management Cloud
  - Oracle Manufacturing Cloud
  - Oracle Procurement Cloud
  - Oracle Product Hub Cloud
  - Oracle Financials Cloud
  - Oracle Configure, Price, and Quote Cloud.

- **Integrate web services.** Use a web service to do batch or real-time order import, send a request to a fulfillment system, receive a status from a fulfillment system, or integrate to some other cloud or on-premise system. Inventory, shipping, receiving, finance, and order management are each an example of a fulfillment system.
The following illustration includes some of the integrations that you can set up with Order Management.

For details, see the chapters that describe integration and importing source orders in the guide titled Oracle SCM Cloud, Implementing Order Management, on Oracle Help Center at https://docs.oracle.com/cloud/latest/scmcs_gs/FAIOM/toc.htm.

For more details, see the following documents on My Oracle Support at https://support.oracle.com:

- Search for Doc ID 1536633.1 (Non-Fusion Fulfillment System Connectors in DOO).
- Search for Doc ID 2051640.1 (Using Web Services with Oracle Fusion Order Management Cloud).

### Integrating Order Management with Upstream Source Systems

#### Creating User Credential Keys: Procedure

You must create a user credential key to integrate Order Management Cloud with an external service.

The external interface layer is an intermediary layer that uses open access protocols, such as HTTP, so extra security setup is required. You must make sure that the user credential is valid in the external system and in the security certificate so that the integration can encrypt and decrypt messages.
Create a user credential key:

1. Use Oracle Wallet Manager to add a user credential key to a credential map. For details, see Oracle Fusion Middleware Security Guide 11g Release 1 (11.1.1). You must use the administration privilege and administrator role.
2. In Oracle Wallet Manager, in the Select Map list, select oracle.wsm.security.
3. Enter the user credential key, user name, and password from the service that you are integrating with Order Management.
4. Log into Order Management with administrative privileges.
5. In the Navigator, click **Setup and Maintenance**.
6. On the Setup page, click **Order Management**.
7. Search for, and then open Manage External Interface Web Service Details.
8. On the Manage Connector Details page, click **Actions**, click **Add Row**, and then set the values that you set in steps 1 through 3.

**Connecting Order Management to Source Systems: Procedure**

If your deployment must integrate with an external system, then you can register a connector that allows Order Management Cloud to communicate with it. You must create, deploy, and register this connector. This topic describes how to register the connector and connect Order Management to an external source system, such as Oracle Configure, Price, and Quote Cloud. To do this, you add a connector that uses a web service that communicates order details with the source system.

The external interface layer is an intermediary layer that uses open access protocols, mainly HTTP, so extra security setup is required. You must make sure the user credential is valid in the source system and in the security certificate so the integration can encrypt and decrypt messages. For details about how to call a web service, search for Doc ID 2051640.1 (Using Web Services with Oracle Fusion Order Management Cloud) on My Oracle Support at https://support.oracle.com. For details about Oracle Configure, Price and Quote Cloud, see https://cloud.oracle.com/cpq.

**Summary of the Work**

To connect Order Management to a source system, do the following work:

1. Create a user credential key. For details, see Creating User Credential Keys: Procedure.
2. Define the source system.
3. Add the connector.

This topic includes example values. You might use different values, depending on your business requirements.

**Define the Source System**

Define the source system:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Trading Community Source Systems.
4. On the Manage Trading Community Source Systems page, click **Actions**, and then click **Create**.
5. On the Create Source System page, set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Enter any text that Order Management can use as an abbreviation for this external system. Order Management uses this code to identify this external system throughout the user interface, such as in lists and logs. For example, assume you work for a company named Vision Corporation, and that your deployment must integrate with a legacy order capture system named Vision Capture. You can enter VCAP.</td>
</tr>
</tbody>
</table>
Implementing Order Management

Chapter 3

Integrating Order Management Cloud

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter text that describes the source system, such as Vision Capture.</td>
</tr>
<tr>
<td>Type</td>
<td>Choose Spoke.</td>
</tr>
<tr>
<td>Enable for Order Orchestration and Planning</td>
<td>Contains a check mark.</td>
</tr>
</tbody>
</table>

6. Click **Save and Close**, and then click **Done**.
7. On the Search page, search for Manage Upstream and Fulfillment Source Systems, and then open it.
8. On the Manage Orchestration Source Systems page, click **Actions**, and then click **Create**.
9. In the Create Source System dialog, set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Choose the code that you created in step 4, such as VCAP.</td>
</tr>
<tr>
<td>Time Zone</td>
<td>Choose the time zone where the server is located.</td>
</tr>
<tr>
<td>Version</td>
<td>Choose Other.</td>
</tr>
<tr>
<td>Order Orchestration Type</td>
<td>Specify how your deployment will use this external system. For example, choose <strong>Order capture</strong>.</td>
</tr>
<tr>
<td>Collections Allowed</td>
<td>Contains a check mark.</td>
</tr>
<tr>
<td>Enable Data Cross-Reference</td>
<td>Contains a check mark.</td>
</tr>
</tbody>
</table>

10. Click **Save and Close**, and then click **Done**.

Add the Connector

Add the connector:

1. On the Search page, search for Manage External Interface Web Service Details, and then open it.
2. On the Manage Connector Details page, click **Actions**, and then click **Add Row**.
3. In the new row, set the following values, and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target System</td>
<td>Choose the code that you created in step 4 on the Create Source System page, such as VCAP.</td>
</tr>
<tr>
<td>Connector Name</td>
<td>Enter text that describes the connector. For example, enter Connector_to_VCAP</td>
</tr>
<tr>
<td>Connector URL</td>
<td>Enter the URL that locates the connector service that resides on the source system. In this example, enter the URL that locates the VCAP system.</td>
</tr>
<tr>
<td>User Name and Password</td>
<td>Enter the values that the Status Update service requires. For example, the user that you specify must be a valid user, and this user must use the privileges that allow this user to run the Status Update service. Order Management uses the credentials that you provide so that it can</td>
</tr>
</tbody>
</table>
4. **Optional.** Allow more than one source system instance to communicate with Order Management:
   - Use Trading Community Architecture to add a value to the Target System drop-down list.
   - Repeat step 3, except set the Target System to the value that you added in Trading Community Architecture. For example, assume you work for a telecommunications company. You add one connection to a system named PER_ORA_BM_CPQ for personal lines, and then add another connection to a system named BUS_ORA_BM_CPQ for business lines. Note that CPQ is an acronym for Configure, Price and Quote. Order Management will deliver status notifications and billing notifications to any system that contains the string ORA_BM_CPQ. You can add a prefix, a suffix, a prefix and a suffix, or no prefix or suffix to this string. For example, you can use ABC_ORA_BM_CPQ_XYZ.

5. Verify that Order Management is connected to the source system, and that it is communicating sales order data:
   - Use a page in the Order Management work area to verify that it updated an order status. For example, verify that it modified the status from Scheduled to Shipped.
   - Log into your source order system, and then verify that it displays the updated status value of the order that you examined in Order Management. For example, if Order Management modified the status from Scheduled to Shipped on the fulfillment line, then verify that your source system also displays Shipped for this value.

If Order Management cannot connect to your source system, then it might display an error message that indicates that it cannot connect. If this situation occurs, then see Troubleshooting Connection Problems With Source Systems: Procedure.

**Add a Test User**

As an option, you can create a test user in your test environment to make sure your deployment can authenticate the message that Order Management sends.

Add a test user:

1. Log into the Administration Console of the Oracle WebLogic Server.
2. In the Administration Console, click Create Users.
3. In the Create a New User area, set the following values, and then click OK.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The value you enter must match the name you used when you created the user credential key earlier in this topic.</td>
</tr>
<tr>
<td>Description</td>
<td>User name and password to use when sending a message to the test client.</td>
</tr>
<tr>
<td>Provider</td>
<td>DefaultAuthentication</td>
</tr>
<tr>
<td>Password</td>
<td>The value you enter must match the name you used when you created the user credential key earlier in this topic.</td>
</tr>
</tbody>
</table>

4. Test the client in your source system to make sure it can send and receive messages to and from Order Management.
Troubleshooting Connection Problems with Source Systems: Procedure

You can troubleshoot an error message that indicates that Order Management Cloud cannot connect to a source system, such as Oracle Configure, Price, and Quote Cloud.

Troubleshoot a connection problem with a source system:

1. In the Navigator, click **Order Management**.
2. On the Overview page, click **Tasks**, and then click **Manage Order Orchestration Messages**.
3. On the Manage Order Orchestration Messages page, in the **Order Orchestration Function** field, select **Send Event Notification**, and then click **Search**.
4. Examine the search results:
   - Make sure the Search Results includes an entry that includes the same URL that you specified in the Connector URL field of the Manage Connector Details page. For details about this field, see Connecting Order Management to Source Systems: Procedure.
   - Make sure this URL correctly identifies the connector service that resides on the source system. To do this, log into your source system, and then examine the connector services that are running.
   - Wait a minute for the log to refresh, requery the Order Orchestration Messages page, and then examine the search results again to determine whether or not the connection successfully restarted. A network error or some other problem might cause the connection to momentarily fail. If the connection restarted successfully, then the list will include details about the events that are associated with the connector URL.

Mapping Names and Codes from Order Capture Systems: Explained

If your deployment includes an external order capture system, then you must map names and codes from this system into Order Management Cloud.

An orchestration reference object is an object that resides in the set of objects that an orchestration process processes so that it can determine the meaning of a name or the description of a code, such as a payment term name, freight code, or transport code. An order capture system typically sends sales order data that contains names or codes to an orchestration process, and this orchestration process must display a meaning for the name or a description for the code. You must collect the data that determines these meanings and descriptions in the Order Orchestration and Planning Data repository.

For example, assume an external order capture system sends sales order data that includes a payment term of 2/10, Net 30 to an orchestration process, and the Order Orchestration and Planning Data repository contains a record that includes a payment term of 2/10, Net 30. In this situation, the orchestration process uses the matching codes to identify the following payment term description:

2% discount earned if paid within 10 days

To get the complete list of orchestration reference objects, you can examine the collected data for them, and view the list of values for the Lookup Type field.
Integrating Order Management with Downstream Fulfillment Systems

Connecting Order Management to Fulfillment Systems: Procedure

You can use a web service to allow Order Management Cloud to communicate with a fulfillment system through the external interface layer. You can use a predefined web service or create a new one.

You must create and register a connector for each fulfillment system. For details about how Order Management routes requests to a fulfillment system, see Architectural Layers in Order Management: How They Work, and search for Doc ID 2123078.1 (Integrating Non-Fusion Fulfillment System With Oracle Fusion Order Management Cloud) on My Oracle Support at https://support.oracle.com.

Connect Order Management to fulfillment systems:

1. Use an integrated development environment of your choice to create a transformation style sheet that can handle the web service payload.
2. Create and deploy the connector:
   - Create an XSLT transformation file that transforms the Common Order Service Data Object (SDO) to the payload of the web service where Order Management must connect. You can use an integrated development environment of your choice or Integration Cloud Services from Oracle to create this file.
   - Create an XSLT transformation file that transforms the reply from the web service to an Order Management message.
   - 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 that you created for the connector.
3. Create a user credential key, and add it to the identity store on the server that will call the web service. For details, see Creating User Credential Keys: Procedure.
4. Get the user credentials that the service provider requires when calling their web service. You can typically get these details from the service provider.
5. Send a request to the IT administrator to add the user credentials that you identified in step 4. Request to add them to the server that will call the web service. Use the CSF-KEY (Credential Store Framework) reference.
6. In the Navigator, click Setup and Maintenance.
7. On the Setup page, click Order Management.
8. Search for, and then open Manage External Interface Web Service Details.
9. On the Manage Connector Details page, click Actions, click Add Row, and then register the service that Order Management must interact with. The external system hosts this service. Set the following values:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target System</td>
<td>Choose a target system.</td>
</tr>
<tr>
<td>Connector Name</td>
<td>Enter the name of the web service.</td>
</tr>
<tr>
<td>Connector URL</td>
<td>Enter the URL that locates the web service that resides on the fulfillment system.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>User Name and Password</td>
<td>Enter the user name and password that the service requires. As an option, you can use the CSF-KEY that references the user credential that Order Management must use to interact with the external web service. There is no requirement to match the actual name of the connector, so you can provide a short name. You can then use this short name in a routing rule. The CSF-KEY applies to all services that the external system provides.</td>
</tr>
</tbody>
</table>

Keystore Recipient Alias

For each server that runs an external web service that Order Management calls, it is recommended that you configure this server to advertise the security certificate in the WSDL. The Oracle WebLogic Server advertises the security certificates, by default. If your servers support this advertisement, then enable it.

If you configure the server to advertise the security certificate, then it is recommended you use the keystore recipient alias. You must do the following work:

- Ask the service provider for the security certificate.
- Make sure an IT administrator imports the target server security certificate into the calling server and provides the keystore recipient alias.
- Add this alias to the external service entry you created when you specified the user credential.
- Add this alias to the Keystore Recipient Alias field on the Manage Connector Details page. This key applies to all services that the target system provides.

If these options do not work, then configure the servers to use the Oracle security certificate, and then import this certificate into your servers. No setup is required on the calling server for the security certificate.

Invocation Mode

Set to one of the following values to determine how Order Management calls the connector when the orchestration process requires an interface to a system that resides outside of Order Management:

- **Business Event.** Raise a business event so that Order Management can interact with a fulfillment system. If you use Integration Cloud Service to integrate Order Management with the fulfillment system, then you must choose Business Event. For details, see Using Integration Cloud Service to Integrate Order Management with Other Systems: Overview.

- **Synchronous Service.** Make a synchronous call to the web service. Requires Order Management to wait for the response from the web service before it continues processing. Use synchronous when the work that Order Management is performing depends on the response. For example, use synchronous when calling credit check because Order Management must wait for credit check to finish and reply with a credit check succeeded status before it can send the sales order to order fulfillment.

- **Asynchronous Service.** Make an asynchronous call to the web service. Allows Order Management to continue other processing while it waits for the response from the web service. Use asynchronous when the work that Order Management is performing does not depend on the response. An asynchronous call is useful in an environment where a service, such as a loan processor, can take a long time to process a client request. For example, scheduling an appointment to install an item that includes a computer network might cause a delay but it does not affect order processing.

Note the following points about the Business Event Invocation mode:

- To support trade compliance screening during order submit, you must use a business event to integrate with Oracle Global Trade Management.
- Business Event supports integration only with Oracle Global Trade Management and Oracle Transportation Management.
Attribute | Description
--- | ---
| Use Business Event to integrate with trade compliance only for compliance screening that occurs when the Order Entry Specialist submits the sales order, and not during order fulfillment.

Send Attachments | Choose Yes to allow Order Management to send the attachments that you add to a sales order.

If you use the web service OrderInformationService to connect with your fulfillment system, then you can use the Send Attachments attribute in the input payload of the GetOrderDetails operation of this web service to allow Order Management to send attachments. For details about using web services, search for Doc ID 2051640.1 (Using Web Services with Oracle Fusion Order Management Cloud) on My Oracle Support at https://support.oracle.com.

10. Repeat step 8 until you register all services that Order Management must interact with.
11. Create a routing rule that selects the web services. For details see Routing Requests to Fulfillment Systems: Explained.

**Related Topics**
- Order-to-Cash with Order Capture Systems: How it Works
- Architectural Layers in Order Management: How They Work

**Routing Requests to Fulfillment Systems: Explained**

A external interface routing rule specifies how to route a fulfillment request to a fulfillment system. You can specify each rule so it selects the fulfillment system connector according to sales order, fulfillment line, or orchestration process attribute. At run time, the rule calls the connector service that translates the payload into the structure that the fulfillment system uses.

The following examples describe ways that you might use an external interface routing rule.

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
</table>
| Route a sales order according to the value of an orchestration process attribute. | Route each sales order that is ready to ship to a fulfillment system according to the task type. You can write the following external interface routing rule to meet this requirement:
- If the type code is Shipment, then route the sales order to the ABCShippingSystem connector.

Route sales order according to the value of a customer attribute. | Assume your company uses two invoicing systems, and that system ABC sends invoices to a customer named Widget Company. You can write the following external interface routing rule to meet this requirement:
- If the product type is Goods, and if the task type is Invoice, then route the request to the ABCInvoicingSystem connector.

**Related Topics**
- Architectural Layers in Order Management: How They Work
Routing Requests to Fulfillment Systems: Procedure

This topic describes how to create an external interface routing rule that routes a request to a fulfillment system. Assume you must create a routing rule that implements the following condition:

- If the task type is Shipment, and if the quantity is 1,000 or more, then route the shipment request to the Big Shipments Warehouse.

You will create the following rule.

![Routing Rule Diagram]

**Description**: Rule that routes requests for large quantity shipments.

**Effective Date**: Always

**Priority**: Medium

**Active**: Yes

**Advanced Mode**: Yes

**Tree Mode**: Yes

**Root**: Order Header

**IF**

- **header**: is a Order Header
- **header.Task Type**: is "Shipment"
- **line**: is a header/Order Fulfill Line
- **line.Ordered Quantity**: is "1000"

**THEN**

- **call**: Set Connector Name ("Big Shipments Warehouse")
Summary of the Work

Create an external interface routing rule:

1. Create the If statement.
2. Create the Then statement.

This topic describes how to declare variables, use Advanced Mode, use Tree Mode, use the Condition Browser, and so on. It is strongly recommended that you make sure you are familiar with these concepts. For details about them and how to use them, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, download the attachment that describes how to use business rules, and then see the chapter that describes how to use Advanced Mode and Tree Mode.

For details about how to use a simplified rule builder, see Using Visual Information Builder to Create Rules: Explained.

This topic includes example values. You might use different values, depending on your business requirements.

Create the If Statement

You will create the following If statement.

```
IF
```

Create the If statement:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup page, click Order Management.
3. Search for, and then open Manage External Interface Routing Rules.
4. On the Manage External Interface Routing Rules page, next to the View list, click Add, click Add Rule, click Expand, click Show Advanced Settings, and then set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Route Shipping Request</td>
</tr>
<tr>
<td>Description</td>
<td>Rule that routes invoice requests for item 2TX to the Big Server fulfillment system.</td>
</tr>
<tr>
<td>Active</td>
<td>Contains a check mark</td>
</tr>
<tr>
<td>Advanced Mode</td>
<td>Contains a check mark</td>
</tr>
</tbody>
</table>
5. In the If area, set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree Mode</td>
<td>Contains a check mark</td>
</tr>
<tr>
<td>Root</td>
<td>Order Header</td>
</tr>
</tbody>
</table>

6. Click **Add Test**, click **Simple Test**, and then click **Left Value**.

7. In the Condition Browser, expand **header**, click **Task Type**, and then click **OK**.

8. Set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field to the left of Is A</td>
<td>header</td>
</tr>
<tr>
<td>Field to the right of Is A</td>
<td>Order Header</td>
</tr>
</tbody>
</table>

9. Click **Add Pattern**, and then set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Value</td>
<td>“Shipment”</td>
</tr>
</tbody>
</table>

You must include the double quotation marks.

10. Click **Add Test**, click **Simple Test**, and then click **Left Value**.

11. In the Condition Browser, expand **fline**, click **Ordered Quantity**, and then click **OK**.

12. Set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Value</td>
<td>“1000”</td>
</tr>
</tbody>
</table>

You must include the double quotation marks.
Create the Then Statement
You will create the following Then statement:

```
THEN
```

1. In the Then area, click Add Action, and then click Call.
2. Click Select a Target, and then click More.
3. Page down a number of times, click Set Connector Name, and then click OK.
4. Click Edit Properties.
5. In the Arguments dialog, enter the following value, and then click OK.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Name</td>
<td>&quot;Big Shipments Warehouse&quot;</td>
</tr>
</tbody>
</table>

You must include the double quotation marks.

6. Click Save and Close.
7. Define a connector named Big Shipments Warehouse.
   
   For details, see Connecting Order Management to Fulfillment Systems: Procedure.

Related Topics
- Using Visual Information Builder to Create Rules: Explained
- Creating Business Rules: Demo

Routing Requests to Fulfillment Systems When Cross-Reference Data Does Not Exist: Procedure

This topic describes how to create a routing rule that routes a request to a fulfillment system when Order Management Cloud cannot find a cross-reference that identifies this system.

Assume you must implement the following routing rules:

- If the task type on the fulfillment line is Shipment, then route the service request to Shipping Fulfillment System A.
- If Order Management cannot find a cross-reference for Shipping Fulfillment System A in an Oracle Application, then get the cross-reference from your fulfillment system.
You will create the following routing rule.

**Summary of the Work**

Create an external interface routing rule that handles a situation where Order Management cannot find a cross-reference:

1. Create the If statement.
2. Create the Then statement.

Note the following points:

- This topic describes how to create a business rule. It is strongly recommended that you get details about how to create business rules. Search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then download the attachment that describes how to use business rules.

- You must complete fields in the same sequence that each table in the following procedure displays them because the rules editor filters the values you can choose in some fields according to the values you set.

- This topic includes example values. You might use different values, depending on your business requirements.
Create the If Statement

You will create the following If statement:

- If the task type on the fulfillment line is Shipment

```plaintext
IF

header is a Order Header and

"Shipment" equals ignore case header.Task Type
```

Create the If statement:

where:

- **Root: Order Header.** Specifies to display objects at the root level or below the root level while you create the rule. This filtering helps to make sure you correctly maintain the object hierarchy.

- **Shipment Equals Ignore Case header.Task Type.** Specifies to examine the value of the Task Type attribute of the header to determine whether it contains a value of Shipment.

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage External Interface Routing Rules.
4. On the Manage External Interface Routing Rules page, click **Add**, click **Add Rule**, and then set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Shipping Fulfillment System A</td>
</tr>
<tr>
<td></td>
<td>You can enter any value. To simplify managing your setup, it is recommended that you enter the name you used when you added the connector to this fulfillment system on the Manage Connector Details page. For details, see Connecting Order Management to Fulfillment Systems: Procedure.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule that routes each request to a fulfillment system when a cross-reference for this system does not exist.</td>
</tr>
</tbody>
</table>

5. Next to Shipping Fulfillment System A, click **Expand**, click **Show Advanced Settings**, and then set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Mode</td>
<td>Contains a check mark.</td>
</tr>
<tr>
<td>Tree Mode</td>
<td>Contains a check mark.</td>
</tr>
<tr>
<td>Root</td>
<td>Order Header</td>
</tr>
</tbody>
</table>
6. In the If area, set the following values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field to the left of Is A</td>
<td>Enter header.</td>
</tr>
<tr>
<td>Field to the right of Is A</td>
<td>Select <strong>Order Header</strong>.</td>
</tr>
<tr>
<td>Left Value</td>
<td>Enter “Shipment”.</td>
</tr>
<tr>
<td></td>
<td>You must include the double quotation marks.</td>
</tr>
<tr>
<td>Field to the right of Left Value</td>
<td>Equals Ignore Case</td>
</tr>
<tr>
<td>Right Value</td>
<td>In the Condition Browser, expand <strong>header</strong>, click <strong>Task Type</strong>, and then click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>

**Create the Then Statement**

You will create the following Then statement:

**THEN**

- ▶️ ▾️ □️ ▬ ▾️ ▼️

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Result ▶️ ▾️</td>
<td>( resultObj:“Shipping Fulfillment System A”, resultObjKey:”SERVICE_NAME” )</td>
</tr>
<tr>
<td></td>
<td>Result ▶️ ▾️</td>
<td>( resultObj:”Yes”, resultObjKey:”RESOLVE_XREF_FOR_CUSTOMERINFO” )</td>
</tr>
</tbody>
</table>

The first assert specifies the following action:
- **Use Shipping Fulfillment System A to fulfill the item**

It uses the following code:
- **Assert New Result**

where:
- **Assert New Result.** Asserts the Result fact into the DOOExternalInterfaceLayer dictionary. Note that the Manage External Interface Routing Rules page implicitly uses this dictionary, so it is not necessary to explicitly reference it.
- **Shipping Fulfillment System A.** Specifies the name of the shipping system to use.
- **SERVICE_NAME.** Instructs the rule to use a service to ship the item.

The second assert specifies the following action:
- *( resultObj:”Yes”, resultObjKey:”RESOLVE_XREF_FOR_CUSTOMERINFO” )*
where:

- **Assert New Result.** Asserts the Result fact into the DOOExternalInterfaceLayer dictionary.
- **RESOLVE_XREF_FOR_CUSTOMERINFO.** Specifies the service to use to resolve the cross-reference.
- **resultObjKey and resultObj.** Are each a property of the Result fact. In this example, resultObj specifies the value to send to the object that resultObjKey identifies.

The value that you set for RESOLVE_XREF_FOR_CUSTOMERINFO instructs the cross-reference service to do one of the following:

- **Yes.** Use a cross-reference from an Oracle Application.
- **No.** Use a cross-reference from the fulfillment system.

If Order Management can find a cross-reference for the fulfillment system in an Oracle Application, then your rule must set the value of RESOLVE_XREF_FOR_CUSTOMERINFO to Yes.

Create the Then statement:

1. In the Then area, click **Add Action**, and then click **Assert New**.
2. Click **Select a Target**, and then click **Result**.
3. Click **Edit Properties**.
4. In the Edit Properties dialog, enter the following values, and then click **OK**.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultObj</td>
<td>“Shipping Fulfillment System A”</td>
</tr>
<tr>
<td></td>
<td>You must include the double quotation marks.</td>
</tr>
<tr>
<td>ResultObjKey</td>
<td>“SERVICE_NAME”</td>
</tr>
<tr>
<td></td>
<td>You must include the double quotation marks.</td>
</tr>
</tbody>
</table>

5. Click **Add Action**, and then click **Assert New**.
6. Click **Select a Target**, and then click **Result**.
7. Click **Edit Properties**.
8. In the Edit Properties dialog, enter the following values, and then click **OK**.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultObj</td>
<td>“Yes”</td>
</tr>
<tr>
<td></td>
<td>You must include the double quotation marks.</td>
</tr>
<tr>
<td>ResultObjKey</td>
<td>“RESOLVE_XREF_FOR_CUSTOMERINFO”</td>
</tr>
<tr>
<td></td>
<td>You must include the double quotation marks.</td>
</tr>
</tbody>
</table>

9. Click **Save and Close**.

**Related Topics**

- Cross-Referencing in Order Management: Explained
Using Integration Cloud Service to Integrate Order Management with External Systems

Using Integration Cloud Service to Integrate Order Management with Other Systems: Overview

You can use Oracle Integration Cloud Service to integrate Order Management Cloud with some other Oracle or non-Oracle application. Oracle provides Integration Cloud Service as an integration platform in the cloud to help you maximize the value of your investments in SaaS (Software as a Service) and on-premise applications.

You can leverage the convenience and ease of use that Integration Cloud Service provides to create integrated business processes that operate across multiple application systems in the cloud or on premise, without having to rely on dedicated information technology resources. You can use Integration Cloud Service to integrate Oracle ERP with your commercial applications and software solutions.

You typically require specialized information technology staff to create an integration between applications. Integration Cloud Service simplifies this work so someone who does not have this level of technical expertise can create the flow. A business analyst can create the flow and map the attributes from the Order Information Service to the web service on the channel that your order fulfillment system uses. The analyst can use a WYSIWYG (what you see is what you get) interface to create the mapping.

Communicating with Subscribers

Oracle ERP Cloud adapter uses business events so that you can achieve the following results:

- Send a notification to a subscriber when a condition occurs.
- Use the setup on the Manage Business Event Trigger Points page that already exists. Use this setup as the criteria to raise business events.
- Use one integration for each subscriber system. This subscriber can use a single web service that listens for the conditions through the business event that you specify.
- Allow the subscriber to request data enrichment in the integration.

The following diagram illustrates how you can deploy a single web service on each of your subscriber systems to receive the notification for each condition that triggers a business event. You can use the Subscribe to ICS integration to subscribe each channel system to this business event. Order Management will send a single event notification to Integration Cloud Service, and then Integration Cloud Service broadcasts it to each of the subscribers that you integrate. If you set up the web service
that you deploy on these channel systems to subscribe to this business event, and if they share the same underlying schema, then mapping between the business event output schema and each web service is very similar.

Order Management can send a notification each time one of the following conditions occurs depending on the business event that you specify.

<table>
<thead>
<tr>
<th>Business Event</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Status Updated</td>
<td>Order Management sends a notification each time one of the following conditions occurs:</td>
</tr>
<tr>
<td></td>
<td>• Update order header status.</td>
</tr>
<tr>
<td></td>
<td>• Update fulfillment line status.</td>
</tr>
<tr>
<td></td>
<td>• Close fulfillment line.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sales Order Notification</th>
<th>Order Management sends a notification each time one of the following conditions occurs:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Update order header status.</td>
</tr>
<tr>
<td></td>
<td>• Update fulfillment line status.</td>
</tr>
<tr>
<td></td>
<td>• Split fulfillment line.</td>
</tr>
<tr>
<td></td>
<td>• Close fulfillment line.</td>
</tr>
<tr>
<td></td>
<td>• Apply hold.</td>
</tr>
<tr>
<td></td>
<td>• Change the jeopardy priority.</td>
</tr>
<tr>
<td></td>
<td>• Update a predefined attribute on a sales order.</td>
</tr>
</tbody>
</table>
Specifying the Data to Communicate

The following diagram illustrates how you can configure the Oracle ERP Cloud Endpoint in Integration Cloud Service to publish web services and business events.

Order Information Service

Order Management publishes the Orchestration Order business object and the web service OrderInformationService to the catalog in the Order ERP cloud. You can choose this business object and web service, and then choose one of the following operations.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
</table>
You can use the OrderImportService web service to send new sales orders and to revise existing sales orders. For details about how to specify the data to communicate, see the Creating Connections section and Creating Integrations section of the guide Using Oracle Integration Cloud Service on Oracle Help Center at https://docs.oracle.com/cloud/latest/intcs_gs/ICSUG.

**Order Fulfillment Response Service**

The following diagram illustrates how you can use the Order Fulfillment Response Service to process the response from a fulfillment system.

**Explanation of Callouts**
Order Management uses the following sequence when you use this service:

1. Order Management raises a business event so that it can send a fulfillment request to the fulfillment system.
2. Integration Cloud Service calls the document service of the business event to send the request to the fulfillment system. The output payload of the document service includes the data that the fulfillment system requires to fulfill this request. You can map this output payload to the web service interface that processes the inbound call on the fulfillment system.
3. The fulfillment system receives the fulfillment request from Order Management, and then calls the processAcknowledgement operation of the Order Fulfillment Response Service to acknowledge that it received the fulfillment request.
4. Integration Cloud Service routes the acknowledgment to Order Management.
5. The fulfillment system fulfills the request, and then calls the processFulfillmentResponse operation to communicate the result of the fulfillment request to Order Management.
6. Integration Cloud Service sends the response to Order Management, and then Order Management processes the response.

You can use the following operations of the OrderFulfillmentResponse service.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>processAcknowledgement</td>
<td>The fulfillment system sends an acknowledgment that confirms it received the fulfillment request. processAcknowledgement accepts and processes this acknowledgment.</td>
</tr>
<tr>
<td>processFulfillmentResponse</td>
<td>The fulfillment system calls processFulfillmentResponse to communicate the result of the fulfillment request to Order Management. Note that the service automatically adds the suffix Response to the operation name, which results in a response port name of processFulfillmentResponseResponse. The operation name processFulfillmentResponse indicates that it processes a fulfillment response in reply to an Order Management fulfillment request. Adding the suffix is by design.</td>
</tr>
<tr>
<td>StageFulfillmentResponse</td>
<td>For future use.</td>
</tr>
</tbody>
</table>
Mapping Source to Target

Integration Cloud Service provides a drag and drop interface that you can use to map the source payload to the target payload. The following diagram includes part of the mapping for the response payload of the GetOrderDetails operation of the Order Information Service, to a target web service.

If you map a data element on the source to an element on the target, then this interface adds a green check mark to the source and target. Note the following points:

- The target also displays the name of the data element at the source.
- The mapping applies to one target web service, so you must perform this mapping for each target system.
- If your deployment includes more than one channel system that must call GetOrderDetails, or if it must perform other operations under the Order Information Service, then you must create an integration in Integration Cloud Service, and then map the source payload to the target payload for each channel. However, you only require one connection between the ERP Service Catalog and Integration Cloud Service because it can support an integration under each web service that you publish under the catalog with a WSDL (Web Services Description Language).

For details about how to create a mapping, see the Mapping Data chapter of the guide Oracle Cloud, Using the Oracle Mapper on Oracle Help Center at https://docs.oracle.com/cloud/latest/intcs_gs/OCMAP.
Using Integration Cloud Service to Integrate Order Management with Other Systems: How It Works

An Order Management Cloud implementation that includes Oracle Integration Cloud Service uses the setup data for the business event trigger points so that it can send a notification to a subscriber when the order status changes. The subscriber can also request to get more details about the sales order.
The following diagram illustrates an example flow that uses Integration Cloud Service to integrate Order Management with a channel system. The diagram includes a channel system that resides in the cloud. However, Integration Cloud Service can integrate Order Management with a cloud or noncloud channel system.

**Explanation of Callouts**

Assume Integration Cloud Service uses the following example flow to integrate Order Management with your subscribers. Assume you configure this integration to broadcast the Order Status Updated Event to subscribers when Order Management updates the order status, such as from Processing to Closed:

1. A channel system receives the shipment for a sales order, and then sends a notification to Order Management regarding the successful delivery.
2. Order Management changes the order status from Processing to Closed.

3. Order Management raises a business event. It examines the configuration on the Manage Business Event Trigger Points page and finds that you configured the Order Header Status Update event so that it is Active, and is also configured to raise an event when the order header status updates to Closed.

4. Assume you configured the Oracle ERP Cloud Adapter to do the following:
   - Use the OrderInformationService to perform the GetOrderDetails operation.
   - Map order attributes in the GetOrderDetailsResponse source payload to the InvokeOrderInformationServiceResponse target payload.
   - Listen for the Order Status Updated events that are occurring on https://my_server.com:9999/fndAppCoreServices/ServiceCatalogService?wsdl.

   The Oracle ERP Cloud Adapter recognizes the event, and then uses the configuration that you specified for the Oracle ERP Cloud Endpoint to determine the service and operation to use.

5. The Oracle ERP Cloud Adapter reads the source to target mapping that you created to determine how to map the source payload to the target payload.

6. Integration Cloud Service sends a notification to the subscriber at the channel system.

Using Integration Cloud Service with Order Management to Communicate Events: Procedure

To use Oracle Integration Cloud Service with Order Management Cloud to communicate business events, you manage trigger points, create a connection, create an integration, track business events, and then test your work.

Summary of the Work

Use Integration Cloud Service with Order Management to communicate events:

1. Manage business event trigger points.
2. Create a connection between Integration Cloud Service and Order Management.
3. Create an integration that monitors business events.
4. Track business events.
5. Test your work.

This topic includes example values. You might use different values, depending on your business requirements.

Manage Business Event Trigger Points

Manage business event trigger points:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup page, click Order Management.
3. Search for, and then open Manage Business Event Trigger Points.
4. Configure the trigger points so that Order Management raises a business event and sends a notification to each subscriber for each of the trigger points that your integration requires.

   For example, send a notification to each subscriber when the status of the sales order header changes to Closed:

   - On the Manage Business Event Trigger Points page, click the Order Header Status Update row, and then make sure the Active option in this row contains a check mark.
   - In the Details area, in the Closed row, add a check mark to the Raise Event option.
Manage Business Event Trigger Points to Send Status Updates on Fulfillment Lines

Configuring the trigger points so that Order Management sends a notification to each subscriber when the status updates on a fulfillment line requires that you modify the orchestration process definition.

Manage business event trigger points to send status updates on fulfillment lines:

1. On the Manage Business Event Trigger Points page, click the **Fulfillment Line Status Update** row, and then make sure the **Active** option in this row contains a check mark.
2. Click **Save and Close**.
3. On the Setup page, search for, and then open Manage Orchestration Process Definitions.
4. On the Manage Orchestration Process Definitions page, search for the orchestration process that your deployment uses.
   For example, search for OrderFulfillmentGenericProcess.
   Each orchestration process controls the status value for each fulfillment line, so you must modify the orchestration process that controls the status value. In this example, you modify the orchestration process that controls the shipping status value.
5. In the Search Results, click the **row** that contains OrderFulfillmentGenericProcess, click **Actions**, and then click **Edit**.
6. In the Process Details area, click **Status Conditions**, click **Fulfillment Line Status Values**, and then click **Edit Status Rule Set**.
7. On the Edit Status Rule Set page, add a check mark to the **Notify External System** option for each Status Value where your deployment must send a notification.
   For example, to send a notification when the fulfillment line status changes to Shipped, add a check mark to the **Notify External System** option in the Shipped row.
8. Repeat step 7 for other status value, as necessary.
   Order management will send a notification when each fulfillment line that references this orchestration process definition reaches the status that you specify in steps 7 and 8.
9. Repeat steps 4 through 8 for each orchestration process definition in your deployment that updates status values, as necessary.

Create a Connection Between Integration Cloud Service and Order Management

Create a connection between Integration Cloud Service and Order Management:

1. Log a service request so the Cloud Operations team can register the CSF-KEY (Credential Store Framework) that allows you to log into Integration Cloud Service. You must request a separate key for each order administrator who must access Integration Cloud Service. For details, see Requesting CSF Keys for Oracle Integration Cloud Service: Procedure.
2. Get access to the SOA Infra Operations Duty role.
   The Order Administrator job role and the Supply Chain Application Administrator job role each come predefined so that they include SOA Infra Operations Duty.
   Note the following points:
   - If you use a different role, such as Order Manager, then use the Security Console to get access to the SOA Infra Operations Duty role.
   - When you create a connection in Integration Cloud Service that you plan to use to publish the Order Status Updated event to Integration Cloud Service, you must specify the login credentials of an Oracle Cloud...
Assign the Order Manager job role so that this same user can call the web services that Order Management uses. Integration Cloud Service requires a single set of Oracle Cloud Application login credentials to define the integrations at design time, and to call the web services at run time. If you use this connection to publish a business event to Integration Cloud Service, and to call Order Management web services, then the Oracle Cloud Application user must use the Order Manager job role and the SOA Infra Operations Duty role.

For details about job roles, see Assigning Job Roles in Order Management: Explained.

3. Log into Integration Cloud Service.
4. On the Welcome page, click Create Connections.
5. On the Connections page, click New Connections.
6. In the Create Connection - Select Adapter dialog, click Oracle ERP Cloud.
7. In the New Connection - Information dialog, set the following values, and then click Create.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Name</td>
<td>Enter any text that helps describe the connection. For example, enter Connection_ for_ Order_ Status_ Update.</td>
</tr>
<tr>
<td>Identifier</td>
<td>Enter any text that helps describe the connection. For example, enter CONNECTION_ FOR_ ORDER_ STATUS_ UPDATE.</td>
</tr>
<tr>
<td>Connection Role</td>
<td>Select Trigger and Invoke.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter any text that helps describe the connection. For example, enter Connection for the order status update.</td>
</tr>
</tbody>
</table>

8. On the page that displays, click Configure Connectivity.
9. In the dialog that displays, set the following values, and then click OK.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERP Services Catalog WSDL URL</td>
<td>Enter the URL that locates the WSDL. Use the following format:</td>
</tr>
<tr>
<td></td>
<td><a href="https://server">https://server</a>: port/fndAppCoreServices/ServiceCatalogServiceWSDL</td>
</tr>
<tr>
<td></td>
<td>For example: https://my_server.com:9999/fndAppCoreServices/ServiceCatalogServiceWSDL</td>
</tr>
<tr>
<td>ERP Events Catalog URL (optional)</td>
<td>Enter the URL that locates the events catalog. Use the following format:</td>
</tr>
<tr>
<td></td>
<td><a href="http://server">http://server</a>: port/soa-infra</td>
</tr>
<tr>
<td></td>
<td>For example: <a href="http://myserver.com:7818/soa-infra">http://myserver.com:7818/soa-infra</a></td>
</tr>
</tbody>
</table>

These URLs allow Integration Cloud Service to connect to Oracle ERP so that Integration Cloud Service can get details about the services and events that are available in Oracle ERP Cloud. It is recommended that you contact your system administrator to determine the URLs that you must use.

10. Click Configure Security.
11. In the Credentials dialog, enter the user name and password that you use to access Order Management as an administrator, and then click OK.
12. At the top of the page, click Test, and then wait for the indicator that displays immediately to the right of Test to change to a green color, and to display 100%.
13. Click Save, and then click Exit Connection.

Create an Integration That Monitors Business Events

In this section, you create an integration that resembles the following setup:

Create an integration that monitors business events:

1. On the Connections page, Click Integrations.
2. On the Integrations page, Click New Integration.
3. In the Create Integration - Select dialog, click Basic Publish to ICS.
   This procedure provides a partial integration example. In an actual integration, it is more likely you will choose Map My Data. To create a full, end-to-end configuration, see the Integration Cloud Service documentation.
4. In the Create New Integration dialog, set the following values, and then click Create.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What do you want to call your integration?</td>
<td>Enter any text that helps describe the connection. For example, enter Connection_for_Order_Status_Update.</td>
</tr>
<tr>
<td>Identifier</td>
<td>Enter any text that helps describe the connection. For example, enter CONNECTION_FOR_ORDER_STATUS_UPDATE.</td>
</tr>
<tr>
<td>Version</td>
<td>Accept the value that displays.</td>
</tr>
<tr>
<td>What does your integration do?</td>
<td>Enter any text that helps describe the connection. For example, enter Connection for the order status update.</td>
</tr>
<tr>
<td>Which package does your integration belong to?</td>
<td>Leave empty.</td>
</tr>
</tbody>
</table>

5. On the page that displays, you will identify the source of the connection that provides the details. For this example, in the search window, enter Connection_for_Order_Status_Update, which is the integration that you created earlier, and then click ENTER.
6. Drag, and then drop Connection_for_Order_Status_Update from the search results onto the Drag and Drop a Trigger area.
7. In the Oracle ERP Cloud Endpoint Configuration Wizard, you specify the events and scenarios that reside in the ERP Cloud connection. Set the following values, and then click Next.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What do you want to call your endpoint?</td>
<td>Enter any text that helps describe the endpoint. For example, enter PublishOrderStatusUpdated.</td>
</tr>
<tr>
<td>What does this endpoint do?</td>
<td>Enter any text that helps describe the connection. For example, enter Publishes the OrderStatusUpdated event.</td>
</tr>
</tbody>
</table>

8. Choose the With Business Events option.

Integration Cloud Service uses the URLs that you defined earlier to get the event catalog, and then display it in the Business Event for Subscription list. This list will help you to choose from the events that are available in Oracle ERP Cloud.

9. In the Business Event for Subscription list, enter order, click Order Status Updated, and then click Next.
10. On the page named Configure the Response to Send to the Oracle ERP Cloud Application, note that Integration Cloud Service is only listening for events in this configuration, and then sends them to subscribers without replying to Order Management. So, click None, and then click Next.
11. On the Summary page, click Done.

**Track Business Events**

In this section, you set up Integration Cloud Service to track business events so that you test, and then monitor your configuration. It is recommended that you track at least one business event so that you can monitor your deployment during normal operations.

Track business events:

1. On the page that displays, click Tracking.

   You can use Tracking for testing purposes. It allows you to view the business events that your integration raises in the monitoring system that Integration Cloud Service provides.

2. For this example, on the Business Identifiers for Tracking page, in the Source tree, under the result branch, click Load More.
You will choose the following business identifiers to track:

3. Drag, and then drop SourceTransactionNumber from the hierarchy tree to the first row of the Tracking Field column.
4. Drag, and then drop Event from the hierarchy tree to the second row of the Tracking Field column.
5. In the result branch, expand LineStatusUpdate, and then click Load More.
6. Drag, and then drop StatusCodeNewValue from the hierarchy tree to the third row of the Tracking Field column.
7. Click Done.
8. On the page that displays, click Save, and then click Exit Integration.
10. In the Confirmation dialog, add a check mark to the Enable Tracing option, and then click Yes.

Test Your Work

Test your work:

1. Log into Order Management, create a sales order, add one order line, use the Shipment Details tab to set the Requested Date to a time that occurs in the future, and then click Submit.
2. In the Confirmation dialog, make a note of the order number that displays, such as 492634, and then click OK.
   The shipment step of an orchestration process runs, sets the Status to scheduled, and then displays this status on the fulfillment line in the Order Lines tab. This shipment step raises the business event that you specified earlier. Note that you must have already configured this orchestration process and finished other setup work in Order Management.
3. Navigate back to Integration Cloud Service, and then click Monitoring.
4. On the Integrations Dashboard, click Tracking, and then click anywhere in the Messages area.
5. In the search window, enter the order number that you noted earlier, such as 492634, and then click ENTER.

6. In the search results, notice that Order Management raised multiple events for the sales order. Click the event at the top of the list, which is the most recent event.

7. On the page that displays, click Business Identifiers, and then verify that the tracking information you set up for tracking earlier displays.

In this example, you set up tracking for the following items:

- **SourceTransactionNumber.** Identifies the order number.
- **Event.** Identifies the business event that gets triggered, which is FulfillmentLineStatusUpdate.
- **StatusCodeNewValue.** Identifies the new status value of the fulfillment line, which is Scheduled.

For example:

![Business Identifiers](image)

**Business Identifiers**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Transaction Number</td>
<td>492634</td>
</tr>
<tr>
<td>Event Code</td>
<td>FulfillmentLineStatusUpdate</td>
</tr>
<tr>
<td>Status Code New Value</td>
<td>SCHEDULED</td>
</tr>
</tbody>
</table>

**Related Topics**

- Security Console: Overview

**Filtering Details from Integration Cloud Service: Procedure**

The setup that you perform on the Manage Business Event Trigger Points page is independent of the systems that monitor these business event notifications. If you must control the event notification that your channel or fulfillment system receives at a finer level, then you can use the filter capability in Oracle Integration Cloud Service to filter this data.

Filter details from the Integration Cloud Service:

1. Use the Manage Business Event Trigger Points page to set up the trigger points that enable the Order Status Updated event.

2. Create the integration in Integration Cloud Service.

For details, see the following resources on the Oracle Help Center:
3. Log into Integration Cloud Service and add the filters:
   - Open the mapping that maps the integration to Order Management Cloud.
   - To allow events that only originate for sales orders from Order Management, create a filter on the Source Transaction System attribute.
   - To ignore events for a trigger that you define on the Manage Business Event Trigger Points page, create a filter on the Event Code attribute for each trigger that you must filter.

For details, see the following topics in the Using Oracle Integration Cloud Service section on the Oracle Help Center:

<table>
<thead>
<tr>
<th>What</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using Integration Cloud Service with Order Management to Communicate Events: Procedure</td>
<td><a href="https://docs.oracle.com/cloud/latest/scmcsugs/FAIOM/FAIOM2061947htm#FAIOM2221912">https://docs.oracle.com/cloud/latest/scmcsugs/FAIOM/FAIOM2061947htm#FAIOM2221912</a></td>
</tr>
<tr>
<td>Tutorials for Oracle Integration Cloud Service</td>
<td><a href="http://docs.oracle.com/cloud/latest/intcsgs/intcs">http://docs.oracle.com/cloud/latest/intcsgs/intcs</a> tutorials.htm</td>
</tr>
<tr>
<td>Videos for Oracle Integration Cloud Service</td>
<td><a href="http://docs.oracle.com/cloud/latest/intcsgs/intcs">http://docs.oracle.com/cloud/latest/intcsgs/intcs</a> videos.htm</td>
</tr>
<tr>
<td>Developing Integration Flows with Integration Cloud Service</td>
<td><a href="http://www.oracle.com/webfolder/technetwork/tutorials/obe/cloud/ics/DevelopingIntegrationFlowswithIntegrationCloudServicesOBE/ICSOBHtml">http://www.oracle.com/webfolder/technetwork/tutorials/obe/cloud/ics/DevelopingIntegrationFlowswithIntegrationCloudServicesOBE/ICSOBHtml</a></td>
</tr>
</tbody>
</table>

### Requesting CSF Keys for Integration Cloud Service: Procedure

The CSF-KEY (Credential Store Framework) key allows you to log into Oracle Integration Cloud Service. You must log a service request so the Cloud Operations team can register the CSF-KEY.

Request the CSF-KEY for Oracle Integration Cloud Service:

1. Create a service request that requests Oracle to create the CSF-KEY. Include the following details in your request.
Service Request Section | Description
--- | ---
Subject line | Include the following text:

Body | Provide the identity domain that your integration can use for Order Management Cloud and for Oracle Integration Cloud Service subscriptions. For example, icssvc.identity.domainidm2152.

Request a separate CSF-KEY for each order administrator who must access Integration Cloud Service.

2. Wait for Oracle to reply to your service request. When Oracle does reply, provide them with the user name and password that you use to access Integration Cloud Service. Oracle will then create the CSF-KEY.

Integrating Order Management with Other Oracle Applications

Integrating Order Management with Other Oracle Applications: Overview

You can integrate Order Management Cloud to send details to another Oracle application. For example, from an upstream source system to a downstream billing system. You can use a predefined attribute to send details, or you can create an extensible flexfield in Order Management to store this data, and then map it another Oracle application.

Consider the following example requirements.

<table>
<thead>
<tr>
<th>Oracle Application</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Receivables</td>
<td>Consider the following examples:</td>
</tr>
<tr>
<td></td>
<td>• Map the fulfillment line attribute named Purchase Order Line Number in Order Management (CUSTOMER_PO_LINE_NUMBER) to a descriptive flexfield named Invoice Line Level in Oracle Receivables. Order Management does not come predefined to display this attribute in the user interface.</td>
</tr>
<tr>
<td></td>
<td>• Create an extensible flexfield on the order header or order line in Order Management. Map this flexfield to an interface column or descriptive flexfield in Oracle Receivables.</td>
</tr>
<tr>
<td></td>
<td>• Send a concatenated value that represents the attributes of a covered item, such as Description, Start Date, and End Date, to the TranslatedDescription attribute in Oracle Receivables. Use an interface column in Oracle Receivables to send the data.</td>
</tr>
<tr>
<td>Oracle Procurement</td>
<td>Create an extensible flexfield on the fulfillment line in Order Management that captures the item price. Send the price to the purchase request in Oracle Procurement.</td>
</tr>
<tr>
<td>Oracle Receiving</td>
<td>Create extensible flexfields on the fulfillment line in Order Management that captures lot details and serial details. Send these details to the receiving request in Oracle Receiving.</td>
</tr>
<tr>
<td>Oracle Shipping</td>
<td>Create an extensible flexfield on the fulfillment line in Order Management that captures shipment details. Send these details to a descriptive flexfield in Oracle Shipping.</td>
</tr>
</tbody>
</table>
Note that you set up an integration algorithm and a service mapping when you create the integration. An integration algorithm uses logic that is similar to a pricing algorithm, and the service mapping that you set up is similar to the service mapping that Oracle Pricing uses. So, you use the interface that the Pricing Administration work area provides to set up the integration algorithm and the service mapping that you use for integration. However, an integration algorithm and the service mapping that you set up does not affect pricing.

**Guidelines for Integrating Order Management with Oracle Receivables**

Note the following points:

- You can map an interface line of type Line.
- You cannot map an interface line of type Discount, Charge, or Freight.
- You can use only one context for each descriptive flexfield.

You must use the following attributes for each context entity that you define for the descriptive flexfield:

- NameSpace
- XsiType
- _FLEX_Context

If you use a flexfield, then you must use one of the following suffixes in the name of the flexfield.

<table>
<thead>
<tr>
<th>Flexfield</th>
<th>Suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td>descriptive flexfield</td>
<td>Dff_Custom</td>
</tr>
<tr>
<td>global descriptive flexfield</td>
<td>gdf_Custom</td>
</tr>
<tr>
<td>extensible flexfield</td>
<td>Eff_Custom</td>
</tr>
</tbody>
</table>

To improve readability, it is recommended that you include an underscore ( _ ) immediately before the suffix. For example, myDescriptiveFlexfield_Dff.

If you must map an attribute from the order header, extensible flexfield, or fulfillment line detail, then you must use an integration algorithm. Note that Order Management uses the following hierarchy:

Order header
- Fulfillment line
- Fulfillment line details

However, other Oracle applications might not organize these objects into a hierarchy. Instead, they might represent them in a single object named ARLLine. ARLLine also includes the descriptive flexfields, so it can represent the same data that the fulfillment line represents. If you must copy attributes from the fulfillment line to ARLLine, then you require an integration algorithm only if you must implement conditional logic.


**Entities and Attributes That You Can Use When Integrating Order Management with Other Oracle Applications**

You can use the following entities:

- ARInterfaceLine
You can map the attributes that the ARInterfaceLine entity contains.

**Attributes That You Cannot Use With Oracle Receivables**

You cannot map the following attributes.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountingRuleId</td>
<td>PaymentAttributes</td>
</tr>
<tr>
<td>Amount</td>
<td>PaymentSetId</td>
</tr>
<tr>
<td>ApprovalCode</td>
<td>PaymentTxnExtensionId</td>
</tr>
<tr>
<td>AssessableValue</td>
<td>ProdFcCategId</td>
</tr>
<tr>
<td>AuthorizationNumber</td>
<td>ProductCategory</td>
</tr>
<tr>
<td>BatchSourceName</td>
<td>ProductFiscClassification</td>
</tr>
<tr>
<td>Comments</td>
<td>ProductType</td>
</tr>
<tr>
<td>ContractEndDate</td>
<td>PurchaseOrder</td>
</tr>
<tr>
<td>ContractStartDate</td>
<td>PurchaseOrderDate</td>
</tr>
<tr>
<td>ConversionDate</td>
<td>Quantity</td>
</tr>
<tr>
<td>ConversionRate</td>
<td>QuantityOrdered</td>
</tr>
<tr>
<td>ConversionType</td>
<td>ReasonCode</td>
</tr>
<tr>
<td>CreditMethodForAccountRule</td>
<td>ReceiptMethodId</td>
</tr>
<tr>
<td>CreditMethodForInstallments</td>
<td>ReferenceLineId</td>
</tr>
<tr>
<td>CurrencyCode</td>
<td>SalesOrder</td>
</tr>
<tr>
<td>CustomerTrxTypeSequenceId</td>
<td>SalesOrderDate</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>DefaultTaxationCountry</td>
<td>SalesOrderLine</td>
</tr>
<tr>
<td>DocumentSubType</td>
<td>SalesOrderRevision</td>
</tr>
<tr>
<td>FinalDischargeLocationCode</td>
<td>SalesOrderSource</td>
</tr>
<tr>
<td>FinalDischargeLocationId</td>
<td>SetOfBooksId</td>
</tr>
<tr>
<td>FirstPtyRegId</td>
<td>ShipDateActual</td>
</tr>
<tr>
<td>FirstPtyRegNumber</td>
<td>ShipVia</td>
</tr>
<tr>
<td>FOBPoint</td>
<td>TaxCode</td>
</tr>
<tr>
<td>IntendedUseClassifId</td>
<td>TaxExempt</td>
</tr>
<tr>
<td>TransactionInterfaceLineDff</td>
<td>TaxExemptNumber</td>
</tr>
<tr>
<td>InternalNotes</td>
<td>TaxExemptReasonCode</td>
</tr>
<tr>
<td>InventoryItemid</td>
<td>TaxInvoiceDate</td>
</tr>
<tr>
<td>InvoicingRuleId</td>
<td>TaxInvoiceNumber</td>
</tr>
<tr>
<td>LastPeriodToCredit</td>
<td>Taxable</td>
</tr>
<tr>
<td>LegalEntityId</td>
<td>PaymentTermsId</td>
</tr>
<tr>
<td>LineIntendedUse</td>
<td>ThirdPtyRegId</td>
</tr>
<tr>
<td>LineType</td>
<td>ThirdPtyRegNumber</td>
</tr>
<tr>
<td>OrgId</td>
<td>TrxBusinessCategory</td>
</tr>
<tr>
<td>OrigSystemShipPartyId</td>
<td>UnitSellingPrice</td>
</tr>
<tr>
<td>OrigSystemShipPartySiteId</td>
<td>UnitStandardPrice</td>
</tr>
<tr>
<td>OrigSystemShipPtyContactId</td>
<td>UOMCode</td>
</tr>
<tr>
<td>OrigSystemSoldPartyId</td>
<td>UserDefinedFiscClass</td>
</tr>
<tr>
<td>OrigSystemBatchName</td>
<td>WarehouseCode</td>
</tr>
</tbody>
</table>
### Attributes That You Can Use With Drop Ship

You can map the following attributes.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount</td>
<td>AutosourceFlag</td>
</tr>
<tr>
<td>ApproverEmail</td>
<td>BaseModelNumber</td>
</tr>
<tr>
<td>ApproverId</td>
<td>BaseModelPrice</td>
</tr>
<tr>
<td>AssessableAmount</td>
<td>BuyerEmail</td>
</tr>
<tr>
<td>Attribute1</td>
<td>BuyerId</td>
</tr>
<tr>
<td>Attribute1</td>
<td>BuyerName</td>
</tr>
<tr>
<td>Attribute10</td>
<td>CategoryId</td>
</tr>
<tr>
<td>Attribute10</td>
<td>CategoryName</td>
</tr>
<tr>
<td>Attribute11</td>
<td>ChangeActionReason</td>
</tr>
<tr>
<td>Attribute11</td>
<td>CurrencyCode</td>
</tr>
<tr>
<td>Attribute12</td>
<td>CustomerItemDescription</td>
</tr>
<tr>
<td>Attribute12</td>
<td>DaysEarlyReceiptAllowed</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Attribute13</td>
<td>DeliverToCustomerContactEmail</td>
</tr>
<tr>
<td>Attribute13</td>
<td>DeliverToCustomerContactId</td>
</tr>
<tr>
<td>Attribute14</td>
<td>DeliverToCustomerContactName</td>
</tr>
<tr>
<td>Attribute14</td>
<td>DeliverToCustomerContactNumber</td>
</tr>
<tr>
<td>Attribute15</td>
<td>DeliverToCustomerName</td>
</tr>
<tr>
<td>Attribute15</td>
<td>DeliverToCustomerLocationCLLICode</td>
</tr>
<tr>
<td>Attribute16</td>
<td>DeliverToCustomerLocationOriginalSystemReference</td>
</tr>
<tr>
<td>Attribute16</td>
<td>DeliverToCustomerLocationOriginalSystemReference</td>
</tr>
<tr>
<td>Attribute17</td>
<td>DeliverToCustomerNumber</td>
</tr>
<tr>
<td>Attribute17</td>
<td>DeliverToLocationCode</td>
</tr>
<tr>
<td>Attribute18</td>
<td>DeliverToLocationId</td>
</tr>
<tr>
<td>Attribute18</td>
<td>Description</td>
</tr>
<tr>
<td>Attribute19</td>
<td>Description</td>
</tr>
<tr>
<td>Attribute19</td>
<td>DestinationSubinventory</td>
</tr>
<tr>
<td>Attribute2</td>
<td>DocumentStatusCode</td>
</tr>
<tr>
<td>Attribute2</td>
<td>EmergencyPONumber</td>
</tr>
<tr>
<td>Attribute20</td>
<td>FulfillLineId</td>
</tr>
<tr>
<td>Attribute20</td>
<td>FulfillLineId</td>
</tr>
<tr>
<td>Attribute3</td>
<td>GroupCode</td>
</tr>
<tr>
<td>Attribute3</td>
<td>HazardClass</td>
</tr>
<tr>
<td>Attribute4</td>
<td>HazardClassId</td>
</tr>
<tr>
<td>Attribute4</td>
<td>HeaderId</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Attribute5</td>
<td>InterfaceSourceLineId</td>
</tr>
<tr>
<td>Attribute5</td>
<td>ItemDescription</td>
</tr>
<tr>
<td>Attribute6</td>
<td>ItemRevision</td>
</tr>
<tr>
<td>Attribute6</td>
<td>ItemRevision</td>
</tr>
<tr>
<td>Attribute7</td>
<td>Justification</td>
</tr>
<tr>
<td>Attribute7</td>
<td>LineType</td>
</tr>
<tr>
<td>Attribute8</td>
<td>LineTypeld</td>
</tr>
<tr>
<td>Attribute8</td>
<td>NegotiatedByPreparerFlag</td>
</tr>
<tr>
<td>Attribute9</td>
<td>NegotiationRequiredFlag</td>
</tr>
<tr>
<td>Attribute9</td>
<td>NewSupplierFlag</td>
</tr>
<tr>
<td>AttributeCategory</td>
<td>NoteToBuyer</td>
</tr>
<tr>
<td>AttributeCategory</td>
<td>NoteToReceiver</td>
</tr>
<tr>
<td>AttributeDate1</td>
<td>NoteToSupplier</td>
</tr>
<tr>
<td>AttributeDate1</td>
<td>Price</td>
</tr>
<tr>
<td>AttributeDate10</td>
<td>Price</td>
</tr>
<tr>
<td>AttributeDate10</td>
<td>ProcurementBUId</td>
</tr>
<tr>
<td>AttributeDate2</td>
<td>ProcurementBUName</td>
</tr>
<tr>
<td>AttributeDate2</td>
<td>PurchaseRequestConfig</td>
</tr>
<tr>
<td>AttributeDate3</td>
<td>PurchaseRequestHeader</td>
</tr>
<tr>
<td>AttributeDate3</td>
<td>PurchaseRequestLine</td>
</tr>
<tr>
<td>AttributeDate4</td>
<td>Rate</td>
</tr>
<tr>
<td>AttributeDate4</td>
<td>RateDate</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>AttributeDate5</td>
<td>RateType</td>
</tr>
<tr>
<td>AttributeDate5</td>
<td>RequesterEmail</td>
</tr>
<tr>
<td>AttributeDate6</td>
<td>RequesterId</td>
</tr>
<tr>
<td>AttributeDate6</td>
<td>RequisitionNumber</td>
</tr>
<tr>
<td>AttributeDate7</td>
<td>SecondaryQuantity</td>
</tr>
<tr>
<td>AttributeDate7</td>
<td>SecondaryUOM</td>
</tr>
<tr>
<td>AttributeDate8</td>
<td>SecondaryUOMCode</td>
</tr>
<tr>
<td>AttributeDate8</td>
<td>ShipToCustomerContactEmail</td>
</tr>
<tr>
<td>AttributeDate9</td>
<td>ShipToCustomerLocationOriginalSystemReference</td>
</tr>
<tr>
<td>AttributeDate9</td>
<td>ShipToLocationCode</td>
</tr>
<tr>
<td>AttributeNumber1</td>
<td>ShipToLocationId</td>
</tr>
<tr>
<td>AttributeNumber1</td>
<td>SoldToLegalEntityId</td>
</tr>
<tr>
<td>AttributeNumber10</td>
<td>SoldToLegalEntityName</td>
</tr>
<tr>
<td>AttributeNumber10</td>
<td>SourceAgreementId</td>
</tr>
<tr>
<td>AttributeNumber2</td>
<td>SourceAgreementLineId</td>
</tr>
<tr>
<td>AttributeNumber2</td>
<td>SourceAgreementLineNumber</td>
</tr>
<tr>
<td>AttributeNumber3</td>
<td>SourceAgreementNumber</td>
</tr>
<tr>
<td>AttributeNumber3</td>
<td>SupplierContactEmail</td>
</tr>
<tr>
<td>AttributeNumber4</td>
<td>SupplierContactFax</td>
</tr>
<tr>
<td>AttributeNumber4</td>
<td>SupplierContactId</td>
</tr>
<tr>
<td>AttributeNumber5</td>
<td>SupplierContactName</td>
</tr>
<tr>
<td>AttributeNumber5</td>
<td>SupplierContactPhone</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>AttributeNumber6</td>
<td>SupplierItemNumber</td>
</tr>
<tr>
<td>AttributeNumber6</td>
<td>SupplierItemNumber</td>
</tr>
<tr>
<td>AttributeNumber7</td>
<td>TaxationCountryCode</td>
</tr>
<tr>
<td>AttributeNumber7</td>
<td>TaxationTerritory</td>
</tr>
<tr>
<td>AttributeNumber8</td>
<td>UNNumber</td>
</tr>
<tr>
<td>AttributeNumber8</td>
<td>UNNumberId</td>
</tr>
<tr>
<td>AttributeNumber9</td>
<td>UrgentRequisitionLineFlag</td>
</tr>
<tr>
<td>AttributeTimestamp1</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp1</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp10</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp10</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp2</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp2</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp3</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp3</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp4</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp4</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp5</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp5</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp6</td>
<td></td>
</tr>
<tr>
<td>AttributeTimestamp6</td>
<td></td>
</tr>
</tbody>
</table>
**Attributes That You Cannot Use With Drop Ship**

You cannot map the following attributes.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssessableAmount</td>
<td>ChargeAccountId</td>
</tr>
<tr>
<td>AttachmentPrimaryKey2</td>
<td>ChargeAccountSegment1</td>
</tr>
<tr>
<td>AttachmentPrimaryKey3</td>
<td>ChargeAccountSegment10</td>
</tr>
<tr>
<td>AttachmentPrimaryKey4</td>
<td>ChargeAccountSegment11</td>
</tr>
<tr>
<td>AttachmentPrimaryKey5</td>
<td>ChargeAccountSegment12</td>
</tr>
<tr>
<td>BaseModelNumber</td>
<td>ChargeAccountSegment13</td>
</tr>
<tr>
<td>Carrier</td>
<td>ChargeAccountSegment14</td>
</tr>
<tr>
<td>ComponentLineNumber</td>
<td>ChargeAccountSegment15</td>
</tr>
<tr>
<td>DeliverToOrganizationCode</td>
<td>ChargeAccountSegment16</td>
</tr>
<tr>
<td>DocumentFiscalClassification</td>
<td>ChargeAccountSegment17</td>
</tr>
<tr>
<td>DocumentFiscalClassificationCode</td>
<td>ChargeAccountSegment18</td>
</tr>
<tr>
<td>ItemNumber</td>
<td>ChargeAccountSegment19</td>
</tr>
<tr>
<td>ItemNumber</td>
<td>ChargeAccountSegment2</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>LineIntendedUse</td>
<td>ChargeAccountSegment20</td>
</tr>
<tr>
<td>LineIntendedUseId</td>
<td>ChargeAccountSegment21</td>
</tr>
<tr>
<td>LineIntendedUseName</td>
<td>ChargeAccountSegment22</td>
</tr>
<tr>
<td>LocationOfFinalDischargeCode</td>
<td>ChargeAccountSegment23</td>
</tr>
<tr>
<td>PreparerEmail</td>
<td>ChargeAccountSegment24</td>
</tr>
<tr>
<td>ProductCategoryName</td>
<td>ChargeAccountSegment25</td>
</tr>
<tr>
<td>ProductFiscalClassification</td>
<td>ChargeAccountSegment26</td>
</tr>
<tr>
<td>ProductFiscalClassificationName</td>
<td>ChargeAccountSegment27</td>
</tr>
<tr>
<td>ProductType</td>
<td>ChargeAccountSegment28</td>
</tr>
<tr>
<td>ProductTypeName</td>
<td>ChargeAccountSegment29</td>
</tr>
<tr>
<td>RequisitioningBUName</td>
<td>ChargeAccountSegment3</td>
</tr>
<tr>
<td>ShipToCustomerContactName</td>
<td>ChargeAccountSegment30</td>
</tr>
<tr>
<td>ShipToCustomerContactNumber</td>
<td>ChargeAccountSegment4</td>
</tr>
<tr>
<td>ShipToCustomerName</td>
<td>ChargeAccountSegment5</td>
</tr>
<tr>
<td>ShipToCustomerNumber</td>
<td>ChargeAccountSegment6</td>
</tr>
<tr>
<td>SupplierName</td>
<td>ChargeAccountSegment7</td>
</tr>
<tr>
<td>SupplierSiteName</td>
<td>ChargeAccountSegment8</td>
</tr>
<tr>
<td>TaxClassificationName</td>
<td>ChargeAccountSegment9</td>
</tr>
<tr>
<td>ThirdPartyTaxRegistrationNumber</td>
<td>ComponentLineId</td>
</tr>
<tr>
<td>TransactionBusinessCategory</td>
<td>ConfiguredItemFlag</td>
</tr>
<tr>
<td>TransactionBusinessCategoryName</td>
<td>CustomerItemNumber</td>
</tr>
<tr>
<td>UnitOfMeasure</td>
<td>CustomerPOLineNumber</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>WorkOrderId</td>
<td>CustomerPONumber</td>
</tr>
<tr>
<td>WorkOrderNumber</td>
<td>CustomerPOScheduleNumber</td>
</tr>
<tr>
<td>WorkOrderOperationId</td>
<td>DaysLateReceiptAllowed</td>
</tr>
<tr>
<td>WorkOrderOperationSequence</td>
<td>DeliverToCustomerLocationId</td>
</tr>
<tr>
<td>WorkOrderProduct</td>
<td>DeliverToOrganizationId</td>
</tr>
<tr>
<td>WorkOrderSubTypeCode</td>
<td>DestinationTypeCode</td>
</tr>
<tr>
<td>AttachmentCategory</td>
<td>DistributionAmount</td>
</tr>
<tr>
<td>AttachmentCategory</td>
<td>DistributionQuantity</td>
</tr>
<tr>
<td>AttachmentCategoryItem</td>
<td>errorProcessingLevel</td>
</tr>
<tr>
<td>AttachmentCategoryShip</td>
<td>ExternallyManagedFlag</td>
</tr>
<tr>
<td>AttachmentEntityName</td>
<td>FirstPartyTaxRegistrationId</td>
</tr>
<tr>
<td>AttachmentEntityName</td>
<td>FirstPartyTaxRegistrationNumber</td>
</tr>
<tr>
<td>AttachmentEntityName</td>
<td>groupBy</td>
</tr>
<tr>
<td>AttachmentPrimaryKey1</td>
<td>initiateApprovalAfterRequisitionImport</td>
</tr>
<tr>
<td>AttachmentPrimaryKey1</td>
<td>InterfaceConfigurationComponentKey</td>
</tr>
<tr>
<td>AttachmentPrimaryKey1</td>
<td>InterfaceDistributionKey</td>
</tr>
<tr>
<td>AttachmentPrimaryKey2</td>
<td>InterfaceHeaderKey</td>
</tr>
<tr>
<td>AttachmentPrimaryKey2</td>
<td>InterfaceLineKey</td>
</tr>
<tr>
<td>AttachmentPrimaryKey3</td>
<td>interfaceSourceCode</td>
</tr>
<tr>
<td>AttachmentPrimaryKey3</td>
<td>ItemId</td>
</tr>
<tr>
<td>AttachmentPrimaryKey4</td>
<td>ItemId</td>
</tr>
<tr>
<td>AttachmentPrimaryKey4</td>
<td>ItemType</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>AttachmentPrimaryKey5</td>
<td>LocationOfFinalDischargeId</td>
</tr>
<tr>
<td>AttachmentPrimaryKey5</td>
<td>maximumBatchSize</td>
</tr>
<tr>
<td>Attribute1</td>
<td>ModeOfTransportCode</td>
</tr>
<tr>
<td>Attribute10</td>
<td>nextRequisitionNumber</td>
</tr>
<tr>
<td>Attribute11</td>
<td>OrchestrationCode</td>
</tr>
<tr>
<td>Attribute12</td>
<td>OverReceiptTolerancePercent</td>
</tr>
<tr>
<td>Attribute13</td>
<td>ParentComponentLineId</td>
</tr>
<tr>
<td>Attribute14</td>
<td>Percent</td>
</tr>
<tr>
<td>Attribute15</td>
<td>PreparerId</td>
</tr>
<tr>
<td>Attribute16</td>
<td>ProductCategory</td>
</tr>
<tr>
<td>Attribute17</td>
<td>ProductFiscalClassificationId</td>
</tr>
<tr>
<td>Attribute18</td>
<td>ProjectBillableFlag</td>
</tr>
<tr>
<td>Attribute19</td>
<td>ProjectCapitalizableFlag</td>
</tr>
<tr>
<td>Attribute2</td>
<td>ProjectContextCategory</td>
</tr>
<tr>
<td>Attribute20</td>
<td>ProjectContractId</td>
</tr>
<tr>
<td>Attribute3</td>
<td>ProjectContractLineId</td>
</tr>
<tr>
<td>Attribute4</td>
<td>ProjectContractName</td>
</tr>
<tr>
<td>Attribute5</td>
<td>ProjectContractNumber</td>
</tr>
<tr>
<td>Attribute6</td>
<td>ProjectExpenditureItemDate</td>
</tr>
<tr>
<td>Attribute7</td>
<td>ProjectExpenditureTypeId</td>
</tr>
<tr>
<td>Attribute8</td>
<td>ProjectExpenditureTypeName</td>
</tr>
<tr>
<td>Attribute9</td>
<td>ProjectFundingAllocationId</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>AttributeCategory</td>
<td>ProjectFundingSourceName</td>
</tr>
<tr>
<td>AttributeDate1</td>
<td>ProjectId</td>
</tr>
<tr>
<td>AttributeDate10</td>
<td>ProjectName</td>
</tr>
<tr>
<td>AttributeDate2</td>
<td>ProjectNumber</td>
</tr>
<tr>
<td>AttributeDate3</td>
<td>ProjectOrganizationId</td>
</tr>
<tr>
<td>AttributeDate4</td>
<td>ProjectOrganizationName</td>
</tr>
<tr>
<td>AttributeDate5</td>
<td>ProjectReservedAttribute1</td>
</tr>
<tr>
<td>AttributeDate6</td>
<td>ProjectTaskId</td>
</tr>
<tr>
<td>AttributeDate7</td>
<td>ProjectTaskName</td>
</tr>
<tr>
<td>AttributeDate8</td>
<td>ProjectUserDefinedAttribute1</td>
</tr>
<tr>
<td>AttributeDate9</td>
<td>ProjectUserDefinedAttribute10</td>
</tr>
<tr>
<td>AttributeNumber1</td>
<td>ProjectUserDefinedAttribute2</td>
</tr>
<tr>
<td>AttributeNumber10</td>
<td>ProjectUserDefinedAttribute3</td>
</tr>
<tr>
<td>AttributeNumber2</td>
<td>ProjectUserDefinedAttribute4</td>
</tr>
<tr>
<td>AttributeNumber3</td>
<td>ProjectUserDefinedAttribute5</td>
</tr>
<tr>
<td>AttributeNumber4</td>
<td>ProjectUserDefinedAttribute6</td>
</tr>
<tr>
<td>AttributeNumber5</td>
<td>ProjectUserDefinedAttribute7</td>
</tr>
<tr>
<td>AttributeNumber6</td>
<td>ProjectUserDefinedAttribute8</td>
</tr>
<tr>
<td>AttributeNumber7</td>
<td>ProjectUserDefinedAttribute9</td>
</tr>
<tr>
<td>AttributeNumber8</td>
<td>ProjectWorkTypeld</td>
</tr>
<tr>
<td>AttributeNumber9</td>
<td>ProjectWorkTypeName</td>
</tr>
<tr>
<td>AttributeTimestamp1</td>
<td>PurchaseRequestInputReqDistInterface</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>AttributeTimestamp10</td>
<td>Quantity</td>
</tr>
<tr>
<td>AttributeTimestamp2</td>
<td>Quantity</td>
</tr>
<tr>
<td>AttributeTimestamp3</td>
<td>RequestedDeliveryDate</td>
</tr>
<tr>
<td>AttributeTimestamp4</td>
<td>RequestedShipDate</td>
</tr>
<tr>
<td>AttributeTimestamp5</td>
<td>requisitioningBUId</td>
</tr>
<tr>
<td>AttributeTimestamp6</td>
<td>RequisitionBUId</td>
</tr>
<tr>
<td>AttributeTimestamp7</td>
<td>requisitioningBUName</td>
</tr>
<tr>
<td>AttributeTimestamp8</td>
<td>SalesOrderLineNumber</td>
</tr>
<tr>
<td>AttributeTimestamp9</td>
<td>SalesOrderNumber</td>
</tr>
<tr>
<td>BackToBackFlag</td>
<td>SalesOrderScheduleNumber</td>
</tr>
<tr>
<td>BaseModelId</td>
<td>ServiceLevelCode</td>
</tr>
<tr>
<td>BaseModelId</td>
<td>ShipToCustomerContactId</td>
</tr>
<tr>
<td>BudgetDate</td>
<td>ShipToCustomerId</td>
</tr>
<tr>
<td>CarrierId</td>
<td>ShipToCustomerLocationCLLICode</td>
</tr>
<tr>
<td></td>
<td>ShipToCustomerLocationId</td>
</tr>
<tr>
<td>SupplierId</td>
<td></td>
</tr>
<tr>
<td>SupplierSiteId</td>
<td></td>
</tr>
<tr>
<td>TaxClassificationCode</td>
<td></td>
</tr>
<tr>
<td>ThirdPartyTaxRegistrationId</td>
<td></td>
</tr>
<tr>
<td>UnitOfMeasure</td>
<td></td>
</tr>
<tr>
<td>UnitOfMeasureCode</td>
<td></td>
</tr>
<tr>
<td>UnitOfMeasureCode</td>
<td></td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>UserDefinedFiscalClassification</td>
<td>UserDefinedFiscalClassificationName</td>
</tr>
</tbody>
</table>

**Attributes That You Can Use With Oracle Receiving**

You can map the following attributes.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllowSubstituteReceipt</td>
<td>ReceiptAdvice</td>
</tr>
<tr>
<td>Comments</td>
<td>ReceiptAdviceLine</td>
</tr>
<tr>
<td>Comments</td>
<td>ReceiptAdviceLineLot</td>
</tr>
<tr>
<td>CountryOfOriginCode</td>
<td>ReceiptAdviceLineLotWithSerials</td>
</tr>
<tr>
<td>CurrencyConversionDate</td>
<td>ReceiptDaysExceptionCode</td>
</tr>
<tr>
<td>CurrencyConversionRate</td>
<td>RoutingCode</td>
</tr>
<tr>
<td>CurrencyConversionType</td>
<td>RoutingHeaderId</td>
</tr>
<tr>
<td>CustomerSiteId</td>
<td>SecondaryQuantityExpected</td>
</tr>
<tr>
<td>DaysEarlyReceiptAllowed</td>
<td>SecondaryTransactionQuantity</td>
</tr>
<tr>
<td>DaysLateReceiptAllowed</td>
<td>SecondaryTransactionQuantity</td>
</tr>
<tr>
<td>DocumentRevisionDate</td>
<td>SecondaryTransactionQuantityUOMCode</td>
</tr>
<tr>
<td>DocumentRevisionNumber</td>
<td>SecondaryTransactionQuantityUOMCode</td>
</tr>
<tr>
<td>DocumentScheduleNumber</td>
<td>SecondaryUnitOfMeasure</td>
</tr>
<tr>
<td>EnforceShipToLocCode</td>
<td>SecondaryUOMCode</td>
</tr>
<tr>
<td>FreightCarrierId</td>
<td>ShipFromLocationCode</td>
</tr>
<tr>
<td>FreightCarrierName</td>
<td>ShipFromLocationCode</td>
</tr>
<tr>
<td>FreightTerms</td>
<td>ShipFromLocationId</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>FulfillLineId</td>
<td>ShipFromLocationId</td>
</tr>
<tr>
<td>FulfillLineId</td>
<td>ShipToLocationCode</td>
</tr>
<tr>
<td>FulfillLineId</td>
<td>ShipToLocationCode</td>
</tr>
<tr>
<td>FulfillLineId</td>
<td>ShipToLocationId</td>
</tr>
<tr>
<td>GradeCode</td>
<td>ShipToLocationId</td>
</tr>
<tr>
<td>GradeCode</td>
<td>ShipToOrganizationCode</td>
</tr>
<tr>
<td>HeaderId</td>
<td>ShipToOrganizationId</td>
</tr>
<tr>
<td>ItemCategory</td>
<td>SourceLineNumber</td>
</tr>
<tr>
<td>ItemRevision</td>
<td>SubstituteItemIds</td>
</tr>
<tr>
<td>LotExpirationDate</td>
<td>SubstituteItemNumber</td>
</tr>
<tr>
<td>LotExpirationDate</td>
<td>TransactionQuantityUOMCode</td>
</tr>
<tr>
<td>NoteToReceiver</td>
<td>TransactionQuantityUOMCode</td>
</tr>
<tr>
<td>NoteToReceiver</td>
<td>VendorId</td>
</tr>
<tr>
<td>OrigSourceFulfillLineNumber</td>
<td>VendorItemNumber</td>
</tr>
<tr>
<td>OutsourcerContactId</td>
<td>VendorName</td>
</tr>
<tr>
<td>OutsourcerContactName</td>
<td>VendorSiteId</td>
</tr>
<tr>
<td>OutsourcerPartyId</td>
<td>VendorSiteName</td>
</tr>
<tr>
<td>OutsourcerPartyName</td>
<td>OverReceiptExceptionCode</td>
</tr>
<tr>
<td>OverReceiptTolerance</td>
<td>OverReceiptTolerance</td>
</tr>
<tr>
<td>ParentLotNumber</td>
<td>ParentLotNumber</td>
</tr>
<tr>
<td>ParentLotNumber</td>
<td></td>
</tr>
</tbody>
</table>
Attributes That You Cannot Use With Oracle Receiving

You cannot map the following attributes.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomerItemNumber</td>
<td>OrigSalesOrderLineNumber</td>
</tr>
<tr>
<td>CustomerPartyName</td>
<td>OrigSalesOrderNumber</td>
</tr>
<tr>
<td>ItemDescription</td>
<td>OrigSourceHeaderNumber</td>
</tr>
<tr>
<td>ItemNumber</td>
<td>OrigSourceLineNumber</td>
</tr>
<tr>
<td>ShipToOrganizationCode</td>
<td>ProductCategory</td>
</tr>
<tr>
<td>UnitOfMeasure</td>
<td>ProductFiscClassId</td>
</tr>
<tr>
<td>ActionCode</td>
<td>ProductType</td>
</tr>
<tr>
<td>AssessableValue</td>
<td>QuantityExpected</td>
</tr>
<tr>
<td>BUId</td>
<td>ReceiptAdviceLineLotSerials</td>
</tr>
<tr>
<td>CurrencyCode</td>
<td>ReceiptAdviceLineNumber</td>
</tr>
<tr>
<td>CustomerId</td>
<td>ReceiptAdviceLineSerials</td>
</tr>
<tr>
<td>CustomerItemId</td>
<td>ReceiptAdviceNumber</td>
</tr>
<tr>
<td>DefaultTaxationCountry</td>
<td>ReceiptAdviceOriginalSystemReference</td>
</tr>
<tr>
<td>DocumentCreationDate</td>
<td>ReceiptSourceCode</td>
</tr>
<tr>
<td>DocumentFiscalClassification</td>
<td>ReceiptSourceCode</td>
</tr>
<tr>
<td>DocumentLastUpdateDate</td>
<td>ShipToOrganizationId</td>
</tr>
<tr>
<td>DocumentLineCreationDate</td>
<td>SourceDocumentType</td>
</tr>
<tr>
<td>DocumentLineLastUpdateDate</td>
<td>SourceFulfillmentLineNumber</td>
</tr>
<tr>
<td>DocumentLineNumber</td>
<td>SourceHeaderNumber</td>
</tr>
<tr>
<td>DocumentNumber</td>
<td>SourceSystemId</td>
</tr>
</tbody>
</table>
Attributes That You Can Use With Oracle Shipping

You can map the following attributes.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrivalSetName</td>
<td>SrcAttribute1</td>
</tr>
<tr>
<td>CarrierPartyId</td>
<td>SrcAttribute10</td>
</tr>
<tr>
<td>CarrierPartyName</td>
<td>SrcAttribute11</td>
</tr>
<tr>
<td>CarrierPartyNumber</td>
<td>SrcAttribute12</td>
</tr>
<tr>
<td>CurrencyCode</td>
<td>SrcAttribute13</td>
</tr>
<tr>
<td>CustomerItemNumber</td>
<td>SrcAttribute14</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>DoNotShipAfterDate</td>
<td>SrcAttribute15</td>
</tr>
<tr>
<td>DoNotShipBeforeDate</td>
<td>SrcAttribute16</td>
</tr>
<tr>
<td>EarliestDropoffDate</td>
<td>SrcAttribute17</td>
</tr>
<tr>
<td>EarliestPickupDate</td>
<td>SrcAttribute18</td>
</tr>
<tr>
<td>EndAssemblyItemNumber</td>
<td>SrcAttribute19</td>
</tr>
<tr>
<td>FinalDischargeLocCode</td>
<td>SrcAttribute2</td>
</tr>
<tr>
<td>FobCode</td>
<td>SrcAttribute20</td>
</tr>
<tr>
<td>FreightTermsCode</td>
<td>SrcAttribute3</td>
</tr>
<tr>
<td>FulfillLineId</td>
<td>SrcAttribute4</td>
</tr>
<tr>
<td>HeaderId</td>
<td>SrcAttribute5</td>
</tr>
<tr>
<td>InitialDestinationLocationId</td>
<td>SrcAttribute6</td>
</tr>
<tr>
<td>IntendedUse</td>
<td>SrcAttribute7</td>
</tr>
<tr>
<td>ItemDescription</td>
<td>SrcAttribute8</td>
</tr>
<tr>
<td>LatestDropoffDate</td>
<td>SrcAttribute9</td>
</tr>
<tr>
<td>LatestPickupDate</td>
<td>SrcAttributeCategory</td>
</tr>
<tr>
<td>ModeOfTransport</td>
<td>SrcAttributeDate1</td>
</tr>
<tr>
<td>OrderedQuantity2</td>
<td>SrcAttributeDate2</td>
</tr>
<tr>
<td>OrderedQuantityUomCode2</td>
<td>SrcAttributeDate3</td>
</tr>
<tr>
<td>OrderedQuantityUomName2</td>
<td>SrcAttributeDate4</td>
</tr>
<tr>
<td>OutsourcerPartyId</td>
<td>SrcAttributeDate5</td>
</tr>
<tr>
<td>OutsourcerPartyName</td>
<td>SrcAttributeNumber1</td>
</tr>
<tr>
<td>OutsourcerPartyNumber</td>
<td>SrcAttributeNumber10</td>
</tr>
</tbody>
</table>

This page contains a table listing various attribute names used in the context of implementing order management in Oracle SCM Cloud.
<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PreferredGrade</td>
<td>SrcAttributeNumber2</td>
</tr>
<tr>
<td>ServiceLevel</td>
<td>SrcAttributeNumber3</td>
</tr>
<tr>
<td>ShipFromLocationId</td>
<td>SrcAttributeNumber4</td>
</tr>
<tr>
<td>ShipmentRequestHeader</td>
<td>SrcAttributeNumber5</td>
</tr>
<tr>
<td>ShipmentRequestLine</td>
<td>SrcAttributeNumber6</td>
</tr>
<tr>
<td>ShipToleranceAbove</td>
<td>SrcAttributeNumber7</td>
</tr>
<tr>
<td>ShipToleranceBelow</td>
<td>SrcAttributeNumber8</td>
</tr>
<tr>
<td>SourceLineUpdateDate</td>
<td>SrcAttributeNumber9</td>
</tr>
<tr>
<td>SourceShipmentId</td>
<td>SrcAttributeTimestamp1</td>
</tr>
<tr>
<td></td>
<td>SrcAttributeTimestamp2</td>
</tr>
<tr>
<td></td>
<td>SrcAttributeTimestamp3</td>
</tr>
<tr>
<td></td>
<td>SrcAttributeTimestamp4</td>
</tr>
<tr>
<td></td>
<td>SrcAttributeTimestamp5</td>
</tr>
<tr>
<td>TradeComplianceScreeningDate</td>
<td></td>
</tr>
<tr>
<td>TradeComplianceScreeningReason</td>
<td></td>
</tr>
<tr>
<td>TradeComplianceScreeningStatus</td>
<td></td>
</tr>
<tr>
<td>TransportationPlanningStatus</td>
<td></td>
</tr>
<tr>
<td>TransportationShipment</td>
<td></td>
</tr>
<tr>
<td>TransportationShipmentLine</td>
<td></td>
</tr>
</tbody>
</table>

**Attributes That You Cannot Use With Oracle Shipping**

You cannot map the following attributes.
<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BaseItemNumber</td>
<td>FinalDischargeLocationId</td>
</tr>
<tr>
<td>BillToContactName</td>
<td>FirstPtyRegId</td>
</tr>
<tr>
<td>BillToContactNumber</td>
<td>FobCode</td>
</tr>
<tr>
<td>BillToPartyName</td>
<td>FreightTermsCode</td>
</tr>
<tr>
<td>BillToPartyNumber</td>
<td>FromSubinventoryCode</td>
</tr>
<tr>
<td>BillToPartySiteNumber</td>
<td>HoldActionCode</td>
</tr>
<tr>
<td>BusinessUnitName</td>
<td>HoldId</td>
</tr>
<tr>
<td>CarrierPartyName</td>
<td>HoldReason</td>
</tr>
<tr>
<td>CarrierPartyNumber</td>
<td>HoldReasonCode</td>
</tr>
<tr>
<td>CategoryName</td>
<td>IntendedUseClassifId</td>
</tr>
<tr>
<td>FirstPtyNumber</td>
<td>InventoryItemld</td>
</tr>
<tr>
<td>ItemNumber</td>
<td>LineActionType</td>
</tr>
<tr>
<td>OrderedQuantityUomName</td>
<td>ModeOfTransport</td>
</tr>
<tr>
<td>OrganizationCode</td>
<td>OrderedQuantity</td>
</tr>
<tr>
<td>OrganizationCode</td>
<td>OrderedQuantityUomCode</td>
</tr>
<tr>
<td>OrganizationName</td>
<td>OrganizationId</td>
</tr>
<tr>
<td>OrganizationName</td>
<td>OrganizationId</td>
</tr>
<tr>
<td>ParentItemNumber</td>
<td>OrgId</td>
</tr>
<tr>
<td>ShipToContactName</td>
<td>PackingInstructions</td>
</tr>
<tr>
<td>ShipToContactNumber</td>
<td>ParentInventoryItemld</td>
</tr>
<tr>
<td>ShipToPartyName</td>
<td>ParentSourceShipmentId</td>
</tr>
<tr>
<td>ShipToPartyNumber</td>
<td>ProductCategory</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>ShipToPartySiteNumber</td>
<td>ProductType</td>
</tr>
<tr>
<td>SoldPartyName</td>
<td>ReleaseLock</td>
</tr>
<tr>
<td>SoldToContactName</td>
<td>RequestDateTypeCode</td>
</tr>
<tr>
<td>SoldToContactName</td>
<td>SalesOrderLineNumber</td>
</tr>
<tr>
<td>SoldToContactNumber</td>
<td>SalesOrderNumber</td>
</tr>
<tr>
<td>SoldToContactNumber</td>
<td>SalesOrderShipmentNumber</td>
</tr>
<tr>
<td>SoldToPartyNumber</td>
<td>SellingPrice</td>
</tr>
<tr>
<td>SourceSystemName</td>
<td>ServiceLevel</td>
</tr>
<tr>
<td>ThirdPtyNumber</td>
<td>ShipmentPriorityCode</td>
</tr>
<tr>
<td>ActionType</td>
<td>ShipmentRequest OrderLine Hold</td>
</tr>
<tr>
<td>AssessableValue</td>
<td>ShippingInstructions</td>
</tr>
<tr>
<td>BaseitemId</td>
<td>ShipSetName</td>
</tr>
<tr>
<td>BillToContactId</td>
<td>ShipToContactId</td>
</tr>
<tr>
<td>BillToContactId</td>
<td>ShipToContactId</td>
</tr>
<tr>
<td>BillToContactName</td>
<td>ShipToContactName</td>
</tr>
<tr>
<td>BillToContactNumber</td>
<td>ShipToContactNumber</td>
</tr>
<tr>
<td>BillToPartyId</td>
<td>ShipToPartyId</td>
</tr>
<tr>
<td>BillToPartyId</td>
<td>ShipToPartyId</td>
</tr>
<tr>
<td>BillToPartyName</td>
<td>ShipToPartyName</td>
</tr>
<tr>
<td>BillToPartyNumber</td>
<td>ShipToPartyNumber</td>
</tr>
<tr>
<td>BillToPartySiteId</td>
<td>ShipToPartySiteId</td>
</tr>
<tr>
<td>BillToPartySiteId</td>
<td>ShipToPartySiteId</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Attribute Name</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>BillToPartySiteNumber</td>
<td>ShipToPartySiteNumber</td>
</tr>
<tr>
<td>CarrierPartyId</td>
<td>SoldToContactId</td>
</tr>
<tr>
<td>CategoryId</td>
<td>SoldToContactId</td>
</tr>
<tr>
<td>ConversionDate</td>
<td>SoldToPartyId</td>
</tr>
<tr>
<td>ConversionRate</td>
<td>SoldToPartyId</td>
</tr>
<tr>
<td>ConversionType</td>
<td>SoldToPartyName</td>
</tr>
<tr>
<td>CurrencyCode</td>
<td>SoldToPartyNumber</td>
</tr>
<tr>
<td>CustPoNumber</td>
<td>SourceDocumentType</td>
</tr>
<tr>
<td>DateRequested</td>
<td>SourceHeaderNumber</td>
</tr>
<tr>
<td>DateScheduled</td>
<td>SourceLineNumber</td>
</tr>
<tr>
<td>DefaultTaxationCountry</td>
<td>SourceShipmentNumber</td>
</tr>
<tr>
<td>DocumentNumber</td>
<td>SourceSystemId</td>
</tr>
<tr>
<td>DocumentSubType</td>
<td>TaxClassificationCode</td>
</tr>
<tr>
<td>ExemptCertificateNumber</td>
<td>TaxInvoiceDate</td>
</tr>
<tr>
<td>ExemptReasonCode</td>
<td>TaxInvoiceNumber</td>
</tr>
<tr>
<td>ThirdPtyRegId</td>
<td></td>
</tr>
<tr>
<td>TrxBusinessCategory</td>
<td></td>
</tr>
<tr>
<td>UnitPrice</td>
<td></td>
</tr>
<tr>
<td>UserDefinedFiscClass</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Managing Pricing Algorithms: Explained
- Managing Service Mappings: Explained
Integrating Order Management with Other Oracle Applications: Procedure

You can configure Order Management Cloud so it uses a view object to integrate with Oracle Receivables, Oracle Shipping, Oracle Receiving, or Oracle Procurement.

You will implement the following integrations in this example:

- Integrate RuleStartDate in Oracle Receivables with ContractStartDate in Order Management
- Integrate RuleEndDate in Oracle Receivables with ContractEndDate in Order Management

You will modify the predefined FulfillmentIntegration service mapping so that it uses the following cross-references.

This topic includes example values. You might use different values, depending on your business requirements.

Integrate Order Management with another Oracle application:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, choose **Order Management**.
3. Click **Actions**, and then click **Change Feature Selection**.
4. On the Edit Features page, in the Enable Custom Payloads for Downstream Integration row, click **Features**.
5. Add a check mark to the following options that you require, click **Save and Close**, and then click **Done**.
<table>
<thead>
<tr>
<th>Option</th>
<th>Enables Integration With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop Ship</td>
<td>Oracle Procurement</td>
</tr>
<tr>
<td>Invoice</td>
<td>Oracle Receivables</td>
</tr>
<tr>
<td>Return</td>
<td>Oracle Receiving</td>
</tr>
<tr>
<td>Shipment</td>
<td>Oracle Shipping</td>
</tr>
</tbody>
</table>

6. In the Navigator, click **Pricing Administration**.

   The Order Administrator job role allows you to access the integration algorithm and service mappings that you use in the Pricing Administration work area to set up the integration.

7. On the Overview page, click **Tasks**, and then, under Order Management Configuration, click **Manage Service Mappings**.

8. On the Manage Service Mappings page, in the Name column, click **FulfillmentIntegration**.

   Notice the entities, such as ARInterfaceLine, and attributes, such as AccountRuleDuration. The FulfillmentIntegration service mapping comes predefined with these entities and attributes so that you can use them to add attributes when you integrate Order Management with the following Oracle applications.

<table>
<thead>
<tr>
<th>Oracle Application</th>
<th>Example Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Receivables</td>
<td>ARInterfaceLine</td>
</tr>
<tr>
<td>Oracle Shipping</td>
<td>ShipmentRequestHeader</td>
</tr>
<tr>
<td></td>
<td>ShipmentRequestLine</td>
</tr>
<tr>
<td>Oracle Receiving</td>
<td>ReceiptAdvice</td>
</tr>
<tr>
<td></td>
<td>ReceiptAdviceLine</td>
</tr>
<tr>
<td></td>
<td>ReceiptAdviceLineLot</td>
</tr>
<tr>
<td></td>
<td>ReceiptAdviceLineLotWithSerials</td>
</tr>
<tr>
<td>Oracle Procurement</td>
<td>PurchaseRequestHeader</td>
</tr>
<tr>
<td></td>
<td>PurchaseRequestLine</td>
</tr>
<tr>
<td></td>
<td>PurchaseRequestConfig</td>
</tr>
</tbody>
</table>

9. Click **Sources**.

   In this example, you examine the entity or the view object that defines the view object as FulfillLineVO, and then map an attribute of this entity or view object from the fulfillment line.

   Note that Order Management uses the following hierarchy.

   **Sales order header**
fulfillLine
Fulfill_line Details

ARLine is on the same level as fulfillLine.

The Entity Mappings tab lists entities, each entity references a view object, and each entity on the Attribute Mappings tab also references one or more attributes. You will create a map that references one of these entities from this view object without using an integration algorithm. However, if you must map an attribute from some other entity that does not reside at the same hierarchy level, then you must use an integration algorithm. For example, if an attribute references an extensible flexfield from the order header or the fulfillment line, then you must use an integration algorithm.

Other Oracle Fusion Applications use a similar hierarchy. For example, Purchasing uses the following hierarchy for a purchase order.

Purchase order
fulfillLine
purchaseLine

Purchasing uses the following hierarchy for shipping.

Shipping header
shipping line
fulfillLine

Purchasing uses the following hierarchy for the receiving receipt.

Receipt header
ReceiptLine
FulfillLine
Lot or lot serial details

10. In the InvoiceSources - Details area, click the row that contains ARInterfaceLine in the Entity attribute.
11. In the ARInterfaceLine - Details area, add the following values, and then click Save.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>View Object Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleStartDate</td>
<td>ContractStartDate</td>
</tr>
<tr>
<td>RuleEndDate</td>
<td>ContractEndDate</td>
</tr>
</tbody>
</table>

The RuleStartDate and RuleEndDate are now integrated to reference the view objects that you specified. At runtime, Order Management will do the following:

- Get the value of the ContractStartDate attribute from the Order Management work area, and then display it in the RuleStartDate attribute in Oracle Receivables.
- Get the value of ContractEndDate from the Order Management work area, and then display it in the RuleEndDate attribute in Oracle Receivables.

Note that you can use Expression Language (EL) in the Expression column to implement logic or a constant value. For example, you can implement the following logic:

- If the value of variable x is greater than the value of variable y, then populate a field.
Using Descriptive Flexfields to Integrate Order Management with Other Oracle Applications: Procedure

You can integrate Order Management Cloud with a descriptive flexfield that resides in Oracle Receivables, Oracle Shipping, Oracle Receiving, or Oracle Procurement.

The example in this topic describes how to integrate Order Management with Oracle Receivables. Integration with Shipping Receiving, or Procurement is similar but with the following differences:

- You make attributes available in the main entity.
- You must make sure your set up only references attributes that each application supports. For a list of these attributes, see Integrating Order Management with Other Oracle Applications: Overview.
- You must reference the database code that identifies each attribute, such as Attribute1 or AttributeDate2. You must not use the implemented names to reference the attributes.
- If you reference a descriptive flexfield according to context, then you must make sure your set up populates AttributeCategory. If you use a global segment, then it is not necessary to populate AttributeCategory.

In this example, you will integrate Order Management with Oracle Receivables. You will integrate a descriptive flexfield named Invoice Lines.

Summary of the Work

Use a descriptive flexfield to integrate Order Management with another Oracle application:

1. Get values that identify the descriptive flexfield.
2. Modify the service mapping.

This topic includes example values. You might use different values, depending on your business requirements.

Get Values That Identify the Descriptive Flexfield

Get values that identify the descriptive flexfield:

1. Use a spreadsheet editor, such as Microsoft Excel, to create a spreadsheet that contains the following values. The Source column describes the source you will use to get the value you will enter in the Value column. Leave the Value column empty for now. You will add values to it during this procedure.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Source</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Namespace</td>
<td>targetNamespace attribute from the XSD file.</td>
<td></td>
</tr>
<tr>
<td>Xsitype</td>
<td>complexType attribute from the XSD file.</td>
<td></td>
</tr>
<tr>
<td>Context Code</td>
<td>DefaultValue property of the FLEX_Context view attribute from the XML file.</td>
<td></td>
</tr>
</tbody>
</table>
2. In the Navigator, click **Setup and Maintenance**.
3. On the Setup page, click **Tasks**, click **Search**, and then search for and open Manage Receivables Descriptive Flexfields.
4. On the Manage Receivables Descriptive Flexfields page, search for the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Invoice Lines</td>
</tr>
</tbody>
</table>

5. In the Search Results area, click the **row** that contains the following value, click **Actions**, and then click **Download Flexfield Archive**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Invoice Lines</td>
</tr>
</tbody>
</table>

6. In the Confirmation dialog, wait for the archive to finish, click **Download**, and then save the file to your local computer.

   The name of this file will include the value of the Flexfield Code attribute. For this example, the file name is **222_RA_CUSTOMER_TRX_LINES.zip**.

7. Use Windows Explorer to navigate to **222_RA_CUSTOMER_TRX_LINES.zip**.

8. Expand **222_RA_CUSTOMER_TRX_LINES.zip**, expand **Oracle**, expand **Apps**, expand **Financials**, expand **Receivables**, expand **publicFlex**, expand **TransactionLineDff**, and then use an editor, such as Notepad++ or an XML editor, to open **RACUSTOMERTRXLINESCon1.xsd**.

   Note that this folder contains several XSD files. You open the XSD file that contains the context code you use for this example, which is **RA_CUSTOMER_TRX_LINES_Con1**.

9. Copy the following values to your spreadsheet, and then close the editor.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>complexType</td>
<td>RACUSTOMERTRXLINESCon1</td>
</tr>
</tbody>
</table>
For example, the following RACUSTOMERTRXLINESCon1.xsd file contains these attributes on lines 2 and 4.

```
1. <?xml version='1.0'?>
2. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="
   http://xmlns.oracle.com/apps/financials/receivables/publicFlex/TransactionLineDff/"
   xmlns="http://xmlns.oracle.com/apps/financials/receivables/publicFlex/TransactionLineDff/"
   elementFormDefault="qualified">
3.   <xs:include schemaLocation="TransactionLineFLEX.xsd"/>
4. </xs:schema>
```

10. In Windows Explorer, expand View, and then use an XML editor to open the file that includes the virtual object with the context that you require. For this example, open the file named TransactionLineFLEXRA_5FCUSTOMER_5FTRX_5FLINES_5FCon1VO.xml.

11. Copy the DefaultValue property of the FLEX_Context view attribute to your spreadsheet.

In this example, the following default value is RA_CUSTOMER_TRX_LINES_Con1.

```
80. <ViewAttribute Name="_FLEX_Context" EntityUsage="FlexfieldDataEO" EntityAttrName="AttributeCategory" IsUpdateable="false" DiscrColumn="true" DefaultValue="RA_CUSTOMER_TRX_LINES_Con1" LOVName="LOV_FLEX_Context">
81.   <Properties>
82.     <Property Name="CONTROLYTYPE" Value="choice"/>
83.     <Property Name="DISPLAYWIDTH" Value="30"/>
84.     <Property Name="FND_ACFF_AkaAttrName" Value="_FLEX_Context_DisplayValue"/>
85.     <Property Name="FND_ACFF_FlexfieldResourceBundleResId" Value="rev.a222.dra_CUSTOMER_TRX_LINES...d5p"/>
86.     <Property Name="FND_ACFF_NLS_UNIT" Value="Context Value"/>
87.   </Properties>
```

12. Identify the name of the view attribute that you must map, and then copy it to your spreadsheet.
In this example, search the XML file for a view attribute that contains RACUSTOMERTRXLINESSSeg1:

```xml
<ViewAttribute Name="RACUSTOMERTRXLINESSSeg1" EntityUsage="FlexfieldDataEO" EntityAttrName="Attribute1" IsUpdateable="false">
  <Properties>
    <Property Name="FND_ACFF_FlexfieldResourceBundleResId" Value="rev.a22.e6RA_CUSTOMER_TRX_LINES..d4p.RA_CUSTOMER_TRX_LINES_C"> CUSTOMER_TRX_LINES_S_Seg1"/>
    <Property Name="FND_ACFF_NLS_INIT" Value=""/>
    <Property Name="FND_ACFF_OpenIdentifier" Value=""/>
    <Property Name="FND_ACFF_SegmentName" Value=""/>
    <Property Name="FORMTYPE" Value="Summary"/>
    <Property Name="LABEL_ResId" Value=""/>
  </Properties>
</ViewAttribute>
```

13. On the Edit Descriptive Flexfield page, click **Manage Contexts**.
14. On the Manage Contexts page, search for the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context Code</td>
<td>RA_CUSTOMER_TRX_LINES_Con1</td>
</tr>
</tbody>
</table>

15. In the Search Results, click **Actions**, and then click **Edit**.
16. On the Edit Context page, copy the following value to your spreadsheet, and then cancel pages until you are back on the Setup and Maintenance page.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>API Name</td>
<td>RACUSTOMERTRXLINESSCon1</td>
</tr>
</tbody>
</table>

17. Verify that your spreadsheet contains the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Source</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xstype</td>
<td>complexType attribute from the XSD file.</td>
<td>RACUSTOMERTRXLINESSCon1</td>
</tr>
<tr>
<td>Context Code</td>
<td>DefaultValue property of the FLEX_Context view attribute from the XML file.</td>
<td>RA_CUSTOMER_TRX_LINES_Con1</td>
</tr>
<tr>
<td>Segmentname</td>
<td>Name of the view attribute from the XML file.</td>
<td>RACUSTOMERTRXLINESSSeg1</td>
</tr>
</tbody>
</table>
Modify the Service Mapping

You will administer a service mapping that implements the following service data object. For details about the service data object, see Pricing Architecture: How It Works.

Explanation of Callouts

This SDO (service data object) includes the following objects:

1. **Source.** Provides a structure so Order Management can model data in the input SDO (service data object). In this example, you will modify the InvoiceSources source on the predefined FulfillmentIntegration service mapping. You will map this source to the TransactionLineDff entity. This entity comes predefined with the NameSpace, XsiType, and _FLEX_Context attributes. You will add the attribute named RACUSTOMERTRXLINESCon1_Custom. The FulfillmentIntegration context comes predefined with the descriptive flexfields and global descriptive flexfields that Order Management supports.

2. **Service.** Requests the service mapping and receives the output SDO. In this example, the InvoiceService service references the entities and attributes that the integration algorithm uses to get the value of the descriptive flexfield.

3. **Entity.** The entity that the service mapping requires to structure the output SDO. In this example, you modify the TransactionLineDff entity so that it references the descriptive flexfield.

4. **Attributes.** The attributes that the service mapping requires to structure the output SDO. You will configure these attributes so that they reference objects and properties in Oracle Receivables.
Modify the service mapping:

1. Log into Oracle Management with the Order Administrator job role, and then, in the Navigator, click **Pricing Administration**.
   The Order Administrator job role allows you to access the integration algorithm and service mappings that you use in the Pricing Administration work area to set up the integration.

2. On the Overview page, click **Tasks**, and then, under Order Management Configuration, click **Manage Service Mappings**.

3. On the Manage Service Mappings page, in the Name column, click **FulfillmentIntegration**.

4. On the Edit Service Mappings page, on the Entities tab, click the **row** that contains the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>TransactionLineDff</td>
</tr>
</tbody>
</table>

5. Click **Sources**.

6. In the InvoiceSources - Details area, on the Entity Mappings tab, click the **row** that contains the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>TransactionLineDff</td>
</tr>
</tbody>
</table>

7. On the Attribute Mappings tab, add the following mappings, and then click **Save**. Copy values from your spreadsheet into the Expression.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameSpace</td>
<td><a href="http://xmlns.oracle.com/apps/financials/receivables/publicFlex/TransactionLineDff/">http://xmlns.oracle.com/apps/financials/receivables/publicFlex/TransactionLineDff/</a></td>
</tr>
<tr>
<td>XsiType</td>
<td>RACUSTOMERTRXLINESCon1</td>
</tr>
<tr>
<td>RACUSTOMERTRXLINESCon1_Custom</td>
<td>Leave empty.</td>
</tr>
</tbody>
</table>

8. Click **Services**, and then click the **row** that contains the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>TransactionLineDff</td>
</tr>
</tbody>
</table>

9. In the TransactionLineDff - Entities area, add a row for each of the following entities. Make sure the Read attribute and the Write attribute each contain a check mark for each row.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>_FLEX_Context</td>
<td>Leave empty.</td>
</tr>
<tr>
<td>XsiType</td>
<td>Leave empty.</td>
</tr>
</tbody>
</table>
Using Extensible Flexfields to Integrate Order Management with Other Applications: Procedure

You can use extensible flexfields to integrate Order Management Cloud with Oracle Receivables, Oracle Shipping, Oracle Receiving, or Oracle Procurement.

In the example in this topic, you will integrate with Oracle Receivables. Assume you implemented the following details for the header extensible flexfield:

- HeaderContext as the context
- H1AttributeNum as the segment

You will add a header extensible flexfield in the service mapping, store the Id of the primary salesperson in the extensible flexfield, and then map it to the Id of the primary salesperson in the Accounts Receivable line.

Summary of the Work

Use an extensible flexfield to integrate Order Management with Oracle Receivables:

1. Get values that identify the extensible flexfield.
2. Modify the service mapping.
3. Create an integration algorithm.

This topic includes example values. You might use different values, depending on your business requirements.

Get Values That Identify the Extensible Flexfield

Get values that identify the extensible flexfield:

1. Use a spreadsheet editor, such as Microsoft Excel, to create a spreadsheet that contains the following values.
2. In the Navigator, click Setup and Maintenance.
3. On the Setup page, click Order Management.
4. Search for, and then open Manage Extensible Flexfields.
5. On the Manage Extensible Flexfields page, search for the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module</td>
<td>Process Order</td>
</tr>
</tbody>
</table>

6. In the Search Results area, click the row that contains the following value, click Actions, and then click Download Flexfield Archive.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Header Information</td>
</tr>
</tbody>
</table>

7. In the Confirmation dialog, wait for the archive to finish, click Download, and then save the file to your local computer.
The name of this file will include the value of the Flexfield Code attribute. For this example, the file name is 10008_DOO_HEADERS_ADD_INFO.zip.
8. Use Windows Explorer to navigate to 10008_DOO_HEADERS_ADD_INFO.zip.
9. Expand 10008_DOO_HEADERS_ADD_INFO.zip, expand Oracle, expand Apps, expand SCM, expand DOO, expand processOrder, expand flex, expand headerContextsB, expand View, and then use an editor, such as Notepad++ or an XML editor, to open HeaderEffBHeaderContext1privateVO.xml.
You will examine the extensible flexfield context named HeaderContext1.
10. Copy the following value to your spreadsheet.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of ViewObject</td>
<td>HeaderEffBHeaderContext1privateVO</td>
</tr>
</tbody>
</table>

For example, the HeaderEffBHeaderContext1privateVO.xml file contains the following view object name.

```xml
<Property Name="INTERNAL_SERVICE_SUPPRESS_XSDGEN" Value="true"/>
<Property Name="CONTEXT_CODE_COLUMN_NAME" Value="CONTEXT_CODE"/>
<Property Name="FND_ACF1_ApplicationID" Value="10008"/>
<Property Name="FND_ACF1_ApplicationModuleDefFullName" Value="oracle.apps.scm.doo.processOrder"/>
<Property Name="FND_ACF1_ApplicationName" Value="Distributed Order Orchestration"/>
<Property Name="FND_ACF1_ApplicationShortName" Value="DOO"/>
<Property Name="FND_ACF1_Rollup" Value="true"/>
```

11. Copy the name of the attribute that you will use to store the Id of the primary salesperson to your spreadsheet.
Look for the ViewAttribute that matches the segment you are using to store the value you require. For this example, copy the following value.
For example, the `HeaderEffBHeaderContext1privateVO.xml` file contains the following view attribute name.

```xml
<Property Name="FND_ACFF_FlexfieldResourceBundleResId" Value="rev.a10008.dDO_HEADERS_ADD_IN"/>
<Property Name="FND_ACFF_MLS_UNIT" Value="H1AttributeNum1"/>
<Property Name="FND_ACFF_Precision" Value="15"/>
<Property Name="FND_ACFF_Scale" Value="0"/>
<Property Name="FND_ACFF_SegmentName" Value="H1AttributeNum1"/>
<Property Name="FND_ACFF_SegmentName" Value="Summary"/>
<Property Name="FND_ACFF_RespId" Value="${adfBundle["oracle.apps.fndapplcore.flex.runtime.flexf..."/>
<Property Name="TOOLTIP_ResId" Value="${adfBundle["oracle.apps.fndapplcore.flex.runtime.file..."
</Properties>
```

### 12. Verify that your spreadsheet contains the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Source</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViewObject</td>
<td>XML file.</td>
<td><code>HeaderEffBHeaderContext1privateVO</code></td>
</tr>
<tr>
<td>ViewAttribute</td>
<td>XML file.</td>
<td><code>_H1AttributeNum1</code></td>
</tr>
</tbody>
</table>
Modify the Service Mapping

You will modify the predefined FulfillmentIntegration service mapping so that it resembles the following set up.

Modify the service mapping:

1. Log into Oracle Management with the Order Administrator job role, and then, in the Navigator, click **Pricing Administration**.

   This job role allows you to access the integration algorithm and service mappings that you use in the Pricing Administration work area to set up the integration.

2. On the Overview page, click **Tasks**, and then, under Order Management Configuration, click **Manage Service Mappings**.

3. On the Manage Service Mappings page, in the Name column, click **FulfillmentIntegration**.
4. On the Edit Service Mappings page, on the Entities tab, click **Actions**, click **Add Row**, and then set the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>HeaderSalesPersonEFF_Custom</td>
</tr>
<tr>
<td>Description</td>
<td>Context for an extensible flexfield on the order header. It contains details about the salesperson.</td>
</tr>
</tbody>
</table>

5. In the Details area, add the following attributes, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeaderId_Custom</td>
<td>Long</td>
</tr>
<tr>
<td>SalesPersonId_Custom</td>
<td>Long</td>
</tr>
</tbody>
</table>

6. Click **Sources**.

7. In the InvoiceSources - Details area, on the Entity Mappings tab, click **View**, click **Columns**, and then add a check mark to each of the following values:

- Joined Entity
- Joined Entity Attribute
- Use Existing View Object

8. Click **Actions**, click **Add Row**, set the following values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>HeaderSalesPersonEFF_Custom</td>
</tr>
<tr>
<td>Type</td>
<td>View Object</td>
</tr>
<tr>
<td>View Object</td>
<td>HeaderEffB-HeaderContext1privateVO</td>
</tr>
<tr>
<td>Query Type</td>
<td>Join</td>
</tr>
<tr>
<td>Query Attribute</td>
<td>HeaderId</td>
</tr>
<tr>
<td>Use Existing View Object</td>
<td>Contains a check mark.</td>
</tr>
<tr>
<td>Joined Entity</td>
<td>Header</td>
</tr>
<tr>
<td>Joined Entity Attribute</td>
<td>HeaderId</td>
</tr>
</tbody>
</table>

9. On the Attribute Mappings tab, add the following attributes, and then click **Save**.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>View Object Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeaderId_ Custom</td>
<td>HeaderId</td>
</tr>
<tr>
<td>SalesPersonId_ Custom</td>
<td>_ H1AttributeNum1</td>
</tr>
<tr>
<td></td>
<td>Copy the value from your spreadsheet.</td>
</tr>
</tbody>
</table>

10. On the Entity Mappings tab, click the **row** that contains the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>View Object Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>ARInterfaceLine</td>
</tr>
</tbody>
</table>

11. In the ARInterfaceLine - Details area, click **Actions**, click **Add Row**, set the following value, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>ResourceSalesrepId</td>
</tr>
</tbody>
</table>

12. Click **Services**.

13. In the InvoiceService - Details area, click **Actions**, click **Add Row**, and then set the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>HeaderSalesPersonEFF_ Custom</td>
</tr>
</tbody>
</table>

14. In the HeaderSalesPersonEFF_Custom - Entities area, add the following attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeaderId_ Custom</td>
<td>Leave empty.</td>
</tr>
<tr>
<td>SalesPersonId_ Custom</td>
<td>Leave empty.</td>
</tr>
</tbody>
</table>

15. In the InvoiceService - Details area, click the **row** that contains the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>Header</td>
</tr>
</tbody>
</table>

16. In the Header - Entities area, add the following attribute.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeaderId</td>
<td>Leave empty.</td>
</tr>
</tbody>
</table>
17. In the InvoiceService - Details area, click the **row** that contains the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>ARInterfaceLine</td>
</tr>
</tbody>
</table>

18. In the ARInterfaceLine - Entities area, add the following attribute, and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResourceSalesrepId</td>
<td>Leave empty.</td>
</tr>
</tbody>
</table>

**Create an Integration Algorithm**

Create an integration algorithm:

1. Click **Tasks**, and then, under Order Management Configuration, click **Manage Algorithms**.
2. On the Manage Algorithms page, click **Actions**, and then click **Create**.
3. On the Create Algorithm page, set the following value, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Accounts Receivable Mapping Custom</td>
</tr>
</tbody>
</table>

4. Click **Variables**, click **Actions**, click **Add Row**, set the following values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>ARIntegration</td>
</tr>
<tr>
<td>Data Type</td>
<td>Data Object</td>
</tr>
<tr>
<td>Input and Output</td>
<td>Input and Output</td>
</tr>
<tr>
<td>Internal Service Schema</td>
<td>FulfillmentIntegration. InvoiceService</td>
</tr>
</tbody>
</table>

5. Click **Algorithm**, click **Add Step**, click **Conditional Action**, set the following values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Map Salesperson Id</td>
</tr>
<tr>
<td>Description</td>
<td>Map the primary salesperson from the extensible flexfield that resides on the order header.</td>
</tr>
</tbody>
</table>

6. In the Data Sets area, add the following data sets, and then click **Save**.
where:

- **ARInterfaceLine**. Specifies to use the ARInterfaceLine entity of the service mapping to get order details, and then store them in ARIntegration, which is the variable you added to this integration algorithm earlier in this topic.

  This integration uses ARIntegration to temporarily store the data that this integration algorithm will communicate through the service mapping. The Data Set Join is empty in the ARLine primary data set because the secondary data sets populate ARLine.

- **FulfillLine**. Filters the results in ARIntegration according to fulfillment line.

  The join \([\text{FulfillLineId: ARLine.FulfillLineId}]\) uses the FulfillLineId attribute of the FulfillLine entity from the service mapping to filter the primary ARLine data set so that it contains only the fulfillment line that FulfillLineId references.

- **HeaderSalesPersonEFF_Custom**. Filters the results in ARIntegration according to order header.

  The join \([\text{HeaderId_Custom: FLine.HeaderId}]\) uses the HeaderId attribute of the HeaderSalesPersonEFF_Custom entity to filter the FLine data set so that it contains only order headers.

Note the following points:

- In the Variable Path, make sure you use the exact entity names from the service mapping, such as ARInterfaceLine, FulfillLine, and HeaderSalesPersonEFF_Custom.
- Make ARLine the primary.
- Each group of data sets includes one primary data set and one or more secondary data sets.
- Cardinality specifies that there is zero or one records in the secondary data set to one or many records in the primary data set. It determines whether the join is an inner join or outer join.
- Data Set Join defines the constraint that this step uses to filter records that the secondary data set saves in the primary.
- FLine represents a row from the FulfillLine entity that matches the join condition.
- HdrEFF represents a row from the HeaderSalesPersonEFF_Custom entity that matches the join condition.
- ARInterfaceLine and FulfillLine come predefined.
- You create HeaderSalesPersonEFF_Custom in the service mapping.
- ARInterfaceLine comes predefined with an attribute named ResourceSalesRepId. Your set up will populate ResourceSalesRepId with the value from the SalesPerson_Custom attribute in the
HeaderSalesPersonEFF_Custom entity that you define. You define HeaderSalesPersonEFF_Custom in the join conditions for the data set so that it uniquely identifies the row that your set up uses to perform the mapping.

7. In the Execute Condition area, click **Add Condition**, click **Default Action**, add the following value, and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>ARLine.ResourceSalesrepIdHdrEFF. SalesPersonCustom</td>
</tr>
</tbody>
</table>

where:

- **ARLine.ResourceSalesrepId**. Specifies to store the result in the ResourceSalesrepId attribute of the ARInterfaceLine entity on the service mapping.

8. On the Manage Algorithms page, click **Actions**, and then click **Publish**.

9. Verify that the Status displays Published, and then click **Done**.


11. Click **Services**, set the following values, and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>InvoiceService</td>
</tr>
<tr>
<td>Implementation Type</td>
<td>Algorithm</td>
</tr>
<tr>
<td>Implementation</td>
<td>Accounts Receivable Mapping Custom</td>
</tr>
</tbody>
</table>

where:

- **Accounts Receivable Mapping Custom** is the name of the integration algorithm that you created.

**Related Topics**

- Pricing Architecture: How It Works
- Managing Pricing Algorithms: Explained
- Managing Service Mappings: Explained

**Specifying Transaction Types to Send from Order Management to Oracle Receivables: Procedure**

A sales order might include details that you must use to correctly determine accounting for a financial transaction, such as an invoice memo, or a debit and credit memo. You can define a business rule that sets the value for the AR Transaction Type attribute according to details from the sales order, such as business unit, order type, line type, or item.

In following example, assume you must create an invoice in Oracle Receivables with a different transaction type for each sales order. You can use AR Transaction Type to specify the accounting entries to use for the transaction.
You specify AR Transaction Type in a pretransformation or posttransformation defaulting rule. Write your rule so it uses one of the following combinations of attributes.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line category and business unit</td>
<td>Use this combination when you must assign a business unit according to the following transaction types:</td>
</tr>
<tr>
<td></td>
<td>• Use the same AR Transaction Type, such as ARTT-Invoice1, for order lines that you are fulfilling.</td>
</tr>
<tr>
<td></td>
<td>• Use the same AR Transaction Type, such as ARTT-Credit1, for return order lines.</td>
</tr>
<tr>
<td>Use one of the following combinations:</td>
<td>Use this combination when you must assign a business unit according to the following transaction types:</td>
</tr>
<tr>
<td>• Line type and line category, and business unit</td>
<td>• Use more than one AR Transaction Type for order lines that you are fulfilling.</td>
</tr>
<tr>
<td>• Line type or line category, and business unit</td>
<td>• Use more than one AR Transaction Type for return order lines.</td>
</tr>
<tr>
<td>Order category and business unit</td>
<td>Use this combination when your source system does not include order lines that you are fulfilling and return order lines on the same source order.</td>
</tr>
<tr>
<td>Order type and business unit</td>
<td>Use this combination when your implementation does not include order lines that you are fulfilling and return order lines on same sales order.</td>
</tr>
</tbody>
</table>

Oracle Receivable does not default a value for Receivables Transaction. If you do not define a pretransformation or posttransformation defaulting rule, then Order Management uses the predefined behavior. It uses the Invoice transaction type for order lines that you are fulfilling, and it uses Credit Memo for return order lines.

In the following example, you define a pretransformation that implements the following pseudocode:

- If the inventory item Id on the fulfillment line is 149, then set the transaction type to 1.

You will define the following rule.

For details about how to create a business rule, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then download the attachment that describes how to use business rules.

Specify the transaction types to send from Order Management to Oracle Receivables:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Orchestration Process Assignment Rules.
4. Create a new rule using the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Billing Rule</td>
</tr>
<tr>
<td>Effective Date</td>
<td>Always</td>
</tr>
<tr>
<td>Active</td>
<td>Contains a check mark</td>
</tr>
<tr>
<td>Advanced Mode</td>
<td>Contains a check mark</td>
</tr>
</tbody>
</table>

5. Create the following If statement:

   `Fline is a PostTransformationRules.FulfillLineVO`

6. Create the following And statement:

   `Fline.InventoryItemId is 149`

7. Create the following Then statement, and then click **Save and Close**:

   `assert new PostTransformationRules.ModifyEntity (attrName:"BillingTrxTypeId", attrValue:1, viewRowImpl:Fline.ViewRowImpl)`

8. Test your work:

   - Navigate to the Order Management work area, and then create a new sales order.
   - In the Order Lines area, add an item that references the inventory item you specified earlier in this procedure.
   - On the Billing and Payment Details tab, in the Order Line Details area, on the order line you must modify, click **Edit Accounting Details**.
   - In the Edit Accounting Details dialog, set the Receivables Transaction attribute, and then click **Submit**.
   - Verify that Oracle Receivables correctly categorized the transaction.

---

**Integrating Order Management with Transportation Management**

**Integrating Order Management with Transportation Management: Overview**

You can integrate Order Management Cloud with Oracle Transportation Management or some other transportation management system that resides outside of Oracle so that it sends sales orders for transportation planning, and receives updates at fulfillment milestones, such as plan completion or proof of delivery.
An integration between Order Management and Oracle Transportation Management can help you achieve the following results:

- Allow your users to search sales orders according to scheduled ship date, scheduled arrival date, actual delivery date, or transportation planning order.
- Use Transportation Management to schedule and optimize shipments, finalize the transportation plan, tender shipments to carriers, and send updated schedule dates to Order Management, providing the Order Manager with visibility to the new shipment schedule.
- Update Order Management after the customer receives the shipment. You can use these dates and statuses to identify exceptions and control the orchestration process flow.
- Provide the final Delivered status after the item ships.
- Allow Order Management to leverage your transportation management system so that it provides efficient transportation planning. The transportation management system can optimize the shipment that ships your sales orders to minimize freight costs while meeting your customer delivery requirements.
- Allow Order Management to use the same process for order changes that it uses for the original sales order.

Integrating Order Management with Transportation Management: How It Works

An integration between Order Management Cloud and Oracle Transportation Management can include several Oracle solutions, such as Oracle Global Trade Management, Inventory Management Cloud, and so on.

Note that Order Management does not come predefined to integrate with Transportation Management or Global Trade Management. However, starting in Release 13, you can use Integration Cloud Service, a synchronous web service, or an asynchronous web service to create an integration with a transportation management system. In releases earlier than Release 13, you can use only an asynchronous web service. For details, see Using Integration Cloud Service to Integrate Order Management with Other Systems: Overview.
The following diagram illustrates the flow that this integration uses.

**Explanation of Callouts**

In general, this integration uses the following flow:

1. A Compliance Manager uses the administrative interface in Global Trade Management to define the party that identifies the customer and to classify the item.
2. An Order Entry Specialist creates a sales order in Order Management. This sales order references the customer and item that the Compliance Manager created.
3. Screening performs the following process:

   This process includes the following details:
   - Inventory Management Cloud sends a screening request through Integration Cloud Service to Global Trade Management. Note that you can also configure Order Management to perform trade compliance screening before submitting the sales order, and you can use Inventory Management Cloud to screen during shipping.
Global Trade Management screens the party for trade compliance according to the request details that you specify in Inventory Management Cloud. For example, you can request to screen shipment lines at a desired stage in the fulfillment process according to your business requirements, such as when fulfillment has staged the fulfillment lines but has not yet shipped them. Global Trade Management can also screen the transaction for export controls, and can screen the party and transaction for licensing.

Global Trade Management sends the screening results through Integration Cloud Service to Inventory Management Cloud.

4. Order Management orchestrates the sales order.

5. Transportation Management receives a transportation management request for the fulfillment line from Order Management, acknowledges the request, and converts it to an order release. Note the following points:

- Transportation Management also creates a transportation planning order.
- Transportation Management examines the supply chain configuration, creates an optimal transportation shipment for one or more fulfillment lines, creates a transportation plan, and then creates and sends a shipment request to Oracle Fusion Shipping.
- Transportation Management can create different types of direct or consolidated shipments, including single origin truckload, single destination truckload, multistop truckload, less than truckload (LTL), parcel, ocean, rail, and so on.

6. The Order Management work area updates the screening status on the Order page during orchestration. It also updates the transportation planning order and orchestration process number on the Fulfillment Lines tab, and it updates the status of the orchestration plan on the Orchestration Process tab with the updates that it receives from Transportation Management. For example:

7. Oracle Fusion Shipping picks and packs the sales order in the warehouse.

8. Global Trade Management creates the export documents, if necessary. It also creates the Shipment Trade Transaction. As an option, it can also screen for compliance according to how you configure it for screening during order fulfillment, and then send the screening results and reason to Shipping.

9. Oracle Transportation Management creates and sends the tender to carrier. If necessary, Global Trade Management declares the export.

10. Oracle Shipping ships the item and sends a shipment advice to Order Management and Transportation Management. The customer receives and acknowledges receipt, and then Transportation Management sends the shipment delivery confirmation to Order Management. Order Management sends the fulfillment line to Oracle Fusion Accounts Receivable for invoicing, and then sets the fulfillment line status to Closed.
Setting Up Orchestration Processes to Perform Transportation Management Tasks

You can set up an orchestration process so that it performs Transportation Management tasks. For details, see Integrating Order Management with Transportation Management: Procedure. Note the following points:

- You use the DOO_TransportationPlanning task type in an orchestration process to create the integration. This task type allows Order Management to send a new request to the transportation management system, and to update or cancel an existing request.
- You can reference DOO_TransportationPlanning at any point in the stepwise sequence of an orchestration process, depending on your business requirements. For example, you can configure an orchestration process so that it identifies the source of supply, reserves inventory, and then performs transportation planning.
- Order Management can process the response it receives from Transportation Management at different steps in the orchestration process, such as after the transportation plan finishes, or after it receives proof of delivery. Order Management can use this response after the transportation plan finishes even if the orchestration process is not currently running the DOO_TransportationPlanning task.
- The DOO_TransportationPlanning task type uses an indicator to mark each fulfillment line that Transportation Management planned so that some other system can process it. For example, the predefined integration with Inventory Cloud sets a transportation planning hold on each shipment request that references a fulfillment line that contains this indicator. This integration releases the hold only after Transportation Management receives the transportation shipment plan.

The following relationship exists between the fulfillment line and the DOO_TransportationPlanning task type in the orchestration plan.
Integrating Order Management with Transportation Management: Procedure

You use Integration Cloud Service to create an integration between Order Management Cloud and Transportation Management. You can also set up an orchestration process so it performs Transportation Management tasks.

In the example in this topic, you will create the following orchestration process so that it sends the request for transportation planning after the fulfillment system identifies a source and reserves inventory.

This topic includes example values. You might use different values, depending on your business requirements.

Integrate Order Management with Transportation Management:

1. Use Integration Cloud Service to create an integration between Order Management and Transportation Management.

   Make sure you set the business event trigger points that your deployment requires to start this integration process. For details, see Using Integration Cloud Service to Integrate Order Management with Other Systems: Overview.

2. Log into Order Management with administrative privileges.

3. In the Navigator, click Setup and Maintenance.


5. Search for, and then open Manage Orchestration Process Definitions.

   For details about how to configure an orchestration process, see Defining Orchestration Processes: Explained.


7. In the Search Results, click Actions, and then click Duplicate.

   Note that you can integrate Order Management with Transportation Management only through a long-running task.

8. In the Edit Orchestration Process Definition dialog, set the following values, and then click Save.
9. Click **Status Conditions**.

10. In the Orchestration Process Status Values area, add the following status values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Awaiting Transportation Planning Response</td>
<td>“Transportation Planning&quot; = “DOO_TP_AWAIT_ACCEPT“</td>
</tr>
<tr>
<td></td>
<td>You can use this expression as an exit condition for the Wait step.</td>
</tr>
<tr>
<td>Transportation Planning Request Accepted</td>
<td>“Transportation Planning&quot; = “DOO_TP_REQ_ACCEPT“</td>
</tr>
<tr>
<td>Transportation Planning Request Rejected</td>
<td>“Transportation Planning&quot; = “DOO_TP_REQ_REJECT“</td>
</tr>
<tr>
<td>Transportation Planning Completed</td>
<td>“Transportation Planning&quot; = “DOO_TP_PLANNED“</td>
</tr>
</tbody>
</table>

11. On the Step Definition tab, add the following step at any location in the sequence of steps. This step sends the sales order to the fulfillment system for transportation planning. For this example, add this step immediately after the Create Reservation step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Send Request to Transportation Management</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task Type</td>
<td>DOO_TransportationPlanning</td>
</tr>
<tr>
<td>Task</td>
<td>TransportationPlanning</td>
</tr>
<tr>
<td>Service</td>
<td>Create Transportation Planning</td>
</tr>
<tr>
<td>Update Service</td>
<td>Update Transportation Planning</td>
</tr>
<tr>
<td>Cancel Service</td>
<td>Cancel Transportation Planning</td>
</tr>
<tr>
<td>Use Transactional Item Attributes</td>
<td>Contains a check mark.</td>
</tr>
</tbody>
</table>
12. Optional. Add the following step immediately after the Send Request to Transportation Management step.

This step temporarily pauses the orchestration process so that Transportation Management can validate the planning request before the orchestration process creates the shipment request. It is recommended that you configure your orchestration process so that it processes the response while waiting on a pause step. However, an orchestration process can process the response on any step. If you do not add this step, then the orchestration process will continue to run and fulfill the sales order without waiting for validation from Transportation Management.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Flexfield Attributes</td>
<td>Contains a check mark.</td>
</tr>
<tr>
<td>Step Name</td>
<td>Wait for Transportation Management</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task Type</td>
<td>DOO_TransportationPlanning</td>
</tr>
<tr>
<td>Task</td>
<td>TransportationPlanning</td>
</tr>
<tr>
<td>Service</td>
<td>Wait for Transportation Planning Validation</td>
</tr>
</tbody>
</table>
| Exit Criteria            | Add a check mark to each of the following criteria:  
  ◦ Canceled  
  ◦ Transportation Planning Request Accepted |
| Next Expected Task Status| Transportation Planning Request Accepted |
| Use Transactional Item Attributes | Contains a check mark. |
| Use Flexfield Attributes | Contains a check mark. |

**Related Topics**

- Defining Orchestration Processes: Explained

**More Details About Integrating Order Management**
Integrating Order Management to Communicate Attachments: Explained

Order Management Cloud allows users to add an attachment to a sales order, such as a document that includes requirements for manufacturing, a memo that includes guidelines for negotiating a price, or a URL to a page that includes item installation instructions. You must integrate Order Management so that it can add these attachments.

Note the following:

- Users can view attachments in the Order Management work area and send them to a fulfillment system.
- Order Management can receive an attachment from a source system and include it as part of a sales order.
- Order Management cannot receive an attachment from a fulfillment system.
- Order Management send an attachment to an order capture system.

To allow Order Management to receive and send sales order attachments, a Supply Chain Administrator or Controller must do the following work:

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow Order Management to receive sales order attachments from an order capture system.</td>
<td>Collect the attachment category during orchestration data collection.</td>
</tr>
<tr>
<td>Allow Order Management to send sales order attachments to a fulfillment system.</td>
<td>Call the AttachmentsAM public service. Use this service to select and send attachments according to the type of the fulfillment request and the attachment category.</td>
</tr>
</tbody>
</table>

For details, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then download the attachment that describes attachments.

Collecting Orchestration Data: Explained

Order Management Cloud cross-references attributes and data that reside in an order capture system and each fulfillment system so that it can use a single representation across these different systems. The order capture systems and fulfillment systems use these values to communicate with Order Management. You must use the Oracle Fusion Customer and Product Models to set up customer data, product master data, and cross-references.

Make sure you collect the following reference entities for validation and cross-referencing:

- Activity types
- Currencies, conversions, and types
- Demand class
- Document category
- FOB points
- Freight term
- Invoicing and accounting rules
- Payment methods
- Payment terms
- Receipt methods
- Return reason
- Resources
- Sales credit types
- Shipment priority
- Shipping carriers
- Shipping class of service
- Shipping mode of transport
- Tax classification code
- Tax exemption reason
- Units of measure
- Warehouse

These attributes are not related to the customer or the item.

Oracle provides the following data collection tools:

- **Continuous collection.** Collects data incrementally. It provides fast collection for each entity that Order Management must source.

- **Targeted collection.** Refreshes data for a selected business object. You can run a scheduled process that performs targeted collection as needed or on a schedule.
4 Importing and Transforming Source Orders

Importing Source Orders

Order Import Template: Explained

You can use the Order Import Template to help reduce errors and simplify order import. This template contains a structure that the Oracle database requires. It includes a tab for each database table, and it displays these tabs in a specific sequence. It includes columns on each tab that represent the table columns that Oracle requires, and it specifies the data type that Oracle requires for each database column.

To get a copy of the Order Import Template, and to get details about the tables that the template references, see the Import Sales Orders topic in the File-Based Data Import for Oracle Supply Chain Management Cloud guide on Oracle Help Center at https://docs.oracle.com/cloud/latest/scmos_gs/OFSC/Import_Sales_Orders_305258851_fbd_16.htm#305258851.

For details about how to use the Order Import Template, see Using Files to Import Orders into Order Management: Procedure.

You can also use a web service to create an integration that imports source orders. For details, search for Doc ID 2051640.1 (Using Web Services with Oracle Fusion Order Management Cloud) on My Oracle Support at https://support.oracle.com.

Guidelines for Using the Order Import Template

It is recommended that you use the following guidelines when you use the Order Import Template:

- The Order Import Template uses a separate spreadsheet in an Excel workbook to represent each interface table. A row near the top of each spreadsheet contains column headers. Each column header represents the name of an interface table column. The spreadsheet displays these columns in a specific sequence. You must not modify this sequence. If you do, then the import will fail.

- The Order Import Template comes predefined with example data already populated, and some column headers include descriptive text. Use this example data and the descriptive text to help you determine the type of data that you must include. For example, click the DOO_ORDER_LINES_ALL_INT tab, and notice that row five in the Source Transaction System column includes a value of LEG, which is a text abbreviation for the term legacy. Position your mouse over the Source Transaction System column header, and then read the descriptive text that indicates this field must contain only VARCHAR2(30) type data. The value LEG meets this requirement.

- The data that you enter into the spreadsheet must use a format that the template can accept and process according to the format that the database column uses. For example, data that you enter in the Source Transaction Identifier column of the DOO_ORDER_LINES_ALL_INT tab must use the VARCHAR2(50) format. For details about these formats, see the topic about Oracle data types at https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SCN73/ch6.htm.

- You must enter data for each required column. The template uses an asterisk ( * ) to indicate each required column.

- Required columns are not always contiguous. It is recommended that you carefully examine the column headings in each spreadsheet to make sure you include data for each required column. Tip: to locate all the required headings on a spreadsheet, press CTRL + F, enter ~* (a tilde and an asterisk), and then click Find All.

- The template uses a blue background color for column headings, by default. It uses other colors to group some columns. For example, the DOO_ORDER_HEADERS_ALL_INT sheet uses a tan color (red 253, green 233, and blue 217) to group the Buying Party Identifier, Buying Party Name, and Buying Party Number columns. You must enter a value in at least one column of this color group. Note that these columns might not display contiguously. It
is recommended that you carefully examine the color group on each spreadsheet to make sure you include at least one value for the color group.

- Some color groups are conditional. For example, you must enter a value for at least one column in the color group that represents the Requested Fulfillment Organization Identifier, Requested Fulfillment Organization Code, and Requested Fulfillment Organization Name columns on the DOO_ORDER_LINES_ALL_INT spreadsheet only for each row in this spreadsheet that references a return order. The spreadsheet includes an instructional note immediately above each conditional color group.

- Some sheets include more than one color group. You must enter at least one value for each color group.

- You must use the YYYY/MM/DD/HH/MI/SS format for each date field.

- You must use a comma (,) for the thousands separator and a period (.) for the decimal separator in the Amount column. For example, use 1,500.25. Do not use 1500,25.

- If a column must use a whole number, then each control file validates that the data in this column includes only whole numbers. If it does not, then the import fails.

- Use the descriptive text in the spreadsheet for assistance with columns that require an internal identifier.

- You can hide columns that you do not require, but you must not delete them. If you delete a column, then the import will fail.

- At a minimum, it is recommended that you include details in each of the following tables. If you do not include these details, then your sales orders will not be complete:
  - DOO_ORDER_HEADERS_ALL_INT
  - DOO_ORDER_LINES_ALL_INT
  - DOO_ORDER_TXN_ATTRIBUTES_INT

### Importing New Data

If you use a file to import a customer, contact, or address that does not already exist in Trading Community Architecture, then you must manually assign the following privileges to the role that the user who runs the scheduled process named Load Interface File for Import.

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HZ_ENTER_TRADING_COMMUNITY_PERSON_PRIV</td>
<td>Enter Trading Community Person</td>
</tr>
<tr>
<td>HZ_UPDATE_TRADING_COMMUNITY_PERSON_PRIV</td>
<td>Update Trading Community Person</td>
</tr>
<tr>
<td>HZ_ENTER_TRADING_COMMUNITY_ORGANIZATION_INFO_PRIV</td>
<td>Enter Trading Community Organization Information</td>
</tr>
<tr>
<td>HZ_UPDATE_TRADING_COMMUNITY_ORGANIZATION_PRIV</td>
<td>Update Trading Community Organization</td>
</tr>
</tbody>
</table>

For details about how to add a privilege, see Data Security: Explained.

### Importing Customer Items

You must make sure you correctly set up the rank for each customer item relationship before you import source orders. The customer item relationship defines a relationship between the customer item that you import, and the item in inventory.
you use to fulfill the customer item. Order Management uses the customer item and sold-to customer details to identify the inventory item to use during order import. If you define more than one customer item relationship for a customer item, then Order Management uses the customer item relationship that specifies the lowest rank.

Note that order import will fail in the following situations:

- A customer item relationship does not exist for the customer item.
- More than one customer item relationship specifies the lowest rank.

In this example, assume you must implement the following behavior:

- Import a record that includes customer item Vision Desktop and customer Vision Operations.
- Create a relationship between Vision Desktop and inventory item AS54888 (Standard Desktop).
- Create a relationship between Vision Desktop and inventory item AS54600 (Sentinel 9000-S Desktop).
- Rank AS54888 higher than AS54600 so Order Management uses AS54888 first to fulfill the item, and then uses AS54600 only if AS54888 is not available in inventory.

Set up customer item relationships so they support order import:

1. Log into Oracle Fusion Applications with a user role that allows you to administer Product Information Management.
   If you do not log in with this user role, then the Product Information Management work area will not display your product details and will not be able to perform this procedure.
2. In the Navigator, click **Product Management**, and then click **Product Information Management**.
3. Click **Tasks**, and then click **Manage Trading Partner Items**.
4. Click **Actions**, and then click **Create**.
5. In the Create Trading Partner Type dialog, set the following value, and then click **OK**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trading Partner Type</td>
<td>Customer</td>
</tr>
</tbody>
</table>

6. On the Create Customer Item page, set the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>Vision Operations</td>
</tr>
<tr>
<td>Customer Item</td>
<td>Vision Desktop</td>
</tr>
</tbody>
</table>

7. In the Relationships area, click **Actions**, and then click **Create**.
8. In the Create Customer Item Relationship dialog, set the following values, click **OK**, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>MASTERORGANIZATION</td>
</tr>
<tr>
<td>Item</td>
<td>AS54888</td>
</tr>
<tr>
<td>Rank</td>
<td>1</td>
</tr>
</tbody>
</table>
Attribute | Value
---|---
You can enter any numeric value. Note the following points:

- It is strongly recommended that you enter a value. If you do not enter a value, then Order Management cannot identify the inventory item to use during order import, and your import will fail.
- Make sure the value you enter is unique across the customer item relationships that you define for this customer item. For example, if you define a rank of 5 for AS54888, and a rank of 5 for AS54600, then the import will fail because Order Management cannot determine which customer item relationship takes precedence.

9. In the Relationships area, click Actions, and then click Create.
10. In the Create Customer Item Relationship dialog, set the following values, click OK, and then click Save and Close.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>MASTERORGANIZATION</td>
</tr>
<tr>
<td>Item</td>
<td>AS54600</td>
</tr>
<tr>
<td>Rank</td>
<td>2</td>
</tr>
</tbody>
</table>

### Tables That the Order Import Template Contains

The Order Import Template includes a tab for each of the following interface tables.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Details That This Table Must Contain</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOO_ORDER_HEADERS_ALL_INT</td>
<td>Order header data.</td>
</tr>
<tr>
<td>DOO_ORDER_HDRS_ALL_EFF_B_INT</td>
<td>Flexfield details for the order header.</td>
</tr>
<tr>
<td>DOO_ORDER_LINES_ALL_INT</td>
<td>Order line data.</td>
</tr>
<tr>
<td>DOO_ORDER_LINES_ALL_EFF_B_INT</td>
<td>Flexfield details for the order line.</td>
</tr>
<tr>
<td>DOO_ORDER_TXN_ATTRIBUTES_INT</td>
<td>Attributes that might be associated with an order line.</td>
</tr>
<tr>
<td>DOO_ORDER_PAYMENTS_INT</td>
<td>Payment details for the order line.</td>
</tr>
<tr>
<td>DOO_ORDER_LOT_SERIAL_NUMS_INT</td>
<td>Lot serial details for an order line.</td>
</tr>
<tr>
<td>DOO_ORDER SALES CREDITS_INT</td>
<td>Sales credits for the sales representative that the source order references.</td>
</tr>
</tbody>
</table>
Using Files to Import Orders into Order Management: Procedure

This topic describes how to use files to import source orders from a source system into Order Management Cloud.

Summary of the Work

To use files to import source orders into Order Management, do the following work:

1. Convert your source data.
2. Create the import file.
3. Upload your source data.
4. Import your source data.
5. Delete imported orders from interface tables.

Order import generally uses this same flow that you use when you integrate Order Management with a source system, except you import orders from a spreadsheet. For details, see Order-to-Cash with Order Capture Systems: How it Works.

Convert Your Source Data

You must convert your source data into a CSV (comma separate values) file that uses the same structure that the Order Import Template uses. Using this template helps to make sure that your converted data uses the same structure that the Oracle database requires. This topic describes one recommended way to convert your source data. The details of your conversion might require a different way. If you cannot use the Order Import Template for some reason, and if you are not familiar with doing data conversion, then it is recommended that you consult Oracle resources about how to use an open interface when importing data. For details about other ways to import your source data into Order Management, such as through web services, see Using Public Services to Integrate with Order Management: Explained.

Convert your source data into a CSV file:

1. Download the Order Import Template from the Import Sales Orders topic in the File-Based Data Import for Oracle Supply Chain Management Cloud guide on Oracle Help Center at https://docs.oracle.com/cloud/latest/scmcs_gs/OEFS/Import_Sales_Orders_305258851_fbd1_16.htm#305258851.
2. Use a spreadsheet editor that can read a CSV file, such as Excel, to open the Order Import Template, and then familiarize yourself with the structure that it uses.
   
   For details about this structure and guidelines for using the template, see Order Import Template: Explained.
3. Use a data manipulation tool to structure your source data so that it mirrors the structure that the Order Import Template contains, and then save this data to a CSV file.

   You can use SQL (Structured Query Language), ODI (Oracle Data Integrator), or some other tool to convert your source data into a CSV file.

Guidelines for Converting Your Source Data into a CSV File
You must create a CSV file that includes data for interface tables and columns when you do the conversion. Do the following work to make sure the structure that this CSV file contains mirrors the structure that the Order Import Template contains:

- Include the same table names and column names.
- Sequence the tables in the same sequential order. The tab sequence that this template uses determines this sequential order.
- Include the same columns in each table, and arrange these columns in the same sequential order inside each table.
- Use the same data type for each column.
- Include data for all required columns. The Order Import Template uses an asterisk (*) to indicate required columns. For example, the Source Transaction Schedule Identifier column on the DOO_ORDER_LINES_ALL_INT tab is required.

If your data includes an internal identifier, then make sure you map it correctly. The Import Template includes multiple columns that specify information for internal identifiers. It uses an _ID suffix to identify these columns, such as SOURCE_TRANSACTION_ID or BUYING_PARTY_ID. For each internal identifier column, you must map your source data value to an Order Management value. If your source data includes an identifier column that does not require setup in the Product Information Management work area, Trading Community Architecture, or data collection, then you can use the implementation pages in your implementation project to get the identifier and other attributes that you must map for the internal identifier.

To view an example of this CSV structure, do the following work:

1. Open the Order Import Template in Microsoft Excel.
2. Click the DOO_ORDER_HEADERS_ALL_INT tab.
3. Choose File, Save As, and then click Excel Workbook.
4. Save the file as a .csv file, and then use a text editor to examine the output.

Create the Import File

Create the import file:

1. Prepare the Order Import Template.
   - Use a spreadsheet editor that can read a CSV file, such as Excel, to open the Order Import Template. For details, see Order Import Template: Explained.
   - Delete the example data from the Order Import Template. This template comes with example data that helps you determine the type of data that you must include. For example, row four of the DOO_ORDER_HEADERS_ALL_INT tab includes example data. Make sure you delete all example data from all tables in the spreadsheet, even from tables that you do not require.
2. Copy and paste your source data into each of the tables in the Order Import Template:
   - Use a spreadsheet editor to open the CSV file that contains your source data.
   - Copy your order line data to the clipboard.
   - In the Order Import Template, click the DOO_ORDER_LINES_ALL_INT tab.
   - Click cell A5, and then paste your data.
   - Examine the results to make sure you correctly pasted the data. For example, make sure the Product Description column contains VARCHAR data, and that the Ordered Quantity column contains NUMBER data.
   - Continue copying data for each table until you finish copying all your source data into the Order Import Template.

It is recommended that you save your work after each copy.
3. Create the import file:
   - Click the Create CSV tab, and then click Generate CSV File.
     
     If the Generate CSV File button is not active, then click Developer in the menu bar, and then click Macros. In the Macro dialog, choose GenCSV, and then click Run.
   - Wait for the macro in Excel to finish running.
     
     When the macro finishes running, Excel displays a dialog that allows you to save a .zip file.
     
     In the save dialog, choose a location to save your .zip file, and then click Save.
     
     The macro creates a .zip file that includes a separate file for each table that the template contains.
     
     Optional. In subsequent save dialogs, save each individual CSV file.
     
     If you must import data for an individual table, then save the data as an individual CSV file; otherwise, click Cancel.

Upload Your Source Data

Upload your source data:

1. Log into Order Management with administrative privileges.

2. In the Navigator, under Tools, click Scheduled Processes.


4. In the Schedule New Process dialog, set the following attribute, and then click OK.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Load Interface File for Import</td>
</tr>
</tbody>
</table>

5. In the Process Details dialog, set the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import Process</td>
<td>Choose Import Sales Orders.</td>
</tr>
</tbody>
</table>

   Data File  
   
   Do the following work:
   
   a. Click the down arrow in the Data File attribute.
   b. Scroll down, and then click Upload a New File.
   c. In the Upload File dialog, click Choose File.
   d. In your Windows Explorer window, locate and choose the .zip file that you created when you used the Order Import Template, and then click Open.
   e. In the Upload file dialog, click OK.
   f. In the Process Details dialog, make sure the Data File attribute displays the name of the .zip before you continue.

6. Click Submit.

7. In the Confirmation dialog, note the value of the Process ID, click OK, and then click Close.

8. Click Actions, and then click Refresh.

9. Use the Process ID that you noted earlier to locate your scheduled process, and then make sure the Status attribute for this process displays Succeeded.
The Succeeded status indicates that the scheduled process successfully uploaded all of your source data. If the upload fails on any row, then the status displays Error. If the Search Results list does not display your process, then click Refresh until it does.

10. Correct errors, if necessary:
    - If the scheduled process ends in an error, then click the Error status in the Search Results list for your scheduled process, and then examine the log and output files to get details about the data that caused the error.
    - Use Excel to open the Order Import Template that contains your source data, and then correct the source data that is causing the error.
    - In the Order Import Template, click Generate CSV File, and then run the scheduled process again.

Repeat this step until the scheduled process successfully uploads all of your source data.

## Import Your Source Data

Import your source data:

1. Run the Import Sales Orders scheduled process. In the Process Details dialog, set one of the following parameters. To avoid a conflict in the data to import, it is recommended that you specify only one parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch Name</td>
<td>Enter a value.</td>
</tr>
<tr>
<td>Source System</td>
<td>To import all records from the source system, enter the value that you set in the Source Transaction System column of the DOO.ORDER_LINES_ALL_INT tab of the Order Import Template. For example, enter LEG.</td>
</tr>
<tr>
<td>Source Order Number</td>
<td>To import the record for only one source order, enter the value that you set in the Source Transaction Identifier column of the DOO.ORDER_LINES_ALL_INT tab of the Order Import Template. For example, enter 12345.</td>
</tr>
</tbody>
</table>

2. Make sure the Status for the Import Sales Orders scheduled process that you submitted is Succeeded.

   The Succeeded status indicates that the scheduled process successfully imported all of your source data. If the import fails on any row, then the status displays Error.

3. Verify your import:
    - In the Navigator, click Order Management.
    - On the Order Management page, click Tasks, and then click Manage Order Orchestration Messages.
    - Query for records that the import includes.
    - Examine the errors in the spreadsheet, and then fix the source data in the Order Import Template that is causing them.
    - Submit the modified data and make sure the scheduled process successfully imports all rows.
    - Navigate to the Order Management work area, and then query for one or more of the orders that you imported.
    - Make sure the work area displays the sales order, and that the order data is identical to the source data.
Delete Imported Orders From Interface Tables

Order Management uses interface tables when you import data from a source system so that it can handle import errors and to retain a data backup during import. As an option, you can remove this data to save storage space.

Delete imported orders from interface tables:

1. In the Navigator, under Tools, click **Scheduled Processes**.
2. On the Scheduled Process page, click **Schedule New Process**.
3. In the Schedule New Process dialog, set the following attribute, and then click **OK**.

```
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Delete Orders From Interface Tables</td>
</tr>
</tbody>
</table>
```

4. In the Process Details dialog, set values, as necessary, and then click **Submit**.

**Related Topics**
- Order-to-Cash with Order Capture Systems: How it Works

How can I fix an order import that cannot find a matching configuration node?

You might encounter an error that is similar to the following when you import a source order:

**Cannot find a matching configuration node for item 12345**

This error might occur even if the node exists in the structure of a configured item that includes an instance type of Optional Single Instance. The Order Import process does a search that validates, and then creates the structure that it includes in the sales order even if a node, such as Option Classes, is missing in the order import data that defines the structure.

However, if a reference model defined as an Optional Single Instance resides in the root of the configured item, or if it is part of an option class, and if the order import data does not include the absolute path to the node, then the search will not find the underlying node.

To avoid this problem, you can modify the instance type of the structure from Optional Single Instance to Required Single Instance. You do this modification in the Product Information Management work area. It will not affect the functional behavior. You must also set the order management parameter Use Configurator for Order Import Validation to Yes.

For example, assume the Product Information Management work area includes the following structure for a configured item, and that the order import data includes M1, M1.M2, M1.M2.SI2:

```
M1
|--M2 (Optional Single Instance)
|  `--OC1
|      |--SI2
```

In this situation, the Order Import search will not find the complete structure, it will create an error during order import, and it will add an entry in the Order Import log that is similar to the following:

**The order import process failed for source order source_order_identifier for the following reason: Cannot find a matching configuration node for item item_number on order line number order_line_number.**
where:

- source_order_identifier identifies the source order
- item_number identifies the item
- order_line_number identifies the order line

This entry indicates that the search could not find a matching configure option in the node named Optional Single Instance.

For another example, assume the Product Information Management work area contains the following structure, and that the Order Import data includes M1, M1.M2, M1.M2.SI1:

```
M1
 |_OC1
 |_M2 (Optional Single Instance)
 |_SI1
```

In this situation, the Order Import search will fail in the same way that it failed in the first example.

**Exporting and Importing Setup Data with Inventory Organizations: Explained**

If you use an inventory organization when you set up Order Management Cloud, then you must import the setup data of the offerings that include your inventory organizations, and then import the setup data from Order Management to your implementation instance. You do this work when you deploy your setup data from one application instance to another application instance.

You can use one of the following options when you export or import your setup data:

- **Create one configuration package.** This package contains the offerings that include the inventory organizations, and that also contains the Order Management offering.
- **Create multiple configuration packages.** Use a separate configuration package for the offering that includes the inventory organizations, and use another package for the Order Management offering.

**Using Web Services to Import Source Orders: Explained**

An order capture service can receive a request for details about an order line from a source system that you have integrated with Order Management Cloud, and then transform it into a structure that Order Management can use.

You can use the Order Import web service to capture sales orders from a source system. For details, search for Doc ID 2051640.1 (Using Web Services with Oracle Fusion Order Management Cloud) on My Oracle Support at https://support.oracle.com. You can also use the following order capture services:

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Order Details</td>
<td>This service does the following work:</td>
</tr>
<tr>
<td></td>
<td>• Communicates status details or order details to the order capture system</td>
</tr>
<tr>
<td></td>
<td>that requests it.</td>
</tr>
<tr>
<td></td>
<td>• Provides details about the entire sales order or a set of order lines.</td>
</tr>
<tr>
<td></td>
<td>• Provides details about a sales order that originates in an integrated</td>
</tr>
<tr>
<td></td>
<td>source system.</td>
</tr>
</tbody>
</table>

164
You can specify a source system so that the query limits results to only this system. If you do not specify a source system, then the service might return order numbers from multiple source systems.

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Order Shipping Details</td>
<td>This service does the following work:</td>
</tr>
<tr>
<td></td>
<td>• Communicates the following shipping details to the order capture system that requests it:</td>
</tr>
<tr>
<td></td>
<td>◦ Current schedules</td>
</tr>
<tr>
<td></td>
<td>◦ Status of each schedule</td>
</tr>
<tr>
<td></td>
<td>◦ Schedule details, such as warehouse and shipping method</td>
</tr>
<tr>
<td></td>
<td>• Organizes the reply according to shipments instead of fulfillment lines.</td>
</tr>
<tr>
<td></td>
<td>• Includes details about fulfillment lines in the reply. Multiple fulfillment lines in Order Management might reference a single order line in the source order.</td>
</tr>
<tr>
<td></td>
<td>• Returns details about the order lines of a sales order, or for only a subset of these lines.</td>
</tr>
<tr>
<td>Apply a Hold</td>
<td>Routes a request for a hold from an order capture system to the Hold task layer. Processes multiple holds in a request, each hold for a single sales order, or for multiple order lines in this sales order.</td>
</tr>
<tr>
<td>Release a Hold</td>
<td>Routes a request to release an existing hold from an order capture system to the Hold task layer.</td>
</tr>
<tr>
<td>Check Availability</td>
<td>Sends a request to Order Management to determine the quantity that is available on a date at a source system. Order Management sends a reply to the order capture system that includes these details.</td>
</tr>
</tbody>
</table>

Transforming Source Orders to Sales Orders

Transforming Source Orders to Sales Orders: How It Works

Order Management Cloud transforms each source order that you create in Order Management or that it receives from a source system so that it can optimize order fulfillment.
The following diagram illustrates how Order Management transforms a source order.

![Diagram of Order Management transformation process]

**Explanation of Callouts**

In this example, Order Management does the following work:

1. Receives a source order that you submit in Order Management or that it gets from a source system. The source order in this example includes three order lines.
2. Separates the source order into logical pieces that it can fulfill, such as fulfillment lines.
3. Assigns each fulfillment line to a new orchestration process that it creates to orchestrate fulfillment for this fulfillment line. For example, it might assign fulfillment line 1-1 to orchestration process 300100090333189, and fulfillment line 1-2 to orchestration process 300100090333210.
4. Starts each orchestration process that performs the tasks that fulfill each fulfillment line. To examine these orchestration processes, you can use the Switch to Fulfillment View action on the Order page of the Order Management work area.

Order Management typically finishes transformation within a few seconds after you click Submit.

Order Management sets the completion date of the final step in each orchestration process to the requested date of the source order. Note that the Last Fulfillment Completion Step specifies the final step of the orchestration process. This final step is not necessarily the last step that Order Management completes in chronological order.
Parts of Sales Orders You Can Use After Order Management Transforms Source Orders

You can use the following parts of a sales order after Order Management transforms a source order. You can use them to monitor and manage order fulfillment.

<table>
<thead>
<tr>
<th>Part of Sales Order</th>
<th>Description</th>
</tr>
</thead>
</table>
| Order               | Includes order lines and fulfillment lines. Note the following:  
  • The relationship that exists between a sales order and a fulfillment line is similar to the relationship that exists between a source order and a source order line.  
  • A sales order that resides in Order Management might not exactly mirror the structure of a source order that resides in an external source system. |
| Order line          | Typically references one fulfillment line. Note the following:  
  • If Order Management splits a fulfillment line, then it maps two or more fulfillment lines to the same order line.  
  • The relationship that exists between a sales order and the order lines preserves some of the original structure of the source order, making it easier to keep track of the original intent of the source order, even if Order Management splits the fulfillment lines that it associates with the sales order. |
| Fulfillment line    | Part of a sales order that you can modify in the Order Management work area. For example, you can do the following modifications:  
  • Schedule or unschedule a fulfillment line.  
  • Reserve items for a fulfillment line.  
  • Substitute an item on a fulfillment line.  
  • Change the warehouse, shipping method, or demand class for a fulfillment line. |
| Orchestration process | An order administrator can set up an orchestration process to determine the tasks that this process must perform. You cannot modify an orchestration process in the Order Management work area, but you can monitor an instance of it while it manages each fulfillment line. You can monitor each fulfillment line as it progresses through the orchestration process. |
| Return fulfillment line | Represents a sales order line that Order Management uses to return items. It is similar to a fulfillment line, but it includes a set of attributes that are specific to returning items. |

Related Topics

- Transforming Order Lines to Fulfillment Lines: How it Works
- Creating Sales Orders: Procedure
- Managing Sales Orders: Overview
- Processing Change Orders: How it Works

Transformation Rules: Explained

Order Management Cloud allows you to write rules that populate order attributes before it transforms a source order, rules that transform a source order, and rules that populate order attributes after it transforms a source order.
Pretransformation Rules

A pretransformation rule populates an order attribute before Order Management transforms the source order. You can use the predefined attribute value that the product transformation rules already contain. Order Management defaults the master inventory organization to the sales order so that you can use it in a transformation rule. Consider the following examples.

<table>
<thead>
<tr>
<th>Example Usage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Populate an attribute on a fulfillment line.</td>
<td>Assume your company receives orders for widgets that include an attribute named Request Date, and that this attribute must display on all fulfillment lines for widgets. You can write a pretransformation rule that implements the following logic:</td>
</tr>
<tr>
<td></td>
<td>• If the item is a widget, then populate the Request Date attribute on the fulfillment line.</td>
</tr>
<tr>
<td>Populate an attribute so you can use it in a transformation rule.</td>
<td>Assume your company receives orders for widgets, and that you plan to write a transformation rule that converts the widget size from centimeters to inches. You must first populate the Size attribute in the fulfillment line. You can write a pretransformation rule that implements the following logic:</td>
</tr>
<tr>
<td></td>
<td>• If the item is a widget, then populate the Size attribute in the fulfillment line.</td>
</tr>
</tbody>
</table>

Transformation Rules

Order Management uses item relationships, item structures, transactional attributes, and business rules to transform a sales representation of the item that a source order contains to a fulfillment representation of the item that a sales order in Order Management contains. You can use the Manage Product Transformation Rules page in the Setup and Maintenance work area to write the transformation rules that this topic describes. You can also use Oracle Fusion Product Model to define item relationships, item structures, and transactional attributes. However, you cannot use a product transformation rule to add a product model to a sales order.

Product-to-Product Transformation

A product-to-product transformation transforms a single item to one or more items according to item structure, item relationship, and the transformation rule. Assume your company receives orders for a laptop that comes with multiple accessories, such as a docking station, mouse, and so on. You can write a transformation rule that transforms an item to another item, and that creates individual fulfillment lines. In the following example, a product-to-product transformation rule
translates a source order for a laptop and accessory package into a sales order that includes individual fulfillment lines for the laptop and for each item that the accessory package contains.

### Source Order

<table>
<thead>
<tr>
<th>Line</th>
<th>Quantity</th>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Laptop</td>
<td>700 USD</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Accessory Package</td>
<td>225 USD</td>
</tr>
</tbody>
</table>

### Sales Order

<table>
<thead>
<tr>
<th>Product</th>
<th>Transformed Order Line</th>
<th>Fulfillment Line</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop</td>
<td>1</td>
<td>1-1</td>
<td>1</td>
</tr>
<tr>
<td>Accessory Package</td>
<td>2</td>
<td>2-1</td>
<td>1</td>
</tr>
<tr>
<td>Docking Station</td>
<td>2.1</td>
<td>2.1-1</td>
<td>1</td>
</tr>
<tr>
<td>Mouse</td>
<td>2.2</td>
<td>2.1-2</td>
<td>1</td>
</tr>
<tr>
<td>AC Adapter</td>
<td>2.3</td>
<td>2.1-3</td>
<td>1</td>
</tr>
<tr>
<td>Keyboard</td>
<td>2.4</td>
<td>2.1-4</td>
<td>1</td>
</tr>
</tbody>
</table>

Order Management does the following work to transform the source order:

- Transforms two source order lines into six fulfillment lines.
- Uses the price from the source order to populate the price in the sales order.
- Transforms the line for the laptop in the source order to fulfillment line 1 in the sales order.
- Transforms the Accessory Package in the source order to multiple fulfillment lines in the sales order. Each line represents part of the content of the Accessory Package, such as docking station, mouse, AC adapter, and so on.

### Product-to-Attribute Transformation

A product-to-attribute transformation translates the item that a source order references according to the attributes of another item. Assume your company receives orders for a Window item. You can write a transformation rule that uses the Width attribute, Height attribute, Pane attribute, and Glass attribute of the source order to get the item number for an item that uses the same dimensions, and that uses double-pane, tempered glass. The transformed order includes the item number for this combination of transactional attributes.
### Attribute-to-Product Transformation

An attribute-to-product transformation uses transactional attributes to transform an attribute to an item number. It can add the transformation to an existing item or replace the item that the source order references. Assume your company receives orders for an MP3 player that includes a Color attribute and a Size attribute, and that you must use a combination of these attributes to reference an item number. In the following example, a transformation rule transforms the Color and Size attributes of the Mini Plus item to the VIS481 item. Order Management displays VIS481 on the order line.
**Source Order**

<table>
<thead>
<tr>
<th>Line</th>
<th>Quantity</th>
<th>Product</th>
<th>Transactional Attributes</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Mini Plus</td>
<td>Color: Silver Size: 8 MB</td>
<td>$100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Engraving: For Maria</td>
<td></td>
</tr>
</tbody>
</table>

**Transformation**

<table>
<thead>
<tr>
<th>Color</th>
<th>4 MB Size</th>
<th>8 MB Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silver</td>
<td>Product: VIS451A</td>
<td>Product: VIS481A</td>
</tr>
<tr>
<td>Black</td>
<td>Product: VIS452A</td>
<td>Product: VIS482A</td>
</tr>
<tr>
<td>Pink</td>
<td>Product: VIS453A</td>
<td>Product: VIS483A</td>
</tr>
<tr>
<td>Blue</td>
<td>Product: VIS454A</td>
<td>Product: VIS484A</td>
</tr>
</tbody>
</table>

**Sales Order**

<table>
<thead>
<tr>
<th>Line</th>
<th>Quantity</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>VIS481A</td>
</tr>
</tbody>
</table>

**Attribute-to-Attribute Transformation**

An attribute-to-attribute transformation transforms the value of a transactional attribute on an order line in a source order to a different transactional attribute on an order line in a sales order. Assume your company resides in Europe, it receives orders from an office in the United States that measures the item size in inches, and that you require Order Management to display the item size in centimeters. In the following example, a transformation rule transforms the width and height from inches on the source order to centimeters on the sales order.
**Context-to-Product Transformation**

A context-to-product transformation uses the context of the source order to determine the item that the sales order must reference. Assume your company receives orders for laptop computers that you must ship to different geographical regions. Each region requires a different electrical adapter, such as 110 volts or 240 volts. In the following example, the transformation rule uses the geographical region where you ship the item to determine the adapter to include with the sales order, transforms the source order to a sales order that includes two order lines, and then adds the adapter to one of these lines.
Source Order

<table>
<thead>
<tr>
<th>Line</th>
<th>Quantity</th>
<th>Product</th>
<th>Transactional Attributes</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Vision Pro Laptop</td>
<td>550 Vision Way, Boston, MA 02108</td>
<td>$1500</td>
</tr>
</tbody>
</table>

Transformation

Europe or Asia
- Product: 240VAC adaptor

North America
- Product: 120VAC adaptor

Sales Order

<table>
<thead>
<tr>
<th>Line</th>
<th>Quantity</th>
<th>Product</th>
<th>Ship To</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Vision Pro Laptop</td>
<td>550 Vision Way, Boston, MA 02108</td>
<td>$1500</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Product: 120VAC adaptor</td>
<td>550 Vision Way, Boston, MA 02108</td>
<td>0</td>
</tr>
</tbody>
</table>

Context-to-Attribute Transformation

A context-to-attribute transformation transforms the source order context to a transactional attribute. Assume your company manufactures laptop computers. Assume it ships some of them to domestic locations that reside in the USA, ships some of them to international locations that reside in other countries, and that the destination location requires different packaging. In the following example, the value of the transactional attribute in the source order includes a domestic address, so the context is domestic, and the transformation rule sets the Domestic Packaging transactional attribute to Yes.
### Source Order

<table>
<thead>
<tr>
<th>Line</th>
<th>Quantity</th>
<th>Product</th>
<th>Transactional Attributes</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Vision Pro Laptop</td>
<td>550 Vision Way, Boston, MA 02108</td>
<td>$1500</td>
</tr>
</tbody>
</table>

### Transformation

<table>
<thead>
<tr>
<th>Ship International</th>
<th>Ship to USA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domestic Packaging: No</td>
<td>Domestic Packaging: Yes</td>
</tr>
</tbody>
</table>

### Sales Order

<table>
<thead>
<tr>
<th>Line</th>
<th>Quantity</th>
<th>Product</th>
<th>Transactional Attributes</th>
<th>Ship To</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Vision Pro Laptop</td>
<td>Domestic Packaging: Yes</td>
<td>550 Vision Way, Boston, MA 02108</td>
</tr>
</tbody>
</table>

### Posttransformation Rules

A posttransformation rule populates an order attribute after Order Management transforms the item. Consider the following examples.
Example Usage | Description
--- | ---
Populate order lines so they reference different warehouses. | Assume your company receives orders for laptop computers, and that you created a transformation rule that transforms the source order into a sales order that includes the following lines:  
- Order line 1 for the laptop computer  
- Order line 2 for an AC adapter

You can write a posttransformation rule that populates order line 2 so that it references a warehouse that is different from the warehouse that supplies the laptop computer.

Populate an order with a new attribute. | Assume your company receives orders that use a MM/DD/YYYY format for the requested date. Your staff finds it useful to also know the day of the week because delivery costs more on Saturday and Sunday. You write a posttransformation rule that populates the day of the week in the new sales order.

Transforming Source Orders from Source Systems: How It Works

Order Management Cloud can support one or more source systems. The source orders that your source system captures probably includes details and uses a structure that is different from the details and structure that Order Management uses. If you use multiple source systems, then the details that each system captures on each source order might be different in each system. To manage and fulfill these source orders, Order Management transforms them into a consistent structure.

Order Management uses the following sequence to transform each source order that it receives from a source system:

1. Order Management receives a source order from a a source system.
2. The connector service structurally transforms the source order to a sales order that Order Management can use.
3. The connector service calls the Receive and Transform service.
4. The Receive and Transform service (SalesOrderOrchestrationEBS), looks up the cross-reference values to determine whether or not it must transform the source order values. It looks up these values in the following sources:
   - Oracle Product Model
   - Oracle Trading Community Model, which is a customer data hub
   - Order Orchestration and Planning Data Repository

   Cross-referencing is required for static data, such as country code and currency codes, and for dynamic data, such as customers and items. For details, see Cross-Referencing: Explained.
5. If the source system and Order Management use different domain values, then the connector service transforms the structure and values from the domain that the source system uses to the domain that Order Management uses.
6. The connector service calls a process that separates the source order into parts.
7. Another service calls the requested operation for the source order, such as create, update, delete, or cancel.
8. Order Management transforms the item according to the following transformation rules:
   - Pretransformation rule
   - Transformation rule
   - Posttransformation rule
   - Process assignment rule

   You can create a transformation rule that specifies how to do transformation and to make sure the Order Management work area correctly displays the source order. For details, see Transformation Rules: Explained.
9. The Assign and Launch service assigns an orchestration process to each line item according to the rules. Order Management uses the following attributes to calculate the planned dates for each step and task, starting with the first step that it performs in chronological order:
   - Default Lead Time
   - Lead Time UOM
   - Lead-Time Expression

Order Management can use an order capture service to communicate updates to the source system. To receive these updates, the source system must subscribe to the events. If Order Management receives an update from the source system, then it replans the orchestration process. It does this replanning every time it receives an update.

Order Management does not perform the same amount of transformation for a sales order that a user creates in the Order Management work area because these orders already include the data and use the structure that Order Management requires.

### Setting Up Transformation: Procedure

You must set up transformation to make sure that Order Management Cloud correctly transforms each source order. Set up transformation:

1. Use the Oracle Product Model to do the following work:
   - Use the product master to define the product that the transformation rule references.
   - If you plan to use a transformation rule that references the product structure, then define product structures.
   - If you plan to use a transformation rule that references transactional item attributes, then define transactional item attributes.
   - Define the relationship between sales product and fulfillment product.

   You must create item substitution rules when you define items. Order Management does not validate item substitutions. For example, assume a fulfillment line includes a nonconfigured item, the Order Manager examines availability for this item, and then order promising displays more than one option. If you do not set up each item substitution rule correctly, then the Order Manager might select an option that uses an incorrect substitution, such as in a kit.

2. Use the Manage Product Transformation Rules page to create transformation rules. For details, see Creating Transformation Rules: Procedure.

### Creating Transformation Rules: Procedure

This topic describes how to define a transformation rule that uses a bucket set in a decision table. For details about how to use a simplified rule builder, see Using Visual Information Builder to Create Rules: Explained.

Assume you work for a company that uses priority shipping for each sales order that includes a Green Server. You can create the following rule to meet this requirement:

- If the item is a Green Server, then use priority shipping.

You will create a bucket set that contains the values you will select when you create your rule, and then create a transformation rule that represents this offer.
It is strongly recommended that you get details about decision tables, bucket sets, and how to use them. Search for document ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, download the attachment that describes how to use business rules, and then see the chapter that describes decision tables and bucket sets.

Summary of the Work

Create transformation rules:

1. Create the bucket set list.
2. Create the decision table and add a condition.
3. Add a rule to the decision table.
4. Add the action to take if the condition is true.

This topic includes example values. You might use different values, depending on your business requirements.

Create the Bucket Set List

You will create the following bucket set list:

Create the bucket set list:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup page, click Order Management.
3. Search for, and then open Manage Product Transformation Rules.
4. On the Manage Product Transformation Rules page, click Bucket sets, click Add Bucket set, and then click List of Values.
5. In the Bucket sets list, click Bucket Set 1, and then click Edit Bucket set.
6. In the Bucket set Editor dialog, set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Server IDs</td>
</tr>
<tr>
<td>Description</td>
<td>List of IDs and aliases for server items to select in a product transformation rule.</td>
</tr>
<tr>
<td>Data Type</td>
<td>String</td>
</tr>
</tbody>
</table>

7. Click Add Bucket, and then set the following values in the Bucket 1 row of the Bucket Values list.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>300000047394016L</td>
</tr>
<tr>
<td></td>
<td>This number identifies the item ID that the database contains. The letter L indicates a long value.</td>
</tr>
<tr>
<td>Alias</td>
<td>“AS85005”</td>
</tr>
<tr>
<td></td>
<td>You must include the double quotation marks.</td>
</tr>
<tr>
<td>Description</td>
<td>Green Server 3500R</td>
</tr>
</tbody>
</table>

8. Click Add Bucket, and then set the following values in the Bucket 2 row of the Bucket Values list.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>300000047393961L</td>
</tr>
<tr>
<td>Alias</td>
<td>“AS85004”</td>
</tr>
<tr>
<td>Description</td>
<td>Green Server 3000</td>
</tr>
</tbody>
</table>

9. Click OK, and then click Save.
Create the Decision Table and Add a Condition

You will create the following decision table, condition, and action:

Create the decision table and add a condition:

1. On the Manage Product Transformation Rules page, click TransformationCustomRS.
2. In the View list, click IF/THEN Rules, click Add, and then click Add Decision Table.
3. Replace Decision Table 1 with AddExpeditedShippingforProductAS85005, and then click Save.
4. Immediately above the decision table, click Add, and then click Add Condition.
5. In the Condition Browser dialog, expand OrderTransformationRules, expand FulfillLineVO, expand InventoryItemId, click toString, and then click OK.

The Condition Browser dialog displays objects from the OrderTransformationRules dictionary. You can use the FulfillLineVO object in this dictionary to specify transformation according to the value of a fulfillment line attribute, such as InventoryItemId. In this example, you use toString to get the value of InventoryItemId as a string so you can compare it in the rule.

Add a Rule to the Decision Table

You add the following rule in this section:

- If InventoryItemId is AS85005

Add a rule to the decision table:

1. In the decision table, click the cell that displays the OrderTransformationRules.FulfillLineVO.InventoryItemId.toString() condition.
For example:

2. Click **Local List of Values**, and then click **Server IDs**, which is the bucket set that you created.
3. In the `OrderTransformationRules.FulfillLineVO.InventoryItemId.toString()` row, double-click the **cell** in the R1 column, single-click the **cell** in the R1 column, and then add a check mark to **AS85005**.

**Add the Action to Take If the Condition Is True**

You add the following action in this section. At run time, order fulfillment will recognize that `ShipClassOfService` is Expedited, and then expedite the shipment:

- Set the `ShipClassOfService` attribute to "Expedited"

Add the action to take if the condition is true:

1. Immediately above the decision table, click **Add**, click **Add Action**, and then click **Modify**.
2. In the Action Editor dialog, in the Target list, click `OrderTransformationRules.FulfillLineVO`.
3. In the Arguments list, locate the `ShipClassOfService` argument, set the following value, and then click **OK**. It might be necessary to scroll or page down through the Arguments list.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameterized</td>
<td>Contains a check mark.</td>
</tr>
<tr>
<td></td>
<td>The Parameterized property makes the object available to the business rule as a parameter.</td>
</tr>
</tbody>
</table>

4. In the decision table, in the `ShipClassOfService` row, double-click the cell in the R1 column, and then enter "Expedited". You must include the double quotation marks.
5. Click **Save**.

**Related Topics**

- Using Visual Information Builder to Create Rules: Explained

**Creating Advanced Product Transformation Rules: Procedure**

This topic describes how to create an advanced transformation rule that can compare two or more lines in a source order. As an example, it uses a rule that removes multiple requests that cancel each other.
Assume you notice that in some situations your deployment includes a fulfillment line that adds an inventory item, but then includes another fulfillment line that deletes this same inventory item. To avoid processing these lines, you can create the following transformation rule:

- If fulfillment line a requests to add inventory item x, and if fulfillment line b requests to delete inventory item x, then delete fulfillment lines a and b.

This topic implements the following logic.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First IF statement</td>
<td>If the change in fulfillment line 1 is Add.</td>
</tr>
<tr>
<td>Second IF statement</td>
<td>If the change in fulfillment line 2 is Delete.</td>
</tr>
<tr>
<td>Third IF statement</td>
<td>If the inventory item in fulfillment line 1 is the same as the inventory item in fulfillment line 2.</td>
</tr>
<tr>
<td>Fourth IF statement</td>
<td>If the fulfillment line ID of fulfillment line 1 is different from the fulfillment ID of fulfillment line 2.</td>
</tr>
<tr>
<td>THEN statement</td>
<td>Delete fulfillment line 1 and fulfillment line 2.</td>
</tr>
</tbody>
</table>
You will create the following transformation rule.

This topic uses Advanced Mode. It is strongly recommended that you familiarize yourself with this mode and with creating business rules. For details, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, download the attachment that describes how to use business rules, and then examine the chapter that describes using advanced mode.

Summary of the Work
To create an advanced transformation rule, do the following work:

1. Create the rule.
2. Create the first IF statement.
3. Create the second IF statement.
4. Create the third IF statement.
5. Create the fourth IF statement.
6. Create the THEN statement.

This topic includes example values. You might use different values, depending on your business requirements.

Create the Rule

Create the rule:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Product Transformation Rules.
4. On the Manage Product Transformation Rules page, in the View list, click **IF/THEN Rules**.
5. Click **Add**, and then click **Add Rule**.
6. Click **Expand**, and then click **Show Advanced Settings**.
7. Set the following values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1</td>
<td>Consolidate Add and Delete Actions</td>
</tr>
<tr>
<td>Description</td>
<td>Rule that removes requests that cancel each other.</td>
</tr>
<tr>
<td>Advanced Mode</td>
<td>Contains a check mark.</td>
</tr>
</tbody>
</table>

8. Click **Save**.

Create the First IF Statement

You will create the following IF statement. It determines whether the change in fulfillment line 1 is Add.

Create the first IF statement:

1. In the field to the left of **Is A**, enter **FulfillLine**.
2. In the field to the right of **Is A**, click the **down arrow**, and then click **OrderTransformationRules.FulfillLineVO**.
3. Click **Add Test**, and then click **Simple Test**.
4. Click **Left Value**.
5. In the Condition Browser, expand **FulfillLine**, click **DeltaType**, and then click **OK**.
6. In the Right Value field, enter “Add”. You must include the double quotation marks.
7. Click **Save**.
Create the Second IF Statement

You will create the following IF statement. It determines whether the change in fulfillment line 2 is Delete.

Create the second IF statement:

1. Click **Add Pattern**.
2. In the window below **And**, enter **FulfillLine2**.
3. In the field to the right of **Is A**, click the **down arrow**, and then click **OrderTransformationRules.FulfillLineVO**.
4. Click **Add Test**, click **Simple Test**, and then click **Left Value**.
5. In the Condition Browser, expand **FulfillLine2**, click **DeltaTime**, and then click **OK**.
6. In the Right Value field, enter **"Delete"**. You must include the double quotation marks.

Click **Save**.

Create the Third IF Statement

You will create the following IF statement. It determines whether the inventory item in fulfillment line 1 is the same as the inventory item in fulfillment line 2.

Create the third IF statement:

1. Click **Add Pattern**.
2. In the window below **And**, enter **FulfillLine**.
3. In the field to the right of **Is A**, click the **down arrow**, and then click **OrderTransformationRules.FulfillLineVO**.
4. Click **Add Test**, click **Simple Test**, and then click **Left Value**.
5. In the Condition Browser, expand **FulfillLine**, click **InventoryItemld**, and then click **OK**.
6. Click **Right Value**.
7. In the Condition Browser, expand **FulfillLine2**, click **InventoryItemld**, and then click **OK**.

Click **Save**.
Create the Fourth IF Statement

You will create the following IF statement. It determines whether the fulfillment line ID of fulfillment line 1 is different from the fulfillment ID of fulfillment line 2.

Create the fourth IF statement:

1. Click Add Pattern.
2. In the window below And, enter FulfillLine.
3. In the field to the right of Is A, click the down arrow, and then click OrderTransformationRules.FulfillLineVO.
4. Click Add Test, click Simple Test, and then click Left Value.
5. In the Condition Browser, expand FulfillLine, click InventoryItemId, and then click OK.
6. Click Is, and then click Isn't.
7. Click Right Value.
8. In the Condition Browser, expand FulfillLine2, click InventoryItemId, and then click OK.

Click Save.

Create the Then Statement

You will create the following action. It deletes fulfillment line 1 and fulfillment line 2.

Create the action:

1. In the Then area, click Add Action, and then click Assert New.
2. Click Select a Target, and then click OrderTransformationRules.DeleteOrderLine.
3. Click Edit Properties.
4. In the Properties dialog, enter the following values, and then click OK

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>fulfillmentLineld</td>
<td>fulfillmentLineld</td>
</tr>
<tr>
<td>viewRowImpl</td>
<td>fulfillmentLineld</td>
</tr>
</tbody>
</table>

5. Click Add Action, and then click Retract.
6. Click **Select a Target**, and then click **FulfillLine2**.

Click **Save**.

**Related Topics**

- Creating Business Rules: Demo

---

### Cross-Referencing in Order Management: Explained

You must create and maintain cross-references that relate business data between order capture systems, order fulfillment systems, and Order Management Cloud.

You can use the following types of cross-references.

<table>
<thead>
<tr>
<th>Type of Cross-Reference</th>
<th>Description</th>
</tr>
</thead>
</table>
| Customer Cross-Reference | The following work occurs:  
  1. The source capture system sends a source order to Order Management that includes customer data.  
  2. If the customer already exists in the Oracle customer master, then Order Management uses a customer cross-reference to get the master customer record and the customer ID of the order fulfillment system.  
  3. Order Management sends the sales order, including the customer ID and other attributes that it gets from the Oracle customer master, to the order fulfillment system.  
  
  Note the following:  
  - The Oracle Trading Community Model maintains the customer cross-references. You can use an external customer hub with Order Management, but you must maintain cross-references in this model.  
  - You can create customer cross-references in the Oracle customer model when you create the customer information when you configure your implementation. This configuration allows Order Management to resolve the Oracle customer values. |
| Item Cross-Reference | An item cross-reference creates the following relationships:  
  - **Source system.** A relationship between the source item and the Oracle item when your deployment uses a product hub.  
  - **Named item.** A relationship between the source item and the item when your deployment brings items from disparate systems into the Oracle Product Model, which is a repository that stores master product information. |
| Other Cross-Reference | The Order Orchestration and Product Data Repository stores the cross-references for all attributes that are not customer attributes or item attributes. You can use a domain value map for attributes that reside in the Order Orchestration and Planning Data Repository. The collections process uses these maps to map values from one domain to another domain, which is useful when different domains represent the same data in different ways. |
5 Controlling Order Processing

Controlling Application Behavior

Managing Order Management Parameters: Procedure

You can use the Manage Order Management Parameters page to manage parameters that affect most or all of Order Management Cloud.

Manage Order Management parameters:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Order Management Parameters.
4. Set the following parameters.

<table>
<thead>
<tr>
<th>Order Management Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow Changes Through Configurator Validation</td>
<td>Allows the Configurator to choose items in a configuration and to modify configuration choices after the Order Entry Specialist adds a configured item to a sales order, and then save the sales order as a draft, but before the Order Entry Specialist submits the sales order.</td>
</tr>
</tbody>
</table>

For example, assume the following occurs:

1. A user adds a configured item to a sales order, and then saves this sales order as a draft.
2. A configuration manager modifies the configuration model and the configuration rules for this configured item in such a way that it affects the configuration that the Order Entry Specialist specified.
3. The Order Entry Specialist returns to the draft several days later, and then submits the sales order.

In this situation, if you set Allow Changes Through Configurator Validation to:

- **Yes**, The configurator uses the modified configuration model and configuration rules to modify the configuration that the Order Entry Specialist specified. It allows the submit to proceed. It does not report these modifications to the Order Entry Specialist. This setting is useful when your business process does not require the Order Entry Specialist to understand and agree to the modifications that the configurator makes.
- **No**, The configurator does not modify the configuration that the Order Entry Specialist specified. Instead, it does not allow the submit to proceed, and displays a validation error when the Order Entry Specialist attempts to submit or validate the sales order. This setting is useful when your business process requires that the Order Entry Specialist understand and agree to the modifications that the configurator makes.

Note the following:

- This parameter does not apply to a source order that includes a configured item. For details about importing source orders, see Order Import Template: Explained.
- This parameter affects all business units.

For details about the configurator, see the guide titled Oracle SCM Cloud, Configurator Modeling.
<table>
<thead>
<tr>
<th>Order Management Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Effective Date</td>
<td>Specify the date that Order Management uses to determine the configure options that it displays for a configured item. The configure options it displays might vary depending on the date you specify. You can choose one of the following values:</td>
</tr>
<tr>
<td>Ordered Date</td>
<td>Use the configure options that the Configurator defined according to the value that the Ordered Date field displays on the Create Order page.</td>
</tr>
<tr>
<td>Configuration Date</td>
<td>Use the configure options that the Configurator defined when you configured the item.</td>
</tr>
<tr>
<td>Current Date</td>
<td>Use the configure options that the Configurator defined as of today.</td>
</tr>
<tr>
<td>Requested Date</td>
<td>Use the configure options that the Configurator defined as of the requested date.</td>
</tr>
<tr>
<td>If you do not specify the Configuration Effective Date, then Order Management uses the Current Date.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customer Relationship Type</th>
<th>Specify how Order Management displays the Ship-to Address and the Bill-to Location for each customer during order entry. Set the Business Unit field to one of the following values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A single business unit, such as Vision Services</td>
<td>Allow the Order Entry Specialist to select only the bill-to account or the ship-to address that is associated with the sold-to customer. For example, assume you set the Business Unit to Vision Services and the Customer Relationship Type to Yes. If the Order Entry Specialist creates a sales order for XYZ Company, then Order Management allows the Order Entry Specialist to select only the bill-to accounts and the ship-to addresses that XYZ Company references.</td>
</tr>
<tr>
<td>All Business Units</td>
<td>Allow the Order Entry Specialist to select any bill-to account or ship-to address for the business unit, regardless of the sold-to customer. For example, assume you set the Business Unit to Vision Services and the Customer Relationship Type to Yes. If the Order Entry Specialist creates a sales order for XYZ Company, then Order Management displays the bill-to account that XYZ Company references, by default, and it also allows the Order Entry Specialist to search and select any bill-to account for this sales order.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Halt Configurator Validation on First Error</th>
<th>Specify one of the following values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>Stop processing on the first error that the Configurator encounters during order entry.</td>
</tr>
<tr>
<td>False</td>
<td>Do not stop processing if the Configurator encounters an error during order entry. Instead, allow the Configurator to continue to run until it finishes processing. This setting allows the Configurator to identify and report all errors that the configuration contains.</td>
</tr>
</tbody>
</table>

Halt Configurator Validation on First Error affects only order entry. It does not affect order import.

<table>
<thead>
<tr>
<th>Item Validation Organization</th>
<th>Specify the item validation organization that Order Management uses to validate and display items for each sales order the Order Entry Specialist creates, according to business unit. Order Management will display only the items that it associates with the item validation organization you specify.</th>
</tr>
</thead>
<tbody>
<tr>
<td>For example, assume you create a record in the Item Validation Organization list, you set the Business Unit for this record to Vision Operations, and you set the Organization to Denver Manufacturing. Order Management will display only the items that it associates with Denver Manufacturing when the Order Entry Specialist creates a sales order for Vision Operations. It displays these items when the Order Entry Specialist searches for items on the Order Lines tab of the Create Order page.</td>
<td></td>
</tr>
<tr>
<td>Note the following:</td>
<td></td>
</tr>
<tr>
<td>You must specify an item validation organization for each business unit. If you do not do this, then Order Management disables search for the item in each sales order that references...</td>
<td></td>
</tr>
<tr>
<td>Order Management Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>a business unit that you do not specify. As an alternative, if you use the same master organization for every business unit, then you can specify this organization in the Item Validation Organization parameter, and then set the Business Unit for this parameter to All Business Units.</td>
</tr>
<tr>
<td></td>
<td>• You can set the Item Validation Organization only for business units that Order Management associates with your login responsibility. If it associates your login responsibility with only one business unit, then it uses this business unit as the item validation organization.</td>
</tr>
<tr>
<td></td>
<td>• The Product Information Management work area associates inventory organizations with business units. If you do not set the Item Validation Organization, and if Product Information Management associates multiple inventory organizations with the same business unit, then Order Management does one of the following:</td>
</tr>
<tr>
<td></td>
<td>◦ If Product Information Management associates the same item master organization with these inventory organizations, then Order Management sets the Item Validation Organization to this item master organization, and then informs the Order Entry Specialist of this setting. The Order Entry Specialist can accept this value and continue entering the sales order, or reject it and contact the Order Administrator with a request to use the Item Validation Organization parameter to set up the inventory organization.</td>
</tr>
<tr>
<td></td>
<td>◦ If Product Information Management associates different item master organizations with these inventory organizations, then Order Management informs the Order Entry Specialist that the Order Administrator must first set up the inventory organization before the Order Entry Specialist can enter the sales order.</td>
</tr>
<tr>
<td></td>
<td>• Order Management displays only inventory organizations in the Organization field that the Product Information Management associates with the business unit that you choose in the Business Unit field. You can use the Product Information Management to set up an inventory organization and create these associations. For more information, see Inventory Organizations: Explained.</td>
</tr>
<tr>
<td>Preparer for Procurement</td>
<td>Specify an Order Management user who the buyer can contact to help resolve a problem that might occur with a fulfillment line that involves a drop ship supplier. The buyer is a procurement application user. Order Management sends this information to purchasing.</td>
</tr>
<tr>
<td></td>
<td>If you do not specify the Preparer for Procurement parameter, then the procurement system will reply with an error that the preparer is missing.</td>
</tr>
<tr>
<td></td>
<td>You can define a preparer for each selling business unit. You can also define a preparer as the default value for all business units. Order Management will use this default value only for business units where you do not define a preparer.</td>
</tr>
<tr>
<td></td>
<td>For details about drop ship, see Drop Ship: Overview.</td>
</tr>
<tr>
<td>Use Configurator for Order Import Validation</td>
<td>Specify one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• True. The Configurator validates each source order that includes a configured item during order import.</td>
</tr>
<tr>
<td></td>
<td>• False. The Configurator does not validate each source order that includes a configured item during order import.</td>
</tr>
<tr>
<td>Activate Credit Check on Order Submit</td>
<td>Specify whether or not to run credit check when the Order Entry Specialist submits the sales order. For details, see Managing Credit Check in Order Management: How it Works.</td>
</tr>
<tr>
<td>Credit Check Failure at Order Submit</td>
<td>Specify how to proceed if credit check fails. You can specify one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• Save Order in Draft Status. Save the sales order in Draft status and do not proceed to fulfillment.</td>
</tr>
<tr>
<td></td>
<td>• Submit the Order with Hold on Lines That Failed Credit Check. Save the sales order in Submitted status, place a hold on each order line that failed credit check, and then proceed with fulfillment for the order lines that are not on hold.</td>
</tr>
</tbody>
</table>
Check for Trade Compliance When User Submits Sales Order

Specify one of the following values:

- **Yes.** Order Management will verify each sales order that the Order Entry Specialist submits in Order Management or that you import from a source system.
- **No.** Order Management will not verify any sales orders.

For details, see Managing Trade Compliance: Overview.

From Address for Email Messages

Specify the From email address that the Universal Messaging Service uses when it sends an email from the order document or from an automatic notification. The From address comes predefined as no-reply@orderreport.com. You can change this value to use an address in the domain that your company uses. You must use a valid email address format. For example, something@somethingelse.xxx. For details, see Administering Email Formats in Order Management: Explained.

Notify Frequency

Specify an integer that represents the number of hours to wait before consolidating, and then sending notifications. It comes predefined with a value of 1. You can set it to any value that is greater than or equal to 0 (zero). If you set it to 0, then Order Management will not consolidate notifications, and will send each notification when the event that it references occurs. For details, see Sending Notifications from Order Management Cloud to External Systems: Explained.

Start Approval Process for Sales Orders

Specify whether or not to get sales order approval. Note the following points:

- If you must enable approval, then you must set this parameter to Yes even if you must enable approval only for Order Management.
- If you set this parameter to No, then Order Management will not route sales orders for approval, even if you define and activate an approval rule.
- If you set this parameter to Yes but you do not define and activate an approval rule, then Order Management will not route the sales order for approval, but will instead send it to order fulfillment.

For details about how Order Management uses this parameter, see Managing Order Approval Rules: How it Works.

Related Topics

- Product Master Data Management Features: Overview
- Inventory Organizations: Explained
- Administering Email Formats in Order Management: Explained
- Managing Trade Compliance: Overview

Managing Order Profiles: Procedure

Order Management Cloud uses predefined profile values for each item. Some of these values control behavior in the Order Management work area. Some values control how Order Management receives and transforms source orders into sales orders. Most profiles include predefined values so you do not need to configure them, unless your organization requires different values.
Manage order profiles:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Order Profiles.
4. On the Manage Order Profiles page, in the Profile Option area, click **Search**.
5. In the Search Results area, in the Profile Options list, click the **profile** that you must edit.
6. In the Profile Values list, add or delete values for the following attributes, as necessary.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Currency Conversion Type</td>
<td>Specifies the value to use when converting a currency in the Order Management work area. This value is a conversion type. You can use this setting for an implementation across a single site and for each user.</td>
</tr>
<tr>
<td>Display Currency</td>
<td>Specifies the currency to display in the Order Management work area. You can use this setting for an implementation across a single site and for each user.</td>
</tr>
<tr>
<td>Required Overview Status Filter</td>
<td>Specifies the default customer to use when filtering the summary of status data on the Overview page of the Order Management work area. It allows you to view summary data for only one customer at a time. It removes the All option. Order Management provides no value, by default. To improve performance, you can enter a customer identification number. You can use this setting only for an implementation across a single site.</td>
</tr>
<tr>
<td>Retain Sales Order Number</td>
<td>Specifies to use the source order number as the order number during transformation. The default value is N. You can use this setting for an implementation across a single site and for each user.</td>
</tr>
<tr>
<td>User Request Waiting Period in Seconds</td>
<td>Specifies the number of seconds to wait after an action finishes. This time allows asynchronous services to complete before displaying a confirmation message or a warning message in the Order Management work area. The default value is 5. You can use this setting only for an implementation across a single site.</td>
</tr>
</tbody>
</table>

**Controlling Decimal Precision: Procedure**

You can use the Quantity Decimal Precision profile to control decimal precision for attributes that store numeric values. For example, to control decimal precision on the order line quantity that the Order Entry Specialist sets on the Create Order page, or to control decimal precision during order import.

**Precision** is the total number of digits that a number contains, where **scale** specifies the number of digits that exist to the right of the decimal point. For example, the number 17.347 uses a precision of 5 and a scale of 3. You cannot modify precision but you can modify scale. For example, if you set the scale to 3 in the Quantity Decimal Precision profile, and if the Order Entry Specialist enters 17.3468 for the quantity, then the profile will round 17.3468 to 17.347.

Note that the quantity for some items is typically a whole number. For example, most companies sell a desktop computer as a whole item. You do not use Quantity Decimal Precision to allow only a whole number in the quantity. Instead, you can set the Indivisible attribute to Yes in the Product Information Management work area when you set up the item.

Assume the following for the example that you will set up in this topic:

- You sell a variety of elements from the periodic table according to weight, such as Diamond.
- You sell these elements in grams and milligrams, which is 1/1000 of a gram.
The value for milligram can range anywhere from .001 to .999.

You already set up an item named Diamond in the Product Information Management work area, and you specified it to use Gram as the unit of measure.

You will set the scale to 3.

Set up decimal precision:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Tasks**, and then click **Search**.
3. On the Search page, search for, and then open Manage Administrator Profile Values.
4. On the Manage Administrator Profile Values page, in the Search area, enter the following value, and then click **Search**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Display Name</td>
<td>Quantity Decimal Precision</td>
</tr>
</tbody>
</table>

5. In the Search Results, Profile Values area, set the following value, click **Save and Close**, and then click **Done**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Value</td>
<td>3</td>
</tr>
</tbody>
</table>

Note that you can set decimal precision only at the Site level. You cannot set it for each item or for each unit of measure.

6. In the Navigator, click **Order Management**.
7. On the Overview page, click **Create Order**.
8. On the Create Order page, in the Select Item window, enter **Diamond**, and then click **Search**.
9. On the catalog search line, in the quantity window to the left of Gram, set the value to 17.3468, and then click **Add**.
10. Verify that the Create Order page displays a message that it rounded your value to 17.347.

### Managing Statuses for Sales Orders

#### Order Management Statuses: Explained

Order Management Cloud displays a separate status for the sales order, fulfillment line, fulfillment task, and orchestration process. Each status indicates the progress of a sales order from beginning to completion. The statuses of the fulfillment lines, orchestration processes, and fulfillment task determines the sales order status. The Order Management work area displays these status values.

Order Management sequentially evaluates each of the status conditions that a fulfillment line contains while it processes a sales order. The true condition that contains the highest sequence number determines the fulfillment line status. Order Management then uses the status of the fulfillment line that has progressed the furthest in the sales order life cycle to update the order status.
Where Order Management Displays Status

Order Management displays status in the following locations.

<table>
<thead>
<tr>
<th>Status</th>
<th>Where Displayed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Status</td>
<td>At the top of the Order page in the order title. For example, the order status of the following sales order is Processing: Order: Computer Service and Rentals - 282079 - Processing</td>
</tr>
<tr>
<td>Fulfillment Line Status</td>
<td>In the Order Lines area of the Order page. For example, the status of the following fulfillment line is Awaiting Shipping: Item, AS54888 - Standard Desktop, Status, Awaiting Shipping</td>
</tr>
<tr>
<td>Orchestration Process Status</td>
<td>In the Status field of the Orchestration Process page. For example, Awaiting Fulfillment.</td>
</tr>
<tr>
<td>Fulfillment Task Status</td>
<td>In the Status column of the Orchestration Plan tab of the Orchestration Process page. For example, Awaiting Fulfillment.</td>
</tr>
</tbody>
</table>

Sales Order Status

The status that Order Management displays for the sales order depends on the following statuses of the fulfillment line.

<table>
<thead>
<tr>
<th>Sales Order Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft</td>
<td>You saved the order to the database but have not submitted to order fulfillment. You can modify a Draft order.</td>
</tr>
<tr>
<td>Processing</td>
<td>You submitted the sales order to order fulfillment, but Order Management has not delivered all fulfillment lines to the customer. You cannot modify a Processing order, but you can revise it. For details, see Revising Sales Orders That You Already Submitted: Procedure</td>
</tr>
<tr>
<td>Open</td>
<td>Order Management has not completed any fulfillment lines.</td>
</tr>
<tr>
<td>Partial</td>
<td>Order Management has completed some, but not all, fulfillment lines.</td>
</tr>
<tr>
<td>Reference</td>
<td>An earlier version of a revised order. For example, if you revise an existing sales order, and then click Submit, then Order Management displays the earlier version of the sales order with a status of Reference.</td>
</tr>
<tr>
<td>Closed</td>
<td>Order Management completed all fulfillment lines. You cannot modify a Closed order, but you can create a return order to return items from it. For details, see Returning Sales Orders: Procedure</td>
</tr>
<tr>
<td>Canceled</td>
<td>Order Management canceled all fulfillment lines. It also removed the sales order from order fulfillment and did not reschedule it. You cannot submit a Canceled sales order to order fulfillment.</td>
</tr>
</tbody>
</table>
Implementing Order Management

Chapter 5

Controlling Order Processing

Sales Order Status | Description
---|---
| | If Order Management canceled only some fulfillment lines, then it ignores the canceled fulfillment lines, and sets the status according to the fulfillment lines that are open.

Approval Pending | Order Management sent your sales order to a sales order approver. For details, see Getting Approvals for Sales Orders: Explained

Order Line Status

The status that Order Management displays for the order line depends on the following statuses of the fulfillment lines.

<table>
<thead>
<tr>
<th>Order Line Status</th>
<th>Fulfillment Line Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Order Management has not completed any fulfillment lines.</td>
</tr>
<tr>
<td>Partial</td>
<td>Order Management has completed some, but not all, fulfillment lines.</td>
</tr>
<tr>
<td>Closed</td>
<td>Order Management completed all fulfillment lines.</td>
</tr>
<tr>
<td>Canceled</td>
<td>Order Management canceled all fulfillment lines. If Order Management canceled only some fulfillment lines, then it ignores the canceled fulfillment lines, and sets the status according to the fulfillment lines that are open.</td>
</tr>
</tbody>
</table>

Related Topics

- Transforming Order Lines to Fulfillment Lines: How it Works
- Revising Sales Orders That You Already Submitted: Procedure

Orchestration Process Status: Explained

An orchestration process status indicates the status of an orchestration process throughout Order Management Cloud. The value Started is an example of this status. It indicates that Order Management has started the orchestration process.

Order Management completes the fulfillment tasks of an orchestration process sequentially according to each orchestration process step when it processes an order.

Order Management comes predefined to use a default set of statuses for fulfillment tasks. You can also set up statuses and sequences. For example, you can set up an orchestration process that uses a set of statuses and rule logic for textbooks for a college, and set up another orchestration process that uses a different set of statuses and rule logic for textbooks for a primary school.

If you set up an orchestration process status, then note the following points:

- You must specify the status that Order Management assigns to an orchestration process at each orchestration process step. For example, if a Schedule Carpet task includes a status of Unsourced, then you must specify the status that Order Management assigns to the orchestration process for this step.
• You must specify a status that indicates when a task is complete. You can select only a status that you define to indicate that a task is complete.

• If you modify the name of the default status, then Order Management displays this modified name throughout the Order Management work area.

• If you do not set up statuses for an orchestration process, then Order Management uses the predefined statuses, by default.

**Status Conditions for Orchestration Processes**

You can create a status condition that determines the orchestration process status. For example, you can create the following condition:

• If the status of the Schedule task is Not Started, then set the orchestration process status to Unscheduled.

Order Management evaluates the status conditions that you create sequentially at run time. The condition that evaluates to true, and that includes the highest sequence number, determines the orchestration process status.

**How Order Management Assigns Statuses**

Order Management uses the following sequence when it assigns statuses:

1. A fulfillment system sends a status update to Order Management.
2. The External Interface Layer translates the status into status values that Order Management uses.
3. The Task Layer determines the status message to send.
4. The Status Service does the following work:
   - Uses source and target mapping to set the task status.
   - Sets the orchestration process status according to the statuses of the tasks that this process contains.
   - Sets the fulfillment line status according to the task statuses that the fulfillment line references.

**Related Topics**

• [Defining Orchestration Processes: Procedure](#)

**Managing Status Values: Explained**

You can use the following tabs on the Manage Status Values page to manage how Order Management Cloud displays statuses.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status Codes</td>
<td>Specify the display name that Order Management displays in the Order Management work area.</td>
</tr>
<tr>
<td>Fulfillment Lines</td>
<td>Create the status values that users can select for a fulfillment line in an orchestration process.</td>
</tr>
<tr>
<td>Task Types</td>
<td>Assign a status code to a task type. If an external system provides the status, then assign this status to the fulfillment task that references this system. For example, if a shipping system provides the status, then assign the code to the Shipping task type.</td>
</tr>
<tr>
<td>Orchestration Process Classes</td>
<td>Assign the status code to fulfillment lines or orchestration processes where Order Management must use the status. For example, you can use an orchestration process class to control the status</td>
</tr>
</tbody>
</table>
Tab | Description
---|---
codes that an administrator can choose when creating a status condition for an orchestration process. You must use the Manage Status Values page to define the status values and to make them available when you create a status condition.

### Splitting Fulfillment Lines: Explained

Order Management Cloud can split a fulfillment line into multiple fulfillment lines to improve order fulfillment performance. You can control some of this behavior.

Note the following behavior that occurs when Order Management splits a fulfillment line.

<table>
<thead>
<tr>
<th>Split</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual split</td>
<td>Note the following:</td>
</tr>
<tr>
<td></td>
<td>• You can use Split Fulfillment Line only for a sales order that is on a manual schedule task. For details, see Splitting Fulfillment Lines: Procedure.</td>
</tr>
<tr>
<td></td>
<td>• To split a fulfillment line that is not on a manual schedule task, you can click <strong>Unschedule</strong> while the orchestration process is running, and then split the fulfillment line. Unschedule moves the orchestration process back to the schedule task, which now is a manual task. For details, see Fixing Problems and Improving Fulfillment Performance: Procedure.</td>
</tr>
<tr>
<td></td>
<td>• Order Management does not run change management for a manual split.</td>
</tr>
<tr>
<td></td>
<td>• Although you cannot use the Check Availability page to manually split a fulfillment line, the availability option that you select might implicitly split the fulfillment line.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Automatic split</th>
<th>Order Management might automatically split a fulfillment line in the following situations:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• If you select a fulfillment option on the Check Availability page, then this option might require Order Management to split a fulfillment line so that it can provide the availability results that you require.</td>
</tr>
<tr>
<td></td>
<td>• Some other fulfillment system might automatically split a fulfillment line.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Splits that Order Management does not allow</th>
<th>Order Management does not allow you to split fulfillment line in the following situations:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• The fulfillment line does not include the Splits Allowed attribute.</td>
</tr>
<tr>
<td></td>
<td>• The quantity of the fulfillment line is one or less.</td>
</tr>
<tr>
<td></td>
<td>• The fulfillment line does not allow a split. If you use an override privilege, then you can override this restriction.</td>
</tr>
<tr>
<td></td>
<td>• A task in the parallel branch of an orchestration process references the fulfillment line.</td>
</tr>
<tr>
<td></td>
<td>• The fulfillment line resides in a shipment set. To split this line, you can remove it from the shipment set, and then split it.</td>
</tr>
</tbody>
</table>

### How Order Management Determines Availability When it Splits Fulfillment Lines

If a fulfillment line allows a split or substitution, then Order Management examines the following possible substitutions when it determines availability.
Where Split Occurs | Description
--- | ---
Split a fulfillment line across multiple warehouses | If the Requested Ship-from Warehouse attribute on the fulfillment line is:
- Empty, Order Management considers multiple warehouses when it splits a fulfillment line.
- Not empty, Order Management considers only the warehouse that the Requested Ship-from Warehouse references. It does not split the fulfillment line.

Each fulfillment line that the split creates might specify a different value for the Expected Ship-from Warehouse attribute.

Order Management uses business rules that your order administrator sets up to determine the warehouses it uses to supply the item.

Split a fulfillment line across substitute items | Order Management substitutes an item only if the Allow Substitutions attribute on the fulfillment line equals Yes. Each fulfillment line that the split creates might specify a different value for the Available Item attribute.

For example, assume you create a fulfillment line that requests 100 units of an item. Assume supply for this item is 80 units and supply for the substitute item is 50 units. Check Availability will split the fulfillment line into the following fulfillment lines:
- One fulfillment line that references 80 units of the item
- Another fulfillment line that references 20 units of the substitute item

Split a fulfillment line across dates | If a fulfillment line requests inventory that is not sufficient on a given date, then Order Management splits this line across the following dates:
- Creates one fulfillment line for the quantity that is available on the requested date.
- Creates another fulfillment line for the remaining quantity that it will deliver on a later date.

For example, assume you create a fulfillment line on November 15, 2016 that requests 100 units of an item, and that supply is available for this item on the following dates:
- 70 units are available on November 15, 2016.
- 40 units are available on November 30, 2016.

Check Availability will split the fulfillment line into the following fulfillment lines:
- One fulfillment line with on-time delivery of 70 units, with an expected delivery date of November 15, 2016.
- Another fulfillment line with delayed delivery for 30 units, with an expected delivery date of November 30, 2016.

How Split Fulfillment Lines Affect Status
Order Management Cloud creates two or more instances of the same fulfillment task when it splits a fulfillment line. These tasks might include different statuses during processing.

Consider the following examples:
- Assume the status of the Schedule task for fulfillment line A1 is Not Scheduled, and the status of the Schedule task for fulfillment line A2 is Scheduled. In this example, Order Management examines the split priority of the task statuses, and then sets the orchestration process status to the task status that includes the highest split priority.
• Assume Order Management splits a fulfillment line that results in two instances of the Schedule task. One of these tasks includes a Complete status, and the other task includes a Pending status. Assume the value of the split priority for Pending is two, and the value of the split priority for Complete is three. In this example, Order Management sets the orchestration process status to Pending.

Related Topics

• Splitting Fulfillment Lines: Procedure
• Fixing Problems and Improving Fulfillment Performance: Procedure
• Transforming Order Lines to Fulfillment Lines: How it Works
• Shipping Order Lines in Shipment Sets: Procedure

Managing Task Status Conditions: Procedure

Order Management Cloud fulfills each fulfillment task one step at a time when it processes a sales order, and it uses a predefined set of sequential statuses to track the progress of each task. You can use the Manage Task Status Conditions page to manage the status conditions that determine the status of these tasks according to each task type.

Manage task status conditions:

1. On the Manage Task Status Conditions page, in the task list, click the task type that you must manage.

   For example, to manage the status conditions when Order Management processes a return, click the row that contains Return in the Type column.

2. In the Status Conditions list, modify or add new conditions as necessary. Note the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Status Value</td>
<td>Displays a status that Order Management receives from a fulfillment system through the task layer. For details, see Task Layer: Explained</td>
</tr>
<tr>
<td>Display Status Value</td>
<td>Specifies the value that Order Management displays. For example, assume the fulfillment system sends a status of Invoiced for a task, but your company uses Billed. You can set Display Status Value to Billed to display Billed throughout the Order Management work area.</td>
</tr>
<tr>
<td>Mark as Complete</td>
<td>If this option contains a check mark, then Order Management considers the task to be complete when it reaches the condition. For example, assume you click the Return row in the task list on the Manage Task Status Conditions page, and then, in the Status Conditions area, you add a check mark to Mark as Complete for the Canceled status condition and for the Delivered status condition. In this situation, Order Management considers the task to be complete when the task reaches the Canceled status or the Delivered status.</td>
</tr>
</tbody>
</table>

Related Topics

• Task Layer: Explained
Setting Up Fulfillment Line Statuses: Explained

Order Management Cloud fulfills each fulfillment task when it processes the orchestration process step that is associated with a fulfillment line. It uses these statuses to represent the fulfillment line status throughout Order Management. You can specify the status that it assigns to this fulfillment line at each step when you define an orchestration process. For example, you can specify that if a Schedule Carpet task includes a status of Pending Scheduling, then set the fulfillment line status to Unscheduled.

You can specify different sets of statuses and rules for different items that a fulfillment line references. For example, you can specify one set of status conditions for a textbook, and another set of status conditions for a paperback book. If you do not create these conditions, then Order Management uses the status rule set that it assigns to the default category.

Using Status Catalogs and Status Categories with Fulfillment Lines

Your organization might require each fulfillment line that an orchestration process references to use a different status. For example, an item that includes shippable fulfillment lines might require statuses that are different from the statuses that an item that includes nonshippable fulfillment lines uses. A status catalog allows you to group items that are similar so that they can achieve the same statuses at the same time.

You can select a status catalog when you define an orchestration process only if this catalog meets the following requirements:

- An item exists in only one category in the catalog.
- The category contains items or subcategories, but not items and subcategories.
- Order Management controls the catalog only for the master, and not for each organization.

You can use catalogs and categories in multiple orchestration process definitions. Use a category to make sure that Order Management applies the same set of status conditions to specific sets of fulfillment lines. It applies the same status conditions to all fulfillment lines that reference the item that resides in this category.

You can use the Oracle Product Model to set up a status catalog.

Using Status Rule Sets with Fulfillment Lines

The status rule set allows you to use a single set of rules with more than one fulfillment line instead of specifying separate rules for each fulfillment line. For example, you can specify the following rules:

- If an item includes a status of Unsourced, then set the fulfillment line status to Unscheduled.
- If the Schedule Carpet fulfillment task reaches a status of Completed, then set the fulfillment line status to Scheduled.

Note the following points:

- You can use a single status rule set with multiple categories. If a parent category and a child category each reference a different status rule set, then Order Management uses the status rule set that the child references. This capability allows you to define an All category to handle all items in one orchestration process. It also allows you to add a subcategory for a subset of items that must use a different status rule set.
- If you use your implementation project to migrate an orchestration process instance from a development environment to a production environment, then do not modify the status rule set name in either environment. Modifying this name might prevent Order Management from updating references to other data in the orchestration process.
Defining the Status Condition Sequence

It is recommended that you define the sequence of status conditions so they match the logical progress of the status that the order line normally takes during the order line lifecycle. If more than one condition evaluates to true, then the orchestration process will set the fulfillment line status to the status value with the highest sequence number.

The following example illustrates how you can use the Sequence attribute in the Orchestration Process Status Values area on the Status Conditions tab of the Create Orchestration Process Definition page. You set the sequence of the status conditions so they match the logical progress of the status. For example, Sequence 150 for status Awaiting Shipping occurs before Sequence 160, status Shipped. If Awaiting Shipping and Shipped each evaluate to true, then the orchestration process will set the status to Shipped because 160 is higher than 150.

Setting Up Fulfillment Line Statuses: Procedure

You can use the Manage Status Values page to set up status values that Order Management Cloud displays for a fulfillment line.

Summary of the Work

Set up status values for fulfillment lines:

1. Add the status codes.
2. Add status codes to fulfillment lines.
This topic includes example values. You might use different values, depending on your business requirements.

### Add the Status Codes

Add the status codes:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Status Values.
4. On the Manage Status Values page, in the Status Codes list, click **Actions**, and then click **Create**.
5. In the Create Status Code dialog, enter the following values, and then click **Save and Create Another**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>SCHED_GOODS</td>
</tr>
<tr>
<td>Name</td>
<td>Scheduled Goods</td>
</tr>
</tbody>
</table>

6. In the Create Status Code dialog, enter the following values, and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>SCHED_CLOTHING</td>
</tr>
<tr>
<td>Name</td>
<td>Scheduled Clothing</td>
</tr>
</tbody>
</table>

### Add Status Codes to Fulfillment Lines

Add the status codes that you created to the fulfillment lines:

1. On the Manage Status Values page, click **Fulfillment Lines**, click **Actions**, and then click **Select and Add**.
2. In the Select and Add dialog, in the Status Code field, enter **SCHED_GOODS**, and then click **Search**.
3. Click **SCHED_GOODS**, and then click **OK**.
4. Click **Actions**, and then click **Select and Add**.
5. In the Select and Add dialog, in the Status Code field, enter **SCHED_CLOTHING**, and then click **Search**.
6. Click **SCHED_CLOTHING**, and then click **OK**.
7. Click **Save and Close**.

You can now set the status value for a fulfillment line in an orchestration process to SCHED_GOODS or SCHED_CLOTHING.

### Creating Orchestration Process Classes: Procedure

You can use the orchestration process class as a way to group orchestration process statuses so that they are meaningful in an orchestration process. For example, the Ship Order Class includes statuses that are meaningful when shipping an order, such as Scheduled, Awaiting Shipping, and Shipped.

This topic includes example values. You might use different values, depending on your business requirements.

Create an orchestration process class:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Status Values.
4. On the Manage Status Values page, click **Orchestration Process Classes**.
5. In the Orchestration Process Classes list, click **Actions**, click **Create**, and then set the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Standard_Class</td>
</tr>
<tr>
<td>Name</td>
<td>Standard_Class</td>
</tr>
<tr>
<td>Description</td>
<td>Class for standard orders</td>
</tr>
</tbody>
</table>

6. Click **Save**.
7. In the Status Values area, click **Actions**, click **Add Row**, and then add the following value.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status Value</td>
<td>Scheduled</td>
</tr>
</tbody>
</table>

8. Repeat step 6 for each of the following values:
   - Shipped
   - Reserved
   - Billed
   - Returned to Customer

9. Click **Save and Close**.

You can now set the Standard_Class class in the Process Class field on the Edit Orchestration Process Definitions page. The class that you set determines the values that you can select when you set the Status Value of the Status Conditions tab on this page. For example, if you set the Process Class to Standard_Class, then the Status Value field allows you to select the statuses that the Standard_Class class references, such as Shipped.

**Related Topics**
- Defining Orchestration Processes: Procedure

**Adding Status Conditions to Orchestration Processes: Procedure**

You can add status conditions that specify when to set the status of an orchestration process. Assume your company requires an orchestration process that can fulfill orders for company t-shirts, and that you must specify the statuses that this process uses throughout the order life cycle according to the status of the fulfillment task. You can create the condition that determines the orchestration process status.

In this topic, you create the following example status condition:

- If the status of the Schedule task is Scheduled, then set the orchestration process status to Scheduled.
Summary of the Work
To add a status condition to an orchestration process, do the following work:

- Set the orchestration process class.
- Add the status condition.

You typically add more than one status condition to an orchestration process. For brevity, in this topic you add only one status condition.

This topic includes example values. You might use different values, depending on your business requirements.

Set the Orchestration Process Class
Set the orchestration process class:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup page, click Order Management.
3. Search for, and then open Manage Orchestration Process Definitions.
4. On the Manage Orchestration Process Definitions page, locate the CallCustomerWhenLargeInvoice orchestration process, click Actions, and then click Edit.

   For details about how to create this process, see Adding Branches to Orchestration Processes: Procedure.
5. On the Edit Orchestration Process Definitions page, set the following value.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Class</td>
<td>Standard_Class</td>
</tr>
</tbody>
</table>

   You must define Standard_Class before you can select it. For details about how to define it, see Creating Orchestration Process Classes: Procedure.

6. Click Save.

Add the Status Condition
Add the status condition:

1. In the Process Details area, click Status Conditions.
2. In the Orchestration Process Status Values list, click Actions, and then click Add Row.
3. In the new row, set the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>1</td>
</tr>
</tbody>
</table>
   
   Status Value | Scheduled

4. Add the expression:
   - In the new row, in the Expression column, click the icon.
   - In the Expression Builder dialog, click CallCustomerWhenLargeInvoice - Schedule, and then click Insert Into Expression. Make sure you do not expand CallCustomerWhenLargeInvoice - Schedule.
Notice that the dialog added a value of “Schedule” in the Expression window of the Expression Builder.

- In the Expression window, click anywhere after “Schedule”, and then enter an equal sign (=).
- Expand CallCustomerWhenLargeInvoice - Schedule, click SCHEDULED, and then click Insert Into Expression.
- Notice that the Expression window contains “Schedule”="SCHEDULED", and then click OK.

5. Click Save.

Related Topics
- Adding Branches to Orchestration Processes: Procedure

Adding Status Conditions to Fulfillment Lines: Procedure

You can add status conditions to a fulfillment line that includes multiple items, and where each of these items requires a different status. If different fulfillment lines must use different statuses, then you must determine how you will use catalogs and categories to group these fulfillment lines. You do this work when you create an orchestration process.

In this topic, assume you must set up orchestration processes that can handle orders for different types of t-shirts. You could use the same orchestration process for multiple types of merchandise, but you prefer to define statuses for each type of clothing separately because each clothing type requires a different status. To do this, you select the status catalog, and then add the status conditions for a single category of items in an orchestration process.

Summary of the Work

To add status conditions to a fulfillment line, do the following work:

- Set the status catalog.
- Add the status condition for each default.
- Add the status condition for an item.

You typically create multiple status conditions for a fulfillment line. For brevity, in this topic you create only one status condition for the default category and another status condition for an item.

This topic includes example values. You might use different values, depending on your business requirements.

Set the Status Catalog

Set the status catalog:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup page, click Order Management.
3. Search for, and then open Manage Orchestration Process Definitions.
4. On the Manage Orchestration Process Definitions page, locate the CallCustomerWhenLargeInvoice orchestration process, click Actions, and then click Edit.

   For this example, you edit the CallCustomerWhenLargeInvoice orchestration process. For details about how to create this process, see Adding Branches to Orchestration Processes: Procedure.
5. On the Edit Orchestration Process Definitions page, set the following value.
Add the Default Status Condition

Add the status condition that this orchestration process will use, by default:

1. In the Process Details area, click **Status Conditions**, and then click **Fulfillment Line Status Values**.
2. In the Default row, in the Status Rule Set field, click the arrow, and then click **Create**.
   
   You will select the statuses that this orchestration process uses, by default. Every row in the Fulfillment Line Status Values list must reference a status rule set, and each rule set must contain at least one status condition.

3. In the Create Status Rule Set dialog, set the following values, and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>LargeOrders</td>
</tr>
<tr>
<td>Name</td>
<td>Large Orders</td>
</tr>
<tr>
<td>Create New</td>
<td>Chosen</td>
</tr>
</tbody>
</table>

4. Click **Save**.
5. Click **Edit Status Rule Set**.
6. On the Edit Status Rule Set page, click **Actions**, and then click **Add Row**.
7. In the Sequence field, enter 1.
8. Set the status value:
   - In the Status Value field, click the arrow, and then click **Search**.
   - In the Search and Select dialog, in the Status Code field, enter `SCHED`, and then click **Search**.
   - In the list, click **Scheduled**, and then click **OK**.

9. Add the expression:
   - On the Edit Status Rule Set page, in the Expression column, click the icon.
   - In the Expression Builder dialog, click **CallCustomerWhenLargeInvoice - Schedule**, and then click **Insert Into Expression**. Make sure you do not expand `CallCustomerWhenLargeInvoice - Schedule`. Notice that the dialog added a value of "Schedule" in the Expression window of the Expression Builder.
   - In the Expression window, click anywhere after "Schedule", and then enter an equal sign (=).
   - Expand `CallCustomerWhenLargeInvoice - Schedule`, click `SCHEDULED`, and then click **Insert Into Expression**.
10. On the Edit Status Rule Set page, add a check mark to Notify External Systems, and then click `Save and Close`.

Notify External Systems allows Order Management to communicate the status to an external system. For details, see `Sending Notifications from Order Management Cloud to External Systems: Procedure`.

**Add the Status Condition for an Item**

Add the status condition for an item:

1. In the Process Details area, in the Fulfillment Line Status Values list, click `Actions`, and then click `Select and Add: Category`.
2. In the Select and Add: Category dialog, select `Retailer`, and then click `Save and Close`.
3. In the Fulfillment Line Status Values list, in the Retailer row, click the `arrow` in the Status Rule Set column, and then click `Create`.
4. In the Create Status Rule Set dialog, set the following values, and then click `Save and Close`.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Shirts</td>
</tr>
<tr>
<td>Name</td>
<td>Shirts</td>
</tr>
<tr>
<td>Create New</td>
<td>Chosen</td>
</tr>
</tbody>
</table>

5. Click `Save`.
6. In the Fulfillment Line Status Values list, in the Retailer row, click `Edit Status Rule Set`.
7. Repeat steps 6 through 10 of the Add the Default Status Condition topic, above.

**Related Topics**

- Adding Branches to Orchestration Processes: Procedure

**Closing Fulfillment Lines That Remain Open: Explained**

Order Management Cloud closes a fulfillment line when it finishes all steps of the orchestration process. In some situations, an orchestration process step might indicate that fulfillment is complete, such as shipping complete for a shipping step. This indicator might mean that your organization considers fulfillment complete when the orchestration process reaches this step, but the fulfillment line is still open.

If the scheduled process named Update or Close Sales Orders did not run since Order Management closed the fulfillment line, then Order Management might display sales orders and order lines as open even if it closed all the fulfillment lines that these orders and lines reference. To fix this situation, you can run this scheduled process in the Scheduled Processes work area.

**Constraining Sales Orders**
Managing Processing Constraints: Explained

If an Order Entry Specialist attempts to submit or change a sales order, order line, or fulfillment line, and if a processing constraint does not allow this submit or change, then Order Management Cloud rejects it and displays a message. If a source system attempts to submit a sales order or make a change, then Order Management rejects it and sends a return message. Order Management also uses the processing constraint to make sure each fulfillment request includes the attributes that it requires.

The processing constraint uses the following logic:

- If the validation rule set that Order Management applies to a record set is true, then apply the processing constraint to the constraint entity.

The processing constraint includes the following parts.

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>The job role that the constraint does not allow to make a change. For example, you can constrain an order manager from changing a sales order when the orchestration process proceeds beyond a step that you specify.</td>
</tr>
<tr>
<td>Action</td>
<td>The action that the constraint does not allow. You can constrain the user from performing the following actions:</td>
</tr>
<tr>
<td></td>
<td>• Create</td>
</tr>
<tr>
<td></td>
<td>• Validate</td>
</tr>
<tr>
<td></td>
<td>• Update</td>
</tr>
<tr>
<td></td>
<td>• Split</td>
</tr>
<tr>
<td></td>
<td>• Submit</td>
</tr>
<tr>
<td></td>
<td>• Cancel</td>
</tr>
<tr>
<td></td>
<td>• Delete</td>
</tr>
<tr>
<td>Condition</td>
<td>The condition that the constraint evaluates to determine whether or not to apply the constraint. For example, you can create a condition that applies a constraint when a sales order is booked.</td>
</tr>
</tbody>
</table>

Note the following points:

- You can use the Manage Processing Constraints page to manage a predefined constraint, such as a predefined record set, validation rule set, or constraint entity. You can also set up your own constraint.
- If you set up a processing constraint that does not use a condition, then the constraint is always true. For example, the predefined processing constraint that prevents the Order Entry Specialist from deleting a sales order prevents deletion in all situations.
- You can use a processing constraint with an extensible flexfield. For details, see Constraining Changes That Users Make in Extensible Flexfields: Procedure.

Objects That You Specify for Processing Constraints

In general, you specify the following objects when you define a processing constraint.
<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation rule set</td>
<td>A group of one or more if statements. For example, you can use the following rule to reject a change on a closed order:</td>
</tr>
<tr>
<td></td>
<td>• If the order is closed, then reject the change.</td>
</tr>
<tr>
<td></td>
<td>In this example, a predefined validation rule set named Order is Closed examines the Open attribute on the order header to determine whether or not the value is N.</td>
</tr>
<tr>
<td></td>
<td>Note the following:</td>
</tr>
<tr>
<td></td>
<td>• You cannot modify or delete a predefined validation rule set, but you can create a new one.</td>
</tr>
<tr>
<td></td>
<td>• You can apply a processing constraint when the condition is true or is not true.</td>
</tr>
<tr>
<td></td>
<td>• You cannot enter a value that contains ~ (a tilde) when you define a validation rule set because Order Management uses the tilde as a delimiter when it evaluates the constraint rule at run time. You might also encounter this problem at run time with some order attributes. For example, a shipping instruction might contain a tilde.</td>
</tr>
<tr>
<td>Record set</td>
<td>A set of records that Order Management groups according to common attribute values so that it can evaluate a constraint. For example, to evaluate all orders for a customer, you can specify to evaluate one of the following entities when you create a record set:</td>
</tr>
</tbody>
</table>
|                   | • Order header  
|                   | • order line  
|                   | • Order fulfillment line                                                        |
|                   | You can then select an attribute to refine the record set. For example, to evaluate all orders for a customer, select the Order Header entity, and then select the Sold-to Customer attribute. |
| Constraint entity | The business object or orchestration process that a processing constraint constrains. For example, an order header, or an attribute of an order header, such as the Latest Acceptable Ship Date field that the order header displays. For details, see Managing Constraint Entities: Explained. |
| Constraint package | A set of triggers that Order Management applies to a table in an Oracle database. A background process sets these triggers when you create a constraint package. |
|                   | A constraint package can activate a new or modified:                                                                                  |
|                   | • Validation rule set that is of a table type  
|                   | • Record set for a processing constraint                                                                                               |
|                   | To create a constraint package, you can do the following:                                                                            |
|                   | • Use the Manage Processing Constraints page or run the Generate Constraint Packages scheduled process.                                 |
|                   | • Create or modify a record set or a validation rule set that is of a table type.                                                       |
|                   | You do not need to create a constraint package for any validation that is not of a table type.                                          |

For example, the following processing constraint prevents a change to any order fulfillment line for a specific customer that is part of an orchestration process that is in the shipping stage:
Object | Example
---|---
Validation rule set | If the Status Code attribute of a sales order is equal to Awaiting Shipment. . .

Record set | . . .for the customer that the Bill-to customer ID attribute of the order fulfillment line identifies. . .

Processing constraint | . . .then do not allow an update. . .

Constraint entity | . . .on any of these order fulfillment lines.

Modifying Constraint Names

You can use one of the following text strings in the constraint name. For example, to indicate payment, use DOO_PAYMENT_EXCEPTION. Note that the text style in the text string applies with any case-sensitivity, such as GTM, Gtm, gtm, gTM, and so on. The text string can occur anywhere in the constraint name.

<table>
<thead>
<tr>
<th>Text String</th>
<th>Constraint That Occurs During Validation With. . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTM</td>
<td>Oracle Global Trade Management.</td>
</tr>
<tr>
<td>PAYMENT</td>
<td>Payment.</td>
</tr>
<tr>
<td>PRICING</td>
<td>Pricing.</td>
</tr>
<tr>
<td>Constraint name does not include GTM, PAYMENT, or PRICING</td>
<td>Screen for all types during order validation.</td>
</tr>
</tbody>
</table>

Managing Processing Constraints: Examples

This topic includes examples of some of the ways that you can use a processing constraint.

Reject Changes at the Shipping Stage

Assume an orchestration process achieves the shipping stage for a sales order, and a user submits a change for this sales order. The shipping stage occurs late in the orchestration process. It is expensive and not practical to change this sales order. You can create a processing constraint that rejects the change if the orchestration process has achieved the shipment step.

Reject Orders That Do Not Include Required Attributes

Assume your company does not deliver items to an address that does not include a ship-to contact. You can create a processing constraint that rejects a sale order if the user does not include a ship-to contact.

Reject Changes That Require Approval

Assume your company does not allow a customer service representative to submit a change if the transaction value exceeds $100, and if a manager has not approved the change. You can create a processing constraint that rejects a change if the
user uses the customer service representative role, and if the transaction exceeds $100, and if a manager has not approved the change.

Managing Processing Constraints: Procedure

This topic describes how to define a processing constraint.

Summary of the Work

To define a processing constraint, do the following work:

1. Create a record set.
2. Create a validation rule set.
3. Create a processing constraint.
4. Test your work.

In this example, you define a processing constraint that prevents a change to any order fulfillment line that is part of an orchestration process that is in the shipping stage. This topic includes example values. You might use different values, depending on your business requirements.

Create a Record Set

Create a record set:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup page, click Order Management.
3. Search for, and then open Manage Processing Constraints.
4. On the Manage Processing Constraints page, click Record Sets, click Actions, click Add Row, and then enter the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Fulfillment Lines That Belong to Same Customer</td>
</tr>
<tr>
<td>Description</td>
<td>A record set created on fulfillment lines that belong to the same customer.</td>
</tr>
<tr>
<td>Short Name</td>
<td>FCST</td>
</tr>
<tr>
<td>Entity</td>
<td>Order Fulfillment Line</td>
</tr>
</tbody>
</table>

5. In the Fulfillment Lines That Belong to Same Customer area, click Actions, click Add Row, and then set the following value.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Name</td>
<td>Bill-to customer ID</td>
</tr>
</tbody>
</table>

6. Click Save, and then click Generate Packages.
7. In the Confirmation dialog, click OK.
## Create a Validation Rule Set

Create a validation rule set:

1. On the Manage Processing Constraints page, click **Validation Rule Sets**.
2. Click **Actions**, click **Add Row**, and then enter the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Shipment VRS</td>
</tr>
<tr>
<td>Description</td>
<td>The validation rule set for lines with status Awaiting Shipment.</td>
</tr>
<tr>
<td>Short Name</td>
<td>SHIP</td>
</tr>
<tr>
<td>Validation Type</td>
<td>Table</td>
</tr>
</tbody>
</table>

3. In the Shipment VRS area, click **Actions**, click **Add Row**, and then set the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Name</td>
<td>Status Code</td>
</tr>
<tr>
<td>Validation Operation</td>
<td>Equal to</td>
</tr>
<tr>
<td>Value String</td>
<td>&quot;Awaiting shipment&quot;</td>
</tr>
<tr>
<td></td>
<td>You must include the double quotation marks.</td>
</tr>
</tbody>
</table>

4. Click **Save**, and then click **Generate Packages**.
5. In the Confirmation dialog, click **OK**.

## Create a Processing Constraint

Create a processing constraint:

1. On the Manage Processing Constraints page, click **Constraints**.
2. Click **Actions**, click **Add Row**, and then enter the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraint Name</td>
<td>Shipping Constraint</td>
</tr>
<tr>
<td>Display Name</td>
<td>Shipping Constraint</td>
</tr>
<tr>
<td>Constraint Entity</td>
<td>Order Fulfillment Line</td>
</tr>
<tr>
<td>Constrained Operation</td>
<td>Update</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Does not contain a check mark. If Enabled contains a check mark, then your constraint might affect other activities.</td>
</tr>
</tbody>
</table>

3. In the Shipping Constraint area, in the Conditions list, click **Actions**, click **Add Row**, and then set the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Number</td>
<td>1</td>
</tr>
<tr>
<td>Validation Entity</td>
<td>Order Fulfillment Line</td>
</tr>
<tr>
<td>Validation Rule Set</td>
<td>Shipment VRS</td>
</tr>
<tr>
<td>Record Set</td>
<td>Fulfillment Lines That Belong to Same Customer</td>
</tr>
<tr>
<td>Message</td>
<td>You cannot update the fulfillment line because it is in the shipping stage.</td>
</tr>
</tbody>
</table>

4. In the Shipping Constraint area, click **Applicable Roles**, make sure All Roles is chosen, and then click **Save**.

Test Your Work

Test your work:

1. In the Order Management work area, access a fulfillment line that includes a Status of Awaiting Shipment.
2. Attempt to change any attribute.
3. Verify that Order Management does not allow you to make the change, and displays the message that you specified in the Message field when you defined the constraint.

Managing Constraint Entities: Explained

You can use a constraint entity to identify the business object or orchestration process that a processing constraint constrains.

You can specify the following types of constraint entities on the Manage Constraint Entities page.

<table>
<thead>
<tr>
<th>Type of Constraint Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Constrained changes according to the value of an attribute that a user can view in the Order Management work area. The order header is an example of a view constraint entity. For example, if you specify Order Header as the constraint entity of a processing constraint, then the constraint constrains changes according to the order header. You can choose one of the following types of view constraint entities:</td>
</tr>
<tr>
<td></td>
<td>• Order Fulfillment Line</td>
</tr>
<tr>
<td></td>
<td>• Order Line</td>
</tr>
<tr>
<td></td>
<td>• Order Header</td>
</tr>
</tbody>
</table>
### Type of Constraint Entity | Description
---|---

| You can use a predefined view constraint entity, but you cannot create a new one. |

To specify the attributes to constrain, you can use the Attribute Details list that the Manage Constraint Entities page displays when you choose a view constraint entity. For details, see Constraining Changes to Attributes: Procedure.

### Process

Constrains an action from occurring at some point in an orchestration process, such as updating an attribute or deleting a table entity, according to a combination of orchestration process, task, and service.

For example, the task layer service named Update Shipping references a process constraint entity named OrderOnlyProcess that uses the following configuration:

- Orchestration Process is OrderOnlyProcess
- Task is Shipping
- Service is UpdateShipment

This process constraint entity constrains a change only if the OrderOnlyProcess orchestration process is running, and if this process is currently on the Shipping task, and if this process calls the UpdateShipment service.

Note the following:

- A process constraint entity considers the current position of the transaction in the orchestration process flow.
- A process constraint entity can validate required attributes for a fulfillment request, such as a Create Shipment request, Update Shipment request, or Create Reservation request.
- You can use a predefined process constraint entity or create a new one.

---

### Constraining Changes to Attributes: Procedure

Order Management Cloud comes predefined to allow processing constraints on some attributes, but not for others. For example, if you must constrain changes to the Latest Acceptable Ship Date attribute of an order header, then you must enable processing constraints for this attribute.

Constrain changes on an attribute:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Constraint Entities.
4. On the Manage Constraint Entities page, in the Entity Type field, click **View Entity**, and then click **Search**.

   The Manage Constraint Entities page displays view constraint entities and process task constraint entities. Order Management can constrain changes to any process task constraint entity, so it is not necessary to enable them for constraint.

5. In the Search Results, click the constraint entity that displays the attribute that you must constrain.

   For example, to constrain changes to the Latest Acceptable Ship Date field that the order header displays, in the Display Name column, click **Order Header**.

6. In the Attribute Details list, add a check mark to the **Constraint Enabled** option next to the attribute that you must constrain, such as Latest Acceptable Ship Date.
Constraining Changes That Users Make in Extensible Flexfields: Procedure

You can constrain the changes that a user can make in an extensible flexfield. For example, you can use a processing constraint to not allow the user to update an extensible flexfield if a fulfillment line is closed, or to require the user to enter a value in an extensible flexfield at a step of an orchestration process.

Note the following points:

- You use the Manage Constraint Entities page to enable an extensible flexfield so that you can use it on the Manage Processing Constraints page to create a processing constraint. These pages display the extensible flexfields that are currently configured.
- Extensible flexfields are not available on the Attributes menu of the Record Sets tab.
- You must enable an extensible flexfield before you can use it.

The example in this topic describes how to create a processing constraint that implements the following logic:

- If a fulfillment line is closed, then do not allow the user to change the value of the extensible flexfield named Subcontractor ID

You can use an extensible flexfield to meet other business requirements. For details, see Configuring Extensible Flexfields in Order Management: Explained.

Summary of the Work

To constrain the changes that a user can make in an extensible flexfield, do the following work:

1. Enable the extensible flexfield.
2. Create a validation rule set.
3. Create a processing constraint.
4. Test your work.

This topic uses example values, such as Subcontractor ID. You might use different values, depending on your business requirements.

Enable the Extensible Flexfield

Enable an extensible flexfield so that a processing constraint can reference it:

1. If necessary, publish and deploy the extensible flexfield that you require.

   For details, see Publishing and Deploying Extensible Flexfields in Order Management: Procedure.

2. In the Navigator, click Setup and Maintenance.

3. On the Setup page, click Order Management.

4. Search for, and then open Manage Constraint Entities.

5. On the Manage Constraint Entities page, set Entity Type to Equals, View Entity, and then click Search.

6. In the Search Results area, select Order Fulfillment Line.

7. In the Attribute Details area, locate the extensible flexfield that you must enable, make sure Constraint-Enabled contains a check mark, click Save and Close, and then click Done.

   For this example, enable Subcontractor:ID.
Create a Validation Rule Set

Create a validation rule set:

1. On the Setup page, search for, and then open Manage Processing Constraints.
2. On the Manage Processing Constraints page, click **Validation Rule Set**.
3. Click **Actions**, click **Add Row**, set the following values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Fulfillment Line Is Closed</td>
</tr>
<tr>
<td>Description</td>
<td>Fulfillment line is closed.</td>
</tr>
<tr>
<td>Short Name</td>
<td>FLCLOSE</td>
</tr>
<tr>
<td>Validation Type</td>
<td>Table</td>
</tr>
<tr>
<td>Entity</td>
<td>Order Fulfillment Line</td>
</tr>
</tbody>
</table>

4. In the Details area, click **Actions**, click **Add Row**, set the following values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Name</td>
<td>Open</td>
</tr>
<tr>
<td>Validation Operation</td>
<td>Equal To</td>
</tr>
<tr>
<td>Value String</td>
<td>N</td>
</tr>
</tbody>
</table>

5. Click **Generate Packages**. Order management makes the validation rule set active and allows you to use it in a processing constraint.

If you add multiple lines in the Details area of a validation rule set, then order management evaluates them together.

You cannot select different contexts of an extensible flexfield in a single validation rule set. If you select two context and segment attributes, then the attributes must use the same context value; that is, the context is mutually exclusive. For example, in the following context and segment combinations, you cannot simultaneously select the ID segment for the Dealer Information context and for the Warranty Information context.

<table>
<thead>
<tr>
<th>Context</th>
<th>Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dealer Information</td>
<td>ID</td>
</tr>
<tr>
<td>Dealer Information</td>
<td>Location</td>
</tr>
<tr>
<td>Warranty Information</td>
<td>ID</td>
</tr>
</tbody>
</table>
Create a Processing Constraint

Create a processing constraint:

1. Click **Constraints**, click **Actions**, click **Add Row**, set the following values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Constrain Subcontractor ID</td>
</tr>
<tr>
<td>Display Name</td>
<td>Constrain Subcontractor ID</td>
</tr>
<tr>
<td>Constraint Entity</td>
<td>Order Fulfillment Line</td>
</tr>
<tr>
<td>Constrained Operation</td>
<td>Update</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Subcontractor: ID</td>
</tr>
</tbody>
</table>

2. In the Details area, click **Actions**, click **Add Row**, set the following values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Number</td>
<td>100</td>
</tr>
<tr>
<td>Validation Entity</td>
<td>Order Fulfillment Line</td>
</tr>
<tr>
<td>Validation Rule Set</td>
<td>Fulfillment Line Is Closed</td>
</tr>
<tr>
<td>Scope</td>
<td>Any</td>
</tr>
<tr>
<td>Record Set</td>
<td>Fulfillment Line Default Record Set</td>
</tr>
<tr>
<td>Message</td>
<td>The fulfillment line is closed. You cannot modify it.</td>
</tr>
</tbody>
</table>

Test Your Work

Test your work:

1. Attempt to modify the Subcontractor ID attribute of a fulfillment line that is closed.
2. Verify that order management does not allow this modification.

Managing Change Orders
Processing Change That Occurs During Order Fulfillment: How It Works

A change that affects a sales order during order fulfillment can originate from a variety of sources, such as from a user through an order capture system, from a user who is using the Order Management work area, or from change orders that you import. Order Management Cloud comes predefined to process these changes.

Order Management cross-references, transforms, validates, and orchestrates the change in a way that is similar to how it does this work for a new sales order. If rules that control how to process a change exist, then Order Management applies them. You can use the setup options to modify how Order Management does this processing.

Consider the following example. Assume a manager at Vision Corporation places a sales order for 12 employee appreciation gifts that include a company logo, and that come in a variety of colors. This sales order includes the following order lines, and each of these order lines references a fulfillment line:

<table>
<thead>
<tr>
<th>Sales Order Line</th>
<th>Fulfillment Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Line 1, 4 red</td>
<td>Fulfillment Line 1, 4 red</td>
</tr>
<tr>
<td>Order Line 2, 4 blue</td>
<td>Fulfillment Line 2, 4 blue</td>
</tr>
<tr>
<td>Order Line 3, 4 black</td>
<td>Fulfillment Line 3, 4 black</td>
</tr>
</tbody>
</table>

Assume Order Management assigns an orchestration process to each fulfillment line, and that this process contains the following steps:

1. Create Scheduling
2. Create Reservation
3. Create Shipping
4. Wait for Shipping
5. Create Billing
6. Wait for Billing

Assume that, during setup, you set the following settings on this orchestration process.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Mode</td>
<td>Advanced</td>
</tr>
<tr>
<td>Order Attributes That Identify Change</td>
<td>You include the following attributes:</td>
</tr>
<tr>
<td></td>
<td>• Ordered Quantity</td>
</tr>
<tr>
<td></td>
<td>• Demand Class Code</td>
</tr>
<tr>
<td></td>
<td>• Requested Ship Date</td>
</tr>
<tr>
<td>Cost of Change</td>
<td>You specify the following rule.</td>
</tr>
<tr>
<td></td>
<td>If the fulfillment line status is:</td>
</tr>
<tr>
<td></td>
<td>• Reserved. The cost of change is 15.</td>
</tr>
</tbody>
</table>
Assume that, within one day of placing the sales order, one employee resigns and one employee transfers to another division of the company. The manager calls the customer service number and requests to change the sales order to fewer gifts and to replace blue with beige. The customer service representative searches for the existing sales order, and then notices that the fulfillment lines are on the Create Shipping step. The representative submits the following change order:

- Change order line 1 to a quantity of 3, color red.
- Change order line 2 to a quantity of 3, color beige.
- Change order line 3 to a quantity of 4, color black.

Order Management uses the following sequence to process this change order:

1. Receives a request to change a sales order.
2. Performs a lookup to determine whether or not it already received the order key. If it already received the order key, then it exits this sequence.
3. Sends a request to the fulfillment system that is responsible for the task that is running. This request includes details about whether or not to stop processing, whether or not to accept the change, and the current status.
4. Determines whether processing constraints on the order header prevent change processing, and whether not the sales order is closed. Note the following points:
   - If these constraints do not allow a change, or if the sales order is closed, then Order Management rejects the entire change and exits this change sequence.
   - The header evaluation prevents needless processing of a closed sales order.
   - Order Management evaluates these attributes when the user creates the revision, and also when the user submits the revision to make sure that other fulfillment changes did not occur after the user created the revision that might affect the sales order.
5. Determines whether the processing constraints on each fulfillment line allow the change. Note the following points:
   - If any constraint does not allow the change on any fulfillment line, then Order Management rejects the entire change and exits this sequence.
   - Some processing constraints on a fulfillment line do not allow some changes, by default, such as updating an order line if the fulfillment line status is Shipped.
   - In this example, only one day has elapsed since the user placed the sales order, the orchestration process has not shipped, and Order Management allows the change.
6. Calls the delta service. This service does the following work:
   - Identifies the orchestration process steps that reference the attributes that Order Management uses to identify change.
   - Analyzes the state of each step that Order Management records every time it called a task layer service when the orchestration process was running.
   - Uses this state information to determine the processing that Order Management must perform to incorporate the change.

Order Management compares the change order to the existing sales order to determine whether or not the value of an attribute that you specify on the Order Attributes That Identify Change page changed. In this example, the
customer changed the quantity on two order lines, and you specified Ordered Quantity as a change attribute, so Order Management starts compensation. You also added a check mark to Use Transactional Item Attributes, so Order Management examines transactional item attributes for changes when it compares the change order to the original sales order. Color is one of these attributes, so the color change would also start compensation.

7. If the delta service determined that the order attributes require compensation, then Order Management uses the compensation pattern that the step references to compensate each of these steps. You can use the following compensation patterns:

- Undo
- Redo
- Update
- Cancel
- None

You can also specify a business rule that determines the action to take according to this compensation pattern. Most orchestration process steps do not include a compensation pattern, and they use Update, by default. In this example, Order Management compensates each of the following steps.

<table>
<thead>
<tr>
<th>Steps That Order Management Compensates</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Scheduling</td>
<td>Cancels the Create Scheduling step, and then creates a new instance of this step. Global Order Promising determines availability. Order Management replans fulfillment, and then assigns a revised date to each orchestration process step.</td>
</tr>
</tbody>
</table>
| Create Reservation                     | Assume the orchestration process in this example includes the following compensation pattern for the Create Reservation step:  
  - If the Demand Class Code is not Gold, then cancel the Create Reservation step, and then create a new instance of this step.  
This rule configures Order Management to release supply and creates a new reservation for all customers except Gold customers. Order Management also updates the Create Reservation step according to the new dates, and updates the reserved quantity. |
| Create Shipping                         | Updates the Create Shipping step with the new dates and new item. |

8. The compensating services run, and then finish. These services use the FIFO (first in, first out) sequence to compensate the sales order, according to the orchestration process sequence, by default. If Order Management must cancel the entire sales order, then it uses a LIFO (last in, first out) sequence.

9. If the original orchestration process:

- **Can accommodate the change.** Order Management uses the original orchestration process to continue processing.
- **Cannot accommodate the change.** Order Management cancels the original orchestration process, and then starts a new one that can accommodate the change.

10. Compensation finishes. Processing for this orchestration process is now at the same step that it was on when Order Management received the change order in step 1.

Order Management sends an update message to each external system to update the original message with the changed order that includes the modified attributes. For example:
Managing Change That Occurs During Order Fulfillment: Explained

You can administer the following features to manage how Order Management Cloud processes changes that occur during order fulfillment, including how it does order compensation.

<table>
<thead>
<tr>
<th>Feature That You Can Administer</th>
<th>Description</th>
</tr>
</thead>
</table>
| Change Mode                    | An option that you can set on an orchestration process that determines how frequently Order Management records the state of this process. It compares these states during order compensation. You can set it to one of the following values:  
  - **None.** Do not record the state of the orchestration process, and do not allow a change.  
  - **Simple.** Record the state of the orchestration process when it starts and at the step where the orchestration process receives the change.  
  - **Advanced.** Record the state of the orchestration process at each orchestration process step. |
| Cost of Change Rule            | A rule that you can define on an orchestration process that specifies the business costs that your organization will incur as a result of processing the requested change. |
| Use Flexfield Attributes       | An option that you can set on an orchestration process or orchestration process step that specifies whether or not to examine flexfield attributes when determining whether or not to compensate a sales order. For example, if you select Use Flexfield Attributes, then Order Management examines the flexfield attributes that are associated with an item to determine whether or not it must compensate the orchestration process. |
| Use Transactional Item Attributes | An option that you can set on an orchestration process or orchestration process step that specifies whether or not to examine transactional attributes when determining whether or not to compensate a sales order. For details, see Transactional Attributes: Explained. |
| Use Dynamic Attributes         | An option that you can set that specifies whether or not to examine dynamic attributes when determining whether or not to compensate a sales order. |
| Compensation Pattern          | A rule that you can define on an orchestration process step that specifies the adjustments to make to a sales order when this order changes. For example, if Order Management receives a change order that specifies to ship an item from a different warehouse for the Create Shipment step, then it runs the Cancel service and the Create service again. If you do not specify a compensation pattern, then Order Management uses the predefined compensation pattern, by default. A predefined pattern uses the Update service, the Cancel service, or the Create service. |
### Feature That You Can Administer

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A field that you can set on an orchestration process step. The task type that you select determines the order attributes that Order Management uses when it determines whether or not it must compensate a sales order for this step.</td>
<td></td>
</tr>
</tbody>
</table>

Order Management comes predefined to use a set of attributes for each predefined task type. You cannot modify this predefined configuration, but you can add more attributes. Order Management does not add any of these attributes to a new task type that you add, and it does not evaluate this task until you add these attributes. To add them, you must click **Add All**.

### Order attribute

You can administer Order Management so that it examines order attributes when it determines whether or not to compensate a sales order. For details, see Managing Order Attributes That Identify Change: Procedure.

### Processing constraints that control change

A processing constraint that you can specify to control the changes that Order Management makes to a sales order, such as what data a user can modify, or when the user can modify data. If the user attempts to modify this data, then Order Management displays an error message. For details, see Processing Constraints: Explained.

---

**Compensating Sales Orders That Are Awaiting Shipping**

The compensation pattern for the Shipping task uses the Cancel and Create pattern for item substitution because Oracle Fusion Shipping supports only cancel and create. It does not support the Update pattern for item substitution during shipping.

If you do not use Oracle Fusion Shipping, and if your shipping system:

- **Supports Update.** No change is required.
- **Does not support Update.** You must modify the compensation rule so it uses the Cancel and Create pattern.

If you require different behavior, then you can write a compensation rule according to an item change. Note that business rules call the Update service for an item change.

Change management can only identify changes to an extensible flexfield. However, you can write a compensation rule in the pattern that examines the extensible flexfield segment so it can determine whether to perform compensation.

**Using Cancel and Create Patterns in Compensation Rules**

This example implements the following logic:

- If the sales order is at the await shipping step, and if the Order Entry Specialist chooses to substitute the item, then cancel the existing request to shipping, create a new request, and then schedule, reserve, and interface the new request to shipping.

Note that each shipping step and reservation step on predefined orchestration processes that interface with shipping already use this logic. You must add this logic only if you define your own orchestration process.

Use the Cancel and Create pattern in a compensation rule:

1. Use the Manage Orchestration Process Definitions page to open and edit the orchestration process you must modify.
2. In the row that ships the item, in the Compensation Pattern column, click **Click for Rule**, and then set the following values.
Your rule must resemble the following set up.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>DooSeededOrchestrationRules.DOOLine</td>
</tr>
<tr>
<td>If header</td>
<td>is a DooSeededOrchestrationRules.DOOLine</td>
</tr>
<tr>
<td>and</td>
<td>there is a case where</td>
</tr>
<tr>
<td>fline</td>
<td>is a header/ childFlines</td>
</tr>
<tr>
<td>and</td>
<td>fline.attributeChanged() is true</td>
</tr>
<tr>
<td>then</td>
<td>assign header.mRuleDecision.compenationPattern = &quot;CANCEL, CREATE&quot;</td>
</tr>
</tbody>
</table>

Managing Order Attributes That Identify Change: Procedure

Order Management Cloud examines order attributes to identify the changes that a change order contains, and then uses these details to compensate a sales order.

For example, Ordered Quantity is an order attribute. If a user increases the Ordered Quantity of a sales order, then Order Management must schedule and ship more supply. To do this work, it compensates the Schedule step and the Shipment step of the orchestration process when it receives the change order that includes the new quantity.

Each step references a task type, and each task type references order attributes that Order Management uses to determine whether or not to compensate a sales order. In this example, the Schedule step and the Shipment step each reference a task type that references the Ordered Quantity attribute.

In this topic, assume you are creating an orchestration process step that must use the Ordered Quantity attribute to identify change.

This topic includes example values. You might use different values, depending on your business requirements.

Manage order attributes that identify change:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup, click **Order Management**.
3. Search for, and then open Manage Order Attributes That Identify Change.
4. In the Task Type list, click a task type that looks like it might meet the needs that your orchestration process step requires.

   Order Management comes predefined with a number of task types. To reduce maintenance, it is recommended that you use one of these predefined task types before you create a new task type.

5. Click **Actions**, and click **Edit**.

   For this example, click **Shipment**.
6. On the Edit Order Attributes That Identify Change page, in the Orchestration Components list, click a component that you think might contain the attribute that Order Management must use to identify the change.

For this example, click **Order Line**.

7. Examine the Attributes list.

When your orchestration process step runs, it will examine each attribute that this list contains to determine whether or not to compensate the order when it receives a change order. If this list includes the attributes that your step requires to identify change, then set the Task Type attribute of your orchestration process step to this task type, and then exit this procedure. For this example, the list does include Ordered Quantity, so you can set Task Type to Shipment. If the Predefined column of the Attributes list includes a check mark, then you cannot remove this attribute.

8. If the Attributes list does not include the attributes that your step requires to identify change, then do the following work:

   - Click **Actions**, and then click **Select and Add**.
   - In the Change Attributes dialog, click an attribute, such as **Scheduled Ship Date**.
   - Click **Apply**, click **OK**, and then exit this procedure.

Note that you can also do the following work on the Manage Order Attributes That Identify Change page:

- To add an attribute to all tasks that the Task Type list displays, click **Actions**, and then click **Edit All**.
- To add a new task type, click **Actions**, and then click **Create**. On the Create Order Attributes That Identify Change page, add attributes, as necessary. If you create a new task type, and if you do not use this page to add at least one attribute to this task type, then Order Management will not examine any attributes when it identifies change.

Measuring the Cost of Change: Explained

Cost of change is a numeric value that measures how much a change impacts an orchestration process. For example, the monetary cost to a company, or the difficulty that is associated with incorporating the change.

If the source system requests a determination for a cost of change, then Order Management Cloud calculates this value, and then returns it to the source system so the customer service representative can choose whether or not to proceed with the change. The source system can request this value before it submits the sales order. Order Management also calculates this cost after it compensates the sales order.

You use a business rule to assign the cost of change to an orchestration process. If you choose not to use values for the cost of change, then Order Management uses a value of zero to calculate the cost.

Measuring the Cost of Change: Procedure

You can create a business rule that measures the cost of change for an orchestration process.

In this example, assume the order administrator of a flooring company requires a business rule that measures the cost of change that the company will incur when a customer requests a change. If the fulfillment line status is:

- Scheduled, then the cost is low
- Shipped, then the cost is high
To create cost of change rules for this example, you create If and Then statements that implement the following business rules:

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

This example includes a simple business rule that you can use with an orchestration process that includes only one line. To write a business rule for an orchestration process that includes multiple lines, it is recommended that you use advanced rules. For details, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then download the attachment that describes how to use business rules.

This topic includes example values. You might use different values, depending on your business requirements.

Summary of the Work

To create cost of change rules for this example, do the following work:

1. Set up your order capture system.
2. Create the If statement for the first rule.
3. Create the Then statement for the first rule.
4. Create the If statement for the second rule.
5. Create the Then statement for the second rule.
6. Test your work.

Set Up Your Order Capture System

Order Management Cloud does not display the result when it calculates the cost of change. Instead, you must set up your order capture system to get this result from Order Management, and then display it so that someone who uses this system can take the necessary action. To get this result, you can use the GetOrderDetails method of the Get Orchestration Order service. For details about how to use web services with your order capture system, search for Doc ID 2051640.1 (Using Web Services with Oracle Fusion Order Management Cloud) on My Oracle Support, at https://support.oracle.com.
Create the If Statement for the First Rule

You will create the following business rule.

Cost of Change Rule Set

where:

- **DooSeededOrchestrationRules**: A dictionary that contains rule sets, facts, functions, variables and so on for order orchestration.
- **DOOFLine**: Is a fact in the DooSeededOrchestrationRules dictionary. It contains fulfillment line attributes. DOOFLine is an abbreviation for distributed order orchestration (DOO) fulfillment line (FLine).
- **statusCode** is a fulfillment line attribute.
- **SCHEDULED** is one value that statusCode might contain.
- **Result** is a fact in the DooSeededOrchestrationRules dictionary. You use it to store the results of the business rule you are defining.
- **resultObjKey** is a property of the Result fact.
- **resultObjKey:5** sets the value of resultObjKey to 5.
Create the If statement for the first rule:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Orchestration Process Definitions.
4. On the Manage Orchestration Process Definitions page, locate the orchestration process where you must add the cost of change rule, and then open it for editing.
5. In the Orchestration Process area, click **Click for Rule** next to Cost of Change Rule.
6. In the Cost of Change Rule dialog, click **Add Rule**, click **Expand**, click **Show Advanced Settings**, and then set the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Cost of Change for Scheduled Lines</td>
</tr>
<tr>
<td>Description</td>
<td>This rule measures the cost to change a sales order when the fulfillment line status is Scheduled. It sets the cost of change to 5.</td>
</tr>
</tbody>
</table>

7. Click **Left Value**.
8. In the Condition Browser dialog, expand **DooSeededOrchestrationRules**, expand **DOOFLine**, click **StatusCode**, and then click **OK**.
9. In the Right Value field, enter “SCHEDULED”. You must include the double quotation marks.

Create the Then Statement for the First Rule

Create the Then statement for the first rule:

1. In the Then area, click **Add Action**, and then click **Assert New**.
2. Click **Select a Target**, and then click **DooSeededOrchestrationRules.Result**.
3. Click **Edit Properties**.
4. In the Properties dialog, in the resultObjKey row, enter the following value, and then click **OK**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>5</td>
</tr>
</tbody>
</table>

5. Click **Collapse**.
Create the If Statement for the Second Rule

You will create the following rule.

1. Click **Add Rule**, click **Expand**, click **Show Advanced Settings**, and then set the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Cost of Change for Shipped Lines</td>
</tr>
</tbody>
</table>

   Description: This rule measures the cost to change a sales order when the fulfillment line status is Shipped. It sets the cost of change to 50.

2. Click **Left Value**.
3. In the Condition Browser dialog, expand **DooSeededOrchestrationRules**, expand **DOOFLine**, click **StatusCode**, and then click **OK**.
4. In the Right Value field, enter "SHIPPED". You must include the double quotation marks.

Create the Then Statement for the Second Rule

Create the Then statement for the second rule:

1. In the Then area, click **Add Action**, and then click **Assert New**.
2. Click **Select a Target**, and then click **DooSeededOrchestrationRules.Result**.
3. Click **Edit Properties**.
4. In the Properties dialog, in the resultObjKey row, in the Value column, enter 50, and then click **OK**.
5. In the Cost of Change Rule dialog, click **Save**.
6. On the Edit Orchestration Process Definition page, click Save.

Test Your Work

Test your work:

1. Change a sales order that includes a fulfillment line status that is Scheduled.
2. Wait for Order Management to process the change.
3. Make sure your order capture system displays a value of 5 for the cost of change.
4. Change a sales order that includes a fulfillment line status that is Shipped.
5. Wait for Order Management to process the change.
6. Make sure your order capture system displays a value of 50 for the cost of change.

Related Topics

- Defining Orchestration Processes: Procedure
- Creating Business Rules: Demo

Compensating Sales Orders That Change: Procedure

You can create a compensation pattern that specifies the adjustments that Order Management Cloud makes when it processes a fulfillment task. It does this work to process the request it receives to change a sales order.

In this topic, assume you are an order administrator who requires a compensation pattern that allows sufficient time to reschedule an orchestration process when a customer requests to change the ship date. You create the following rule:

- If the requested ship date is less than or equal to the current date plus three days, then cancel, and then redo the Ship Product step.

This example includes a simple rule that you can use with an orchestration process that includes only one line. To write a rule for an orchestration process that includes multiple lines, it is recommended that you use advanced rules. For details, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then open the attachment that describes how to use business rules.

For a demonstration, see Creating Business Rules: Demo.

This topic includes example values. You might use different values, depending on your business requirements.

Summary of the Work

To compensate sales orders that change, do the following work:

1. Create the If statement.
2. Create the Then statement.
You will create the following rule.

**Compensation Pattern Set**

![IF/THEN Rules](image)

**IF**

- DooSeededOrchestrationRules
- DOOFLine
- requestShipDate
- time
- CurrentDate.date.timeInMillis+3*24*60*60*1000

**THEN**

- assert new
- Result
- resultObj

where:

- **DooSeededOrchestrationRules.** The dictionary you will use in this rule to store objects and their values. Note that DOO means distributed order orchestration, which is term that Order Management used for order orchestration in earlier releases.
- **DOOFLine.** An object in the DooSeededOrchestrationRules dictionary. It contains fulfillment line attributes and their values.
- **requestShipDate.** A fulfillment line attribute in the DOOFLine object.
- **time.** A property of the toLocateString function. This function references the requestShipDate attribute.
- **CurrentDate.date.timeInMillis+3*24*60*60*1000.** An equation that specifies to use the date that occurs three days after the current date.
- **Result.** An object in the DooSeededOrchestrationRules dictionary that you will use to store the result of the If statement.
- **resultObj.** A property of the Result object. It stores "CANCEL_CREATE", which is the value that this rule sends to end the condition.

**Create the If Statement**

You will create the following statement:

- If the requested ship date is less than or equal to the current date plus three days

Create the If statement:

1. In the Navigator, click **Setup and Maintenance.**
2. On the Setup page, click **Order Management.**
3. Search for, and then open Manage Orchestration Process Definitions.
4. On the Manage Orchestration Process Definitions page, locate the CallCustomerWhenLargeInvoice orchestration process, click Actions, and then click Edit.
For details about how to create CallCustomerWhenLargeInvoice, see Adding Branches to Orchestration Processes: Procedure.
6. In the Ship Item row, in the Compensation Pattern column, click Click for Rule.
7. In the Compensation Pattern dialog, click Add Rule, click Expand, and then set the following value:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Rule to Compensate When Ship Date Changes</td>
</tr>
</tbody>
</table>

8. In the If area, click Left Value, expand DOOSeededOrchestrationRules, expand DOOFline, expand requestShipDate, click time, and then click OK.
9. In the Is list, click Is, and then click Same or Less Than.
10. Click Right Value.
11. In the Condition Browser, expand CurrentDate, expand Date, expand Time, and then click timeInMillis.
Do not click OK.
12. Near the top of the Condition Browser, in the window, after CurrentDate.date.timeInMillis, add +3*24*60*60*1000.
Make sure this window now contains the following value:
CurrentDate.date.timeInMillis+3*24*60*60*1000
The calculation 3*24*60*60*1000 calculates the total number of milliseconds that three days contains, where:
- 3 is the number of days.
- 24 is the number of hours in one day.
- 60 is the number of minutes in one hour.
- 60 is the number of seconds in one minute.
- 1000 is the number of milliseconds in one second.
13. Click OK.

Create the Then Statement
This statement cancels, and then restarts the Ship Product step so that Order Management can reschedule the shipment.
Create the Then statement:

1. In the Then area, click Add Action, and then click Assert New.
2. Click Select a Target, and then click DooSeededOrchestrationRules.Result.
3. Click Edit Properties.
4. In the Properties dialog, in the ResultObj row, enter the following value. You must include the double quotation marks (").

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>&quot;CANCEL_CREATE&quot;</td>
</tr>
</tbody>
</table>
5. Click **OK**.
6. Click **Validate**, make sure the Validation Log that displays does not contain any errors, and then click **Save**.
7. On the Edit Orchestration Process Definition page, click **Save**.

**Related Topics**
- Creating Business Rules: Demo

## Notifying Systems When Orders Change

### Sending Notifications from Order Management to External Systems: Explained

Order Management Cloud uses a business event to send a notification to an external system when a change occurs, such as when information in a sales order or fulfillment line changes. You can control these business events.

You can use a web service to communicate information to an upstream system, such as a source system, or to a downstream system, such as a fulfillment system. You can also use a predefined business event to send information to these systems about a significant event that occurs in the context of a business process. The following diagram illustrates how you can use a business event to communicate with these systems.

A business event can notify a system when one of the following changes occur:

- A change to an order attribute occurs.
- A hold is applied.
- A jeopardy priority changes.
- A fulfillment line splits.
- The status on an order header changes.
- A fulfillment line closes.
A fulfillment line achieves a status.
Compensation for an orchestration process finishes successfully or not successfully.

You can specify whether or not the business event is sufficiently important to communicate in the context of your business requirements.

Managing Business Event Trigger Points
You can use the Manage Business Event Trigger Points page in the Setup and Maintenance work area to configure a business event trigger point. Note the following:

- You must associate a connector with any business event that you specify. For details, see Sending Notifications from Order Management Cloud to External Systems: Procedure.
- The business events that this page displays are not active, by default, except for business events that monitor closing a fulfillment line.
- A modification that you make on this page takes effect immediately.
- The settings that you make affect only one instance of Order Management.
- You cannot add or delete the business events that this page displays.
- It is recommended that you configure only the business event trigger points that you require. Each business event trigger point might affect system performance.

Using Predefined Business Events
Order Management comes predefined with the following business events. You can specify them on the Manage Business Event Trigger Points page.

<table>
<thead>
<tr>
<th>Business Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Order Compensation Complete</td>
<td>Sends a notification when Order Management finishes processing for a change order. If a change order results in an error, then this business event reports the error.</td>
</tr>
<tr>
<td>Fulfillment Line Status Update</td>
<td>Sends a notification when Order Management changes the status value of a fulfillment line. To specify the status values to notify, do the following:</td>
</tr>
<tr>
<td></td>
<td>1. Navigate to the Manage Orchestration Process Definitions page.</td>
</tr>
<tr>
<td></td>
<td>2. On the Edit Status Rule Set page, add a check mark to the Notify External System option next to each status value to notify.</td>
</tr>
<tr>
<td>Fulfillment Line Closed</td>
<td>Sends a notification when Order Management closes a fulfillment line. Some applications, such as Cost Management, can use this information to perform downstream processing on the fulfillment line.</td>
</tr>
<tr>
<td>Hold</td>
<td>Sends a notification when a user applies a hold, or when the order capture system or fulfillment system applies a hold on one of the following items:</td>
</tr>
<tr>
<td></td>
<td>- Order</td>
</tr>
<tr>
<td></td>
<td>- Order line</td>
</tr>
<tr>
<td></td>
<td>- Fulfillment line</td>
</tr>
<tr>
<td>Business Event</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Note the following:</td>
<td></td>
</tr>
<tr>
<td>• If a hold is not immediately effective, for example, if the hold applies to a future task, then Order Management communicates a business event when the hold becomes effective, not when the request to add the hold occurs.</td>
<td></td>
</tr>
<tr>
<td>• If change management releases or applies a hold, then Order Management does not communicate a business event.</td>
<td></td>
</tr>
<tr>
<td>Jeopardy</td>
<td>Sends a notification when process planning changes a jeopardy priority value, such as High, Medium, or Low. You can specify the value to notify.</td>
</tr>
<tr>
<td>Order Attribute Update</td>
<td>Sends a notification when Order Management changes a fulfillment line attribute during order fulfillment. You can choose each attribute from the set of attributes that a task references.</td>
</tr>
<tr>
<td>Note the following:</td>
<td></td>
</tr>
<tr>
<td>• Each attribute that the Order Update: Attributes That Trigger Events area displays can trigger a business event. These attributes are predefined. You can also configure an extensible flexfield that modifies these attributes. If you do configure an extensible flexfield, then Order Management displays the segments that you select in the Order Attribute Update: Attributes That Trigger Events area.</td>
<td></td>
</tr>
<tr>
<td>• If Order Management updates more than one attribute of the Order Attribute Update business event at the same time, then it sends only one notification. This notification includes information about the attributes that it updated.</td>
<td></td>
</tr>
<tr>
<td>• The status values on the Order Header Status Update: Status Values That Trigger Events area start the Order Header Status Update business event. These values are predefined. You cannot add status values in this area.</td>
<td></td>
</tr>
<tr>
<td>Order Header Status Update</td>
<td>Sends a notification when Order Management cancels, partially closes, or closes a sales order. You can specify the status value to notify.</td>
</tr>
<tr>
<td>Split</td>
<td>Sends a notification when a partial shipment occurs or when an integrated system splits a fulfillment line.</td>
</tr>
</tbody>
</table>

**Sending Notifications From Order Management to External Systems: Procedure**

Order Management Cloud uses a connector to communicate with your external system. The connector specifies the URL that locates your external system and the security credentials that your external system requires during communication. You must configure this connector, and then associate it with the business event that sends notifications to your external system.

Order Management monitors the conditions that your business events specify during order fulfillment. If Order Management determines that it must send a notification according to one of these business events, then it communicates this information to the endpoint URL for each connector that you associate with the event trigger point. For example, if you add a check mark to the Notify External System option for the Shipped status value on a status rule set, then Order Management communicates a business event any time it sets a fulfillment line status to Shipped, and it sends this business event to the endpoint URL of the connector that you associate with a business event named Fulfillment Line Status Update. For details about this option, see Adding Status Conditions to Fulfillment Lines: Procedure.

**Summary of the Work**

To configure Order Management to send notifications to external systems, do the following work:

1. Configure the connector.
2. Manage business event trigger points.
3. Set the notification frequency.

Configure the Connector

Configure the connector that Order Management uses to communicate with your external system:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage External Interface Web Service Details.
4. On the Manage Connector Details page, click **Actions**, click **Add Row**, and then set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target System</td>
<td>Select the target system that must receive the notification. This system subscribes to the business event.</td>
</tr>
<tr>
<td>Connector Name</td>
<td>Enter text that describes the purpose of the connector. For example, to indicate that you use this connector to send notifications to a legacy system, enter LEG_Notification.</td>
</tr>
<tr>
<td>Connector URL</td>
<td>Specify an endpoint URL. This URL is the address of a web service that you deploy on the subscriber system. Order Management will call this web service, and the web service must accept the event payload that Order Management sends.</td>
</tr>
<tr>
<td>User Name</td>
<td>Enter the user name that the subscriber system requires.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password that the subscriber system requires.</td>
</tr>
</tbody>
</table>

Manage Business Event Trigger Points

Manage the business event trigger points that determine when to communicate business events to your external system:

1. On the Search page, search for, and then open Manage Business Event Trigger Points.
2. On the Manage Business Event Trigger Points page, choose a trigger point.

At run time, if an order reaches the trigger point that you specify, then Order Management calls the web service that you referenced in the Connector URL attribute. This web service then sends details of the business event to the subscriber system. For example, to send a notification when a fulfillment line closes, choose the Fulfillment Line Closed trigger point.

3. Click **Actions**, and then click **Add Row**.

Note that some trigger points include an Associated Connectors tab in the details area. If this tab displays, then click it to add the connector.

4. Set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Name</td>
<td>Select the connector that you configured in the Configure the Connector topic.</td>
</tr>
<tr>
<td>Override Default Visibility</td>
<td>Set one of the following values:</td>
</tr>
<tr>
<td></td>
<td>Contains a check mark. Send notifications about each sales order even if the target system that receives the notification does not contain knowledge about this sales order.</td>
</tr>
</tbody>
</table>
Attribute | Description
--- | ---
| **Does not contain a check mark.** Do not send notifications about each sales order to a target system that does not already contain knowledge about this sales order. |

Assume your implementation includes Source System 1 and Source System 2, and that each of these systems can send a source order to Order Management Cloud. Assume that Source System 2 has no knowledge of source orders that originate in Source System 1, and that you do not want to notify Source System 2 when an event occurs that is associated with a source order that originates in Source System 1. In this situation, if Override Default Visibility does not contain a check mark, then Order Management Cloud will not call the connector service for the subscriber system, and Source System 2 will not receive the notification.

5. Optional. Repeat step 3 through step 4.

You can associate the same connector with multiple business events, as necessary.

6. Optional. Repeat step 2 through 5 for each trigger point that you must administer.

7. Click **Save and Close**.

Set the Notification Frequency

You can use the Notify Frequency parameter to specify how frequently Order Management sends notifications. For details, see Managing Order Management Parameters: Procedure.

---

### Using Jeopardy to Manage Delays

#### Jeopardy Priority: Explained

Jeopardy indicates the severity of a delay of a task in an orchestration process. You can modify the predefined range of jeopardy scores for a jeopardy priority to control how Order Management Cloud calculates and displays jeopardy.

For example, you can specify a minimum jeopardy score of 0, and a maximum score of 100 for the Low jeopardy priority. You cannot add or delete a jeopardy priority. You cannot modify the value of a predefined jeopardy priority, such as Low.

Note the following:

- Order Management uses forward planning and backward planning across an orchestration process to calculate the promise dates for each orchestration process task. If one of these tasks is delayed, then Order Management uses jeopardy to indicate the severity of the delay according to the jeopardy settings that you make.
- Order Management calculates the jeopardy score and jeopardy priority every time it plans or replans an orchestration process.
- You can use jeopardy on planned dates or on actual dates. This configuration allows an order manager to become aware that a jeopardy condition might exist before it actually occurs, and to take action to fix the condition and reduce the jeopardy.

#### Jeopardy Score: Explained

Order Management Cloud maps the jeopardy score to a jeopardy priority, and then displays this score in the Order Management work area. This score provides a quick visual cue so that you can take action to reduce the delay.
How Order Management Displays Jeopardy Priority

A jeopardy priority references the fulfillment task that contains the highest jeopardy score. If multiple fulfillment tasks are in jeopardy in an orchestration process, then Order Management uses the highest jeopardy score when it displays the jeopardy for this orchestration process.

For example, assume the supply in the warehouse that the Carpet Processing orchestration process references is not sufficient, and that this insufficient supply causes a delay for the following fulfillment tasks:

- A three day delay for the Deliver Carpet task that results in a jeopardy score of 100 and a jeopardy priority of Medium
- A three day delay for the Invoice Carpet task that results in a jeopardy score of 200 and a jeopardy priority of High

Two hundred is the higher score, so Order Management uses the jeopardy score for the Invoice Carpet task to represent the jeopardy of the Carpet Processing orchestration process. It displays a jeopardy priority of High in the Order Management work area.

How Order Management Assigns Jeopardy Score to Fulfillment Tasks

Order Management assigns a jeopardy score to a fulfillment task according to the jeopardy threshold. If a fulfillment task is delayed, then Order Management calculates the difference between the required completion date and the planned completion date, and then searches for a threshold that applies to the largest number of entities that the fulfillment task references. It uses the following sequence when it searches for this threshold:

1. Search the process name, process version, task name, and task type.
2. Search the process name, process version, and task name.
3. Search the process name and task name.
4. Search the process name, process version, and task type.
5. Search the process name and task type.
6. Search the process task name.
7. Search the process name and process version.
8. Search the process name.
9. Search the task type.

For example, Order Management does the following work:

1. Search for a jeopardy threshold that applies to the task type, task name, process name, and process version. If Order Management does not find a threshold that applies to all of these attributes, then continue to step 2.
2. Search for a jeopardy threshold that applies to the process name, process version, and task name. If Order Management does not find a threshold that applies to all of these attributes, then continue to step 3.
3. And so on.

Order Management continues this work until it finds a threshold that applies to one attribute or a combination of attributes, and then sets the jeopardy score according to the threshold that it assigns to the fulfillment task.

The jeopardy score might change during replanning. If this change results in a change to the jeopardy priority, and if this jeopardy priority value is enabled, then Order Management updates the jeopardy values that it displays.

Related Topics
- Transforming Order Lines to Fulfillment Lines: How it Works
Jeopardy Threshold: Explained

You can use a jeopardy threshold to measure and monitor an orchestration process. You can define a set of ranges for each fulfillment task of an orchestration process, and then assign a score that indicates the severity of the delay. Order Management Cloud uses this information in the indicators that it displays on the Order Management work area. These indicators can help an order manager to quickly determine the severity of a delay, and then to take an appropriate action.

Note the following:

- Order Management uses the lead time of each orchestration process step and the dates from the sales order, such as required completion date, when it plans an orchestration process and assigns it to this sales order.
- Each fulfillment task includes a planned start date and a completion date.
- If a fulfillment task is delayed, then Order Management replans the entire orchestration process.
- If Order Management expects that a fulfillment task will finish after the required completion date of this task, then it assigns a jeopardy score to each fulfillment task according to the jeopardy threshold that you define.

You can define a jeopardy threshold for any combination of the following attributes. If you leave these attributes at their default setting of All, then Order Management applies the jeopardy threshold to all fulfillment tasks:

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

If you apply a jeopardy threshold, then you must first define the orchestration process, fulfillment task, and task type that this threshold must reference so that you can select these objects when you apply the threshold.

Defining Jeopardy and Lead Time to Manage Delay: Procedure

To manage delay, you can define jeopardy priorities for your implementation, and then define lead times and jeopardy thresholds for each orchestration process.

Summary of the Work

To define jeopardy and lead time to manage delay, do the following work:

1. Manage jeopardy priorities.
2. Specify lead time for orchestration process steps.
3. Manage jeopardy thresholds.

This topic describes how to manage jeopardy priorities, lead times, and jeopardy thresholds because you often do this work together. However, you can manage them independently of one another. This topic includes example values. You might use different values, depending on your business requirements.

Manage Jeopardy Priorities

Assume your business requirement specifies a higher range for the MEDIUM jeopardy priority, and a higher but more narrow range for the HIGH jeopardy priority. You can manage these priorities. For details, see Jeopardy Priority: Explained.
Manage jeopardy priorities:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for Manage Jeopardy Priorities, and then open it.
4. On the Manage Jeopardy Priorities page, set the following values, and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Priority</th>
<th>Minimum Score</th>
<th>Maximum Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOW</td>
<td>10</td>
<td>199</td>
</tr>
<tr>
<td>MEDIUM</td>
<td>200</td>
<td>399</td>
</tr>
<tr>
<td>HIGH</td>
<td>400</td>
<td>600</td>
</tr>
</tbody>
</table>

It is recommended that you do not:

- Overlap scores across priorities. For example, do not configure a Low priority of 10 to 210, and a Medium priority of 200 to 400. In this example, an overlap exists between 200 and 210.
- Configure priorities that intersect at the same value. If you configure two priorities that intersect at the same value, then Order Management assigns the intersecting value to the lower priority. For example, if you configure a Low priority of 10 to 200, and a Medium priority of 200 to 400, then Order Management considers a score of 200 as a Low priority.

**Specify Lead Time for Orchestration Process Steps**

Assume that, through research, you determine that your implementation requires a specific amount of lead time to complete each orchestration process step in an orchestration process that ships goods. You can specify these lead times. For details about lead times, see Defining Orchestration Process Steps: Points to Consider.

Specify lead time for orchestration process steps:

1. On the Setup page, search for, and then open Manage Orchestration Process Definitions.
2. On the Manage Orchestration Process Definitions page, search for an orchestration process, and then open it.
   
   For this example, search for Orchestration_Process_1. For details about how to create this process, see Pausing Orchestration Processes Until an Event Occurs: Procedure.
3. On the Step Definition tab, set the Default Lead Time for each of the following steps. Set the Lead Time UOM to Days for each of these steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Default Lead Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule Goods</td>
<td>2</td>
</tr>
<tr>
<td>Reserve Goods step</td>
<td>1</td>
</tr>
<tr>
<td>Ship Goods</td>
<td>4</td>
</tr>
<tr>
<td>Wait for Ship Goods</td>
<td>6</td>
</tr>
<tr>
<td>Prepare Documentation</td>
<td>1</td>
</tr>
</tbody>
</table>
Manage Jeopardy Thresholds

In this example, to meet your business requirements, assume you determine that you can use thresholds on the reservation tasks that your orchestration process performs as a way to manage how long this process can delay each of these tasks. For details, see Jeopardy Threshold: Explained.

Manage jeopardy thresholds:

1. On the Setup page, search for, and then open Manage Jeopardy Thresholds.
2. On the Manage Jeopardy Threshold Definitions page, click Actions, and then click Create.
3. On the Create Jeopardy Threshold Definition page, set the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Enter any text that Order Management can use as an abbreviation for this threshold. Order Management uses this text as a code to identify this threshold throughout the user interface, such as in lists. For this example, enter t-shirt_threshold.</td>
</tr>
<tr>
<td>Name</td>
<td>Clothing Jeopardy</td>
</tr>
<tr>
<td>Description</td>
<td>Jeopardy thresholds for company t-shirts.</td>
</tr>
<tr>
<td>Process Name</td>
<td>Choose the name of the orchestration process where Order Management must apply this jeopardy threshold. For this example, choose Orchestration_Process_1. For details about how to create this process, see Pausing Orchestration Processes Until an Event Occurs: Procedure</td>
</tr>
<tr>
<td>Task Name</td>
<td>Reserve</td>
</tr>
<tr>
<td>Task Type</td>
<td>Reservation</td>
</tr>
</tbody>
</table>

4. Set the threshold for the Low range of the jeopardy priority. Click Actions, click Add Row, and then set the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Delay</td>
<td>2</td>
</tr>
<tr>
<td>Maximum UOM</td>
<td>Days</td>
</tr>
</tbody>
</table>
5. Set the threshold for the Medium range of the jeopardy priority. Click **Actions**, click **Add Row**, and then set the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Delay</td>
<td>5</td>
</tr>
<tr>
<td>Maximum UOM</td>
<td>Days</td>
</tr>
<tr>
<td>Jeopardy Score</td>
<td>Enter a value that resides in the Medium range of the jeopardy priority that you modified above, such as 300.</td>
</tr>
</tbody>
</table>

Description: Threshold for the Medium range of the jeopardy priority.

6. Set the threshold for the High range of the jeopardy priority. Click **Actions**, click **Add Row**, and then set the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Delay</td>
<td>8</td>
</tr>
<tr>
<td>Maximum UOM</td>
<td>Days</td>
</tr>
<tr>
<td>Jeopardy Score</td>
<td>Enter a value that resides in the High range of the jeopardy priority that you modified above, such as 500.</td>
</tr>
</tbody>
</table>

Description: Threshold for the High range of the jeopardy priority.

7. Click **Save and Close**.

**Related Topics**

- Defining Orchestration Processes: Explained
- Pausing Orchestration Processes Until an Event Occurs: Procedure
- Defining Orchestration Process Steps: Points to Consider

### Setting Up Fields
Setting Up Extensible Flexfields in Order Management: Explained

You can use an extensible flexfield to add your own attribute to Order Management Cloud. Define and capture order details that are unique to your business, and then allow other uses for these details during order fulfillment.

You can use the details in an extensible flexfield to determine process automation, send and receive details when communicating with systems outside of Order Management Cloud, and provide criteria for use in a complex business rule. For example:

- Capture consumer details in an extensible flexfield, and then use these details to select and add free samples to a shipment.
- Capture customer loyalty details in an extensible flexfield, and then determine whether to make a follow up phone call to the customer or to upgrade the shipping priority.
- Receive a sales order request from your customer that includes unique build specifications, and then route it to different manufacturing facilities according to the build specifications that you store in an extensible flexfield.
- Select, and then ship a bottle of wine according to the dining preferences of each of your customers. You can store these preferences in an extensible flexfield.

You can use an extensible flexfield to gather, organize, and display your own order details when the predefined data model does not contain these details. It is similar to a descriptive flexfield in that it is a field that you can set up and that contains segments. Note the following points:

- An extensible flexfield supports a one-to-many relationship between one entity and one or more attributes. You can use it to add multiple segments that are context sensitive.
- You can set up an extensible flexfield for a fulfillment line, or on some other object that supports an extensible flexfield.

Examples of Using Extensible Flexfields

Consider the following examples of using extensible flexfields in Order Management.

<table>
<thead>
<tr>
<th>Example Usage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get details from a source system.</td>
<td>Get source order details from one or more source systems. A source order contains a set of attributes. If you require more details or more attributes, then you can use an extensible flexfield to get them, and then use these details during order fulfillment. You can use the same extensible flexfield attributes to receive details from each source system, or use different extensible flexfield attributes according to the unique requirements of each source system.</td>
</tr>
<tr>
<td>Send details to a fulfillment system.</td>
<td>Order Management sends a fulfillment request that includes a predefined set of attributes to a fulfillment system. You can use an extensible flexfield to send details that these attributes do not include, but that the fulfillment system requires to complete the fulfillment request.</td>
</tr>
<tr>
<td>Receive details from a fulfillment system.</td>
<td>A fulfillment system might send attributes that provide some business value, and that the Order Entry Specialist must view in Order Management or in the source system. In some situations, Order Management might also use these details in the next set of tasks that it runs in an orchestration process. You can use an extensible flexfield to receive these attributes.</td>
</tr>
</tbody>
</table>
### Example Usage

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an extensible flexfield attribute with the following types of business rules:</td>
<td></td>
</tr>
<tr>
<td>• Transformation rule. Write a rule that references an extensible flexfield to add order details, delete unnecessary details, or modify details. Use an extensible flexfield to determine the transformations to perform.</td>
<td></td>
</tr>
<tr>
<td>• External interface routing rule</td>
<td></td>
</tr>
<tr>
<td>• Assignment rule</td>
<td></td>
</tr>
</tbody>
</table>

For details, search for Document ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, download the attachment that describes how to use business rules, and then see that chapter that describes extensible flexfields.

<table>
<thead>
<tr>
<th>Manage change.</th>
<th>Use an extensible flexfield as an order attribute to store details about a change that occurred. For example, use an extensible flexfield as input to calculate the cost of change and to select a compensation pattern when Order management receives a change order.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Display attributes in the Order Management work area.</th>
<th>Allow the Order Entry Specialist to search for sales orders in the Order Management work area according to the value of an extensible flexfield.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Note that the Order Entry Specialist can read but not edit an extensible flexfield attribute in the Order Management work area.</td>
</tr>
</tbody>
</table>

### Examples of Using Extensible Flexfields with Orchestration Processes

Consider the following examples of using extensible flexfields with an orchestration process.

<table>
<thead>
<tr>
<th>Example Usage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign an orchestration process.</td>
<td>Use the value of an extensible flexfield in an assignment rule to automatically assign an orchestration process during order fulfillment to fulfill each sales order and each order line.</td>
</tr>
<tr>
<td></td>
<td>Use an extensible flexfield as part of the selection criteria during assignment.</td>
</tr>
<tr>
<td>Set up an orchestration process.</td>
<td>Create business rules that determine branching and that determine the tasks to run in an orchestration process.</td>
</tr>
<tr>
<td>Select the fulfillment lines to process.</td>
<td>Determine whether a task in an orchestration process will process the fulfillment line or ignore it during order fulfillment.</td>
</tr>
<tr>
<td>Calculate the lead time that an orchestration process step requires.</td>
<td>Use an extensible flexfield to calculate lead time according to your data. For example, create an extensible flexfield named Engraving that includes a Yes or No value, and then define the following conditions:</td>
</tr>
<tr>
<td></td>
<td>• If the item is silver cup, and if engraving is yes, then set the lead time to 3 days.</td>
</tr>
<tr>
<td></td>
<td>• If the item is silver cup, and if engraving is no, then set the lead time to 1 day.</td>
</tr>
</tbody>
</table>

For details, see Defining Jeopardy and Lead Time to Manage Delay: Procedure.
Objects That Support Extensible Flexfields

You can use an extensible flexfield with the following objects on a sales order:

- Header
- Order line
- Fulfillment line
- Fulfillment line detail
- Price adjustment
- Sales credit
- Payment
- Lot serial
- Activity

Deploying Extensible Flexfields in Order Management: Procedure

This topic describes how to publish and deploy an extensible flexfield in Order Management Cloud.

For details about how to configure Order Management to use an extensible flexfield with a source system, how to use an extensible flexfield in a business rule, or how to use an extensible flexfield with a fulfillment system, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then open the attachment that describes how to use extensible flexfields.

Summary of the Work

Publish Extensible Flexfield Attributes

You must publish extensible flexfield attributes into the business rules and processing constraints that Order Management uses. Publishing allows you to reference these attributes from rules and constraints.

You can publish only the extensible flexfield attributes that come predefined with Order Management and that are part of the following predefined extensible categories. The Category Hierarchy and Preconfigured Context Values are not available for these extensible flexfields.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td>DOO_ HEADERS_ ADD_INFO</td>
</tr>
<tr>
<td>Lines</td>
<td>DOO_ LINES_ ADD_INFO</td>
</tr>
<tr>
<td>FulfillLines</td>
<td>DOO_ FULFILL_ LINES_ ADD_INFO</td>
</tr>
<tr>
<td>Payments</td>
<td>DOO_ PAYMENTS_ ADD_INFO</td>
</tr>
<tr>
<td>PriceAdjustments</td>
<td>DOO_ PRICE_ ADJUSTMENTS_ ADD_INFO</td>
</tr>
<tr>
<td>SalesCredits</td>
<td>DOO_ SALES_ CREDITS_ ADD_INFO</td>
</tr>
</tbody>
</table>
Publish extensible flexfield attributes:

1. Log in with a user who possesses the DOO_ORDER_ADMINISTRATOR job role.

   The scheduled process that you run requires that you use this role.

2. In the Navigator, click Scheduled Processes.

3. On the Scheduled Processes page, click Schedule New Process, and then run the scheduled process named Publish Extensible Flexfield Attributes.

4. Click Actions, click Refresh, and then verify the following value. It might be necessary to wait a few minutes and refresh again.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Succeeded</td>
</tr>
</tbody>
</table>

5. In the Navigator, click Setup and Maintenance.


7. Search for, and then open Manage Order Extensible Flexfields.

8. On the Manage Order Extensible Flexfields page, click Search, and then verify that the search results contains a list of extensible flexfields.

Deploy Extensible Flexfield Attributes

Deploy extensible flexfield attributes:

1. In the Search Results, choose the extensible flexfield you must deploy, click Actions, and the click Edit.

2. On the Edit Extensible Flexfield page, define contexts.

3. A context determines the attributes to display according to the criteria that you specify. A context is part of a hierarchy of attributes, where the entity and entity category represent the top level of the hierarchy, contexts represent the lower levels, and attributes represent the lowest level.

4. Use the Pages tab of the Categories: Details area to define pages and to associate contexts to Order Management pages.

5. Use the Manage Order Extensible Flexfields page to deploy each of your extensible flexfields.

6. Log out of Order Management, and then log back in to Order Management.

   If you create a new segment, create a new context, change the default on a segment, and so on, then you must perform this step so that Order Management can update your changes throughout the Order Management work area.
Managing Order Types: Procedure

An order type is an optional attribute on the order header that you can administer to arrange sales orders into groups that you define. You can also use it to control order processing.

Note the following points:

- The Order Entry Specialist can search for sales orders on the Manage Orders page according to order type. This capability focuses the search and can help to improve search productivity.
- The Manage Orders page does not come predefined to display the Order Type attribute in the search area. The Order Entry Specialist must click Add Fields, and then add it to the search area.
- If you import a source order from a source system, then Order Management Cloud will examine the value of the Order Type attribute in the source order to make sure it matches one of the values that you define for the lookup. If it does not match, then Order Management will reject the import and log an error.

Using Order Types to Control Processing

You can use the Order Type to control processing.

**Using Order Types in Rules**

Use the Order Type in the following business rules.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product transformation</td>
<td>Use the order type as part of a pretransformation, transformation, or posttransformation rule. For example:</td>
</tr>
<tr>
<td></td>
<td>• Write a rule that sets the default value for the order type according to the value of some other attribute on the order header. For example, assume your company writes a contract to fulfill sales orders from Customer Y over all other customers. You can write a rule if Customer is Y, then set Order Type to Top Priority, and then use an external interface routing rule that routes the sales order to a fulfillment system that prioritizes shipment speed over other factors, such as cost.</td>
</tr>
<tr>
<td></td>
<td>• Write a rule that uses the value of the order type to determine the default value to display for an attribute on the order header or order line.</td>
</tr>
</tbody>
</table>

Order Management runs each pretransformation rule when you create or revise a sales order in the Order Management work area. For example, when you set or update the Business Unit, Customer, or Order Type, or add an order line. To avoid overwriting these selections, you can write a pretransformation rule so it sets a value only if the attribute is empty.

<table>
<thead>
<tr>
<th>Process assignment</th>
<th>Write a rule that uses the order type to assign an orchestration process. For example, if Order Type is Government - Department of the Interior, then set up an orchestration process that fulfills order lines according to Department of the Interior procurement requirements.</th>
</tr>
</thead>
<tbody>
<tr>
<td>External interface routing</td>
<td>Write a rule that uses the order type to determine which fulfillment system will fulfill the sales order. For example, if Order Type is Emergency, then route the sales order to a fulfillment system you have defined that prioritizes shipment speed over other factors.</td>
</tr>
</tbody>
</table>

Using Order Types in Processing Constraints
You can use the order type in the condition or result of a processing constraint. For example, prevent Order Management from updating an attribute according to the value of some other attribute. You can also use the order type to prevent Order Management from submitting a sales order. For example:

- If Order Type is x, then do not allow change to attribute y after submit.
- If Business Unit is x, then do not allow change to Order Type after submit.
- If a submit validation occurs, and if Order Type is empty, then raise an error.
- If a submit validation occurs, and if Order Type is x, and if Purchase Order is empty, then raise an error.

### Setting Up Order Types

In this example, you set up four order types so you can differentiate between sales orders in a deployment that uses drop ship.

This topic includes example values. You might use different values, depending on your business requirements.

Set up the order types:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Order Lookups.
4. On the Manage Order Lookups page, in the Search area, enter the following value, and then click **Search**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Code</td>
<td>ORA_DOO_ORDER_TYPES</td>
</tr>
</tbody>
</table>

Note that Order Management uses the ORA_DOO_ORDER_TYPES lookup for the Order Type attribute, but does not come predefined with values for this lookup. If you do not add any values, then Order Management will still display the Order Type attribute but this attribute will not contain any values.

5. In the Lookup Codes area, click **Actions**, and then click **New** to add each of the following lookup codes.

<table>
<thead>
<tr>
<th>Lookup Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>Standard sales order</td>
</tr>
<tr>
<td>Drop_ship</td>
<td>Drop ship sales order</td>
</tr>
<tr>
<td>Mixed</td>
<td>Sales order that includes standard and drop ship order lines</td>
</tr>
<tr>
<td>Return</td>
<td>Return sales order</td>
</tr>
</tbody>
</table>

You can modify the other attributes, as necessary. For details, see Lookups: Explained.

6. Click **Save and Close**.

**Related Topics**

- Lookups: Explained
• Using Visual Information Builder to Create Rules: Explained

• Assigning Orchestration Processes: Explained

• Routing Requests to Fulfillment Systems: Explained
6 Setting Up Features

Defining Extensions That Modify Order Management Cloud: Overview

You can use the Manage Extensions page in the Setup and Maintenance work area to write your own Groovy script that modifies your Order Management Cloud deployment, and to specify the extension point that determines when to run this script.

An order management extension is a Groovy script that runs at an extension point during order creation. An extension point is an event that you can specify that starts the extension. You can use a flexfield to modify the data model, or Page Composer to modify the user interface, but you cannot use them to programmatically modify logic or to modify your order management implementation. Instead, you can use an order management extension.

For details, see Defining Extensions That Modify Order Management Cloud (Doc ID 2252683.1) on My Oracle Support at https://support.oracle.com.

Related Topics
- Defining Extensions That Modify Order Management Cloud (Doc ID 2252683.1)

Managing Credit, Payment, and Accounting

Managing Credit Check in Order Management: How It Works

You can use credit check in Order Management Cloud to minimize the financial risk that your organization assumes during day-to-day operations. You can use credit check to validate each sales order and verify that your customer has credit available that is sufficient to cover the cost of the purchase, and to allow Order Management to process the sales order through order fulfillment.

Order Management can perform a credit check on each sales order or on each order line, and can hold an order line that does not pass the credit check. Order Management uses various Oracle Cloud Applications to check credit, such as Oracle Fusion Pricing, Oracle Fusion Shipping, Oracle Fusion Receivables, and Oracle Fusion Credit Management.
Order Management uses the following flow during check credit.

**Explanation of Callouts**

Order Management performs the following steps during check credit:

1. The Order Entry Specialist creates a sales order in Order Management, adds order lines, sets the payment terms, payment method, and Bill-to Customer.
2. Pricing calculates the price, tax, and shipping charges.
3. The Order Entry Specialist clicks Submit.
4. If the order management parameter Activate Credit Check on Order Submit is:
   - Yes. Order Management sends the sales order to Receivables.
Oracle SCM Cloud
Implementing Order Management
Chapter 6

Setting Up Features

5. Receivables performs a credit check that returns one of the following values:
   - Authorized. Receivables authorizes the credit, sets the Authorization Status to Open, and then sends the credit authorization number, credit authorization amount, and credit expiration date to Order Management.
   - Denied. Receivables sends the credit check result to Order Management.

6. Order Management routes the sales order depending on the following value of the order management parameter Credit Check Failure at Order Submit:
   - Save Order in Draft Status. Saves the sales order in Draft status and does not proceed to fulfillment.
   - Submit the Order with Hold on Lines That Failed Credit Check. Saves the sales order in Submitted status, places a hold on each order line that failed credit check, and then sends the order lines that are not on hold to Shipping. This feature is useful when your sales order includes multiple Bill To Customer values on the order lines. For example, if credit check for Bill To Customer x succeeds, and if credit check for Bill To Customer y fails, then Order Management can send order lines for customer x to order fulfillment, and place order lines for customer y on hold. If the order lines include only one Bill To Customer value, then Order Management uses the order total when it checks credit.

7. Shipping ships the order lines, and then sends the shipping result to Order Management.
   
   As an option, you can enable credit check to occur during order fulfillment before shipping occurs. For details, see Managing Credit Check in Order Management: Procedure.

8. Order Management updates the sales order status, and then sends details about the order lines that shipped to Receivables, including the credit authorization number.

9. Receivables creates the invoice and sets the attribute Open Credit Authorization Against This Order to closed.

Note the following points:

- Credit check establishes a credit line only for a single currency, and only for the customer account.
- Order Management runs the credit check when the user submits the sales order, and immediately before running the approvals. If credit check succeeds, and if approval fails, then Order Management removes authorization for the authorization number, and then gets a new authorization number after the user resubmits the sales order.
- If the Order Entry Specialist cancels a sales order before fulfillment finishes, then Order Management communicates with Credit Management to reverse the credit amount of the sales order. It reverses this amount in the customer account in Oracle Fusion Financials.
- For details about the parameters that Order Management uses during credit check, see Managing Order Management Parameters: Procedure.

Calculating Available Credit

Credit Management comes predefined to use the following equation when it computes the available credit:

- Credit limit, minus the open balance, minus the total outstanding authorization, equals the available credit.

Credit Management performs the following calculations:

- Uses invoice details to determine the open balance
- Sums the amounts of an authorization to determine the total outstanding authorization

For example, assume a customer account contains the following values:

- Credit limit is $2,000
- Open balance is $0
Outstanding authorization is $500

Customer orders an item that costs $1,000

In this situation, Credit Management performs the following calculation:

- $2,000 credit limit, minus $0 open balance, minus $500 outstanding authorization, equals $1,500 available credit.
- $1,500 available credit exceeds the $1,000 item cost, so Credit Management authorizes the purchase, and then sends an authorization number, expiration date, authorization amount of $1,000, and the available credit to Order Management. Credit Management uses the same currency that the sales order references.

You can use Credit Management to define a different formula that more closely meets your business requirements, such as modifying the credit limit, including shipping charges and taxes in the formula, and so on. For details, see Oracle Credit Management User Guide on the Oracle Help Center.

Related Topics

- Releasing Holds for Order Lines That Fail Credit Check: Procedure
- Managing Credit Cards and Payment Authorizations: Procedure
- Managing Order Management Parameters: Procedure

Managing Credit Check in Order Management: Procedure

To manage credit check in Order Management Cloud, you must perform a few setup tasks, such as making sure the customer record meets requirements, verifying the customer account setup, and so on.

Manage credit check in Order Management:

1. Make sure the customer record meets the following requirements:
   - Exists in Oracle Fusion Financials.
   - Contains a billing account that Order Management can reference.
   - The billing account has an established credit line.

2. Verify the customer account setup:
   - Log into Order Management with administrative privileges.
   - In the Navigator, click Setup and Maintenance.
   - On the Setup page, click Order Management.
   - Search for, and then open Create Customer.
   - On the Organization Information page, search for the customer account you must verify.
   - Click Profile History, and then verify the attributes that affect credit, such as Credit Line and Credit Currency.

   For details about how to use Credit Management in Oracle Fusion Financials to enable the customer for credit check, set up the customer credit profile, and so on, see Oracle Credit Management User Guide.

3. Set up the order management parameters named Activate Credit Check on Order Submit and Credit Check Failure at Order Submit.

   For details, see Managing Order Management Parameters: Procedure.
4. Optional. Enable credit check to occur during order fulfillment before shipping occurs.

### Enabling Credit Check to Occur During Order Fulfillment

In some situations, order fulfillment might require days to complete because of a variety of factors. For example, inventory is out of stock for the item and requires several days to replenish. The credit that is available for a customer might decrease during this time because the customer continues to make other purchases. You can enable credit check to occur during order fulfillment immediately before shipping. This functionality allows you to make sure the credit check is accurate and up to date so that you do not ship an item when the customer credit no longer covers the purchase amount.

For example, you can add a credit check step to an orchestration process immediately after the Create Reservation step, and immediately before the Create Shipment Request step. Use the following values for the credit check step that you add to your orchestration process.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>You can enter any text. It is recommended that you enter text that describes the step, such as Credit Check.</td>
</tr>
<tr>
<td>Task Type</td>
<td>DOO_CreditCheck</td>
</tr>
<tr>
<td>Task</td>
<td>DOO_CreditCheck</td>
</tr>
<tr>
<td>Service</td>
<td>Create Credit Check Request</td>
</tr>
</tbody>
</table>

This credit check runs in the same way as any other credit check. For example, if it fails, then it puts the order lines on hold and pauses the orchestration process.

If you add a credit check step to an orchestration process, and if this orchestration process already performed a credit check for the sales order before order submit, then Order Management checks the expiration date of the credit check that it already performed. If the current date:

- **Occurs after the expiration date.** Order Management performs the credit check.
- **Occurs on or before the expiration date.** Order Management does not perform the credit check.

For details, see Defining Orchestration Processes: Explained.

**Related Topics**

- Managing Order Management Parameters: Procedure
- Defining Orchestration Processes: Explained

### Importing Source Orders That Include Credit Check Details: Explained

You can control how Order Management Cloud handles credit check for each source order that you import from a source system.

Use the following attributes in the SOAP payload that you use to import your source order.
### Attribute: PreCreditCheckedFlag

Contains one of the following values:

- **true.** Do not perform credit check. Order Management will not perform credit check on the source order even if the credit check is enabled for the business unit.
- **false.** Perform credit check.

### Attribute: CreditCheckAuthorizationCode

Any numeric value, such as 9090.

Order Management uses the value that you provide to populate the Credit Reference attribute of the order line.

If you import a source order, and if a sales order already exists in Order Management that matches this source order, then Order Management examines the value of the Credit Reference attribute of the sales order. If the value in the Credit Reference attribute is not equal to the value of CreditCheckAuthorizationCode, then Order Management runs a credit check for the source order.

### Attribute: CreditCheckAuthorizationExpiryDate

Date value, such as 2016-10-05T10:00:00Z.

Order Management uses the value that you provide to populate the Expiration Date attribute of the order line.

If the current date:

- **Occurs after CreditCheckAuthorizationExpiryDate.** Order Management performs the credit check.
- **Occurs on or before CreditCheckAuthorizationExpiryDate.** Order Management does not perform the credit check.

### Related Topics

- Managing Credit Cards and Payment Authorizations: Procedure
- Order Import Template: Explained

### Specifying How to Track Items as Assets: Procedure

You can specify how to track an item as an asset in Order Management Cloud.

Specify how to track an item as an asset:

1. In the Navigator, click **Product Management**, and then click **Product Information Management**.
2. On the Product Information Management page, click **Tasks**, and then click **Manage Items**.
3. Locate the item you must modify, and then open it for editing.
4. On the Edit Item page, click **Specifications**, and then click **Service**.
5. Set the following value.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>◦ Customer Asset</td>
</tr>
</tbody>
</table>

For details about these values, see Monitoring Order Fulfillment: Procedure.

6. Click **Sales and Order Management**, and then set one of the following values, or leave it empty.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales Product Type</td>
<td>Use one of the following values:</td>
</tr>
<tr>
<td></td>
<td>◦ Goods</td>
</tr>
<tr>
<td></td>
<td>◦ Software</td>
</tr>
</tbody>
</table>

If Sales Product Type contains one of the following values, then Oracle Fusion Installed Base ignores the item even if you set Enable Asset Tracking to Customer Asset or Full Lifecycle:

- Included Warranty
- Extended Warranty
- Service Level Agreement
- Software Maintenance
- Preventive Maintenance
- Installation
- Training
- Subscription
- One Time Service

7. Optional. The DOO_AssetManagement task type allows Order Management to send a request to Oracle Fusion Installed Base to create or update an asset. You can define your own task that the DOO_AssetManagement task type references, and then reference your task from various steps in different orchestration processes that you define. For details about task types, see Task Types: Explained.

Define your own task:

- In the Navigator, click **Setup and Maintenance**.
- On the Setup page, click **Order Management**.
- Search for, and then open Manage Task Types.
- On the Manage Task Types page, click **View**, click **Query by Example**, enter the following value, and then click the **Enter key** on your keyboard.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Type</td>
<td>DOO_AssetManagement</td>
</tr>
</tbody>
</table>

- In the DOO_AssetManagement Details area, click **Tasks**, click **Actions**, click **Add Row**, set the following values, and then click **Save and Close**.
Oracle SCM Cloud
Implementing Order Management

Chapter 6
Setting Up Features

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Create_Vision_Asset</td>
</tr>
<tr>
<td>Name</td>
<td>Create_Vision_Asset</td>
</tr>
<tr>
<td>Display Name</td>
<td>Create Vision Asset</td>
</tr>
</tbody>
</table>

8. Create an orchestration process or revise the copy of an existing one:

   - Add a step that references the following task type. Add this step after the shipment step and before the invoicing step. For example, if you copy, and then modify the copy of the predefined orchestration process DOO_OrderFulfillmentGenericProcess, then add this step immediately before the Create Invoice step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Type</td>
<td>DOO_AssetManagement</td>
</tr>
</tbody>
</table>

   - If you defined your own task earlier in this procedure, then set the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Create Vision Asset</td>
</tr>
</tbody>
</table>

For details about how to define an orchestration process, see Defining Orchestration Processes: Explained.

**Related Topics**

- Defining Orchestration Processes: Explained
- Monitoring Order Fulfillment: Procedure
- Task Types: Explained

Managing Sales Order Approvals

Managing Approval Rules for Sales Orders: How It Works

You can configure Order Management Cloud to route each sales order for sales order approval. You can configure sales order approval to accept or reject the sales order according to a variety of approval rules, and on a variety of sales order attributes.

Order approval allows you to streamline policy enforcement, such as to streamline communication between the Order Entry Specialist and one or more approvers, and to improve the sales order experience for order entry and approval personnel. Approving each sales order before starting order fulfillment can minimize the sales orders that Order Management must...
return to Draft status because they do not meet fulfillment requirements for some reason, which in turn reduces rework after fulfillment starts.

Sales order approval includes at least one Order Entry Specialist and one or more approvers. The following diagram illustrates how order approval works.

**Explanation of Callouts**

Order approval includes the following steps:

1. An Order Entry Specialist creates a sales order, and then clicks Submit.

   Alternatively, Order Management receives a source order from a source system.
2. If the Source Order System value in the order management parameter named Start Approval Process for Sales Orders matches the source order system attribute of the sales order, and if Source Order System is Yes, and if rules are defined for approval:
   - Then. Order Management sets the order status to Approval Pending, and then routes the sales order for approval.
   - Else. Order Management does not evaluate any order approval rules, even if defined, and proceeds to step 5.

3. Approval rule logic that you define on the Manage Order Approval Rules page routes the sales order through the approval process, such as assigning and routing the sales order to an order manager. For details, see Managing Order Approval Rules: Procedure.

4. The approver uses their Pending Notifications page or worklist to view and approve the approval request.
   - If the approver rejects the approval request, then Order Management sets the sales order status to Draft, unlocks it, routes it back to the person who created the sales order, and displays a message on the Order page that describes the rejection. The creator can use the Approval Notes tab on the sales order to get approval details, including suggested actions that the approver entered in comments when rejecting the approval request, such as Split the sales order into two separate orders so that the total amount for each sales order does not exceed $10,000.

5. Order Management sets the order status to Processing, and then sends it to order fulfillment.

6. Order fulfillment fulfills the sales order.

Note the following points:

- You can define approval conditions according to the value of an order header attribute, order line attribute, order total, order charge, price adjustment, or validation.
- You can use approvals only before Order Management submits the sales order to order fulfillment. You cannot use approvals during order fulfillment.
- Order Management approves only the entire sales order. It cannot approve an order line.
- Order Management does not come with predefined approval rules. If you require approval, then you must create an approval rule.
- If you must create an approval group, then it is recommended that you use the Manage Approval Groups for Supply Chain Management page in the Setup and Maintenance work area.
- It is recommended that you define your approval rule so the rule conditions evaluate to True only one time for the same sales order, and then creates only one approval request. For example, a sales order header includes more than one header total. The Order Management work area displays only the header total that it specifies as the primary total. Assume you create an approval rule that filters only according to Order Total Amount is greater than 10,000. At runtime, the rule will create a separate approval request for each order total that is greater than 10,000 although you require only one approval request for the sales order. To avoid this situation, you can add another condition that filters approval according to Primary equals Yes for the header total. This way, the rule evaluates only the primary order total, ignores the other order totals, and sends only one approval request.

Note the following points regarding trade compliance:

- You can define a condition for trade compliance, such as trade compliance status, screening type, compliance type, result, or reason.
- If you require approval for trade compliance, and if the trade compliance status is Hold or Failed, then, as an option, you can route the sales order for approval. It is a business decision whether or not to reject the sales order or release it to order fulfillment. For example, if the hold occurs because of a licensing effective date, then the Order Manager can choose to release the sales order to fulfillment, and then manage the license problem at some later time.
Managing Approval Rules for Sales Orders: Procedure

You can create an approval rule that uses IF THEN logic to create a condition that routes a sales order to one or more approvers.

Assume you must create a routing rule that implements the following condition:

- If the order total is greater than $10,000, and if the Primary is yes, and if the customer is Computer Service and Rentals, then request and get approval from a manager.

In this example, you create the following rule:

**Summary of the Work**

Manage approval rules for sales orders:

1. Create the approval rule and the IF.
2. Create the first AND.
3. Create the second AND.
4. Create the DO.
5. Activate and publish your rule.
6. Test your work.
This topic includes example values. You might use different values, depending on your business requirements.

Create the Approval Rule and the IF

Create the approval rule and the IF condition:

1. Log into Order Management with administrative privileges.
2. Set the order management parameter named Start Approval Process for Sales Orders to Yes for all source systems. For details, see Managing Order Management Parameters: Procedure.
3. In the Navigator, click Setup and Maintenance.
5. Search for, and then open Manage Order Approval Rules.
7. Change the rule name.

<table>
<thead>
<tr>
<th>Current Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Rule</td>
<td>Get Approval for High Value Sales Orders</td>
</tr>
</tbody>
</table>

8. Name the If statement so that it reflects the meaning of the condition:
   - Click Enter Description.
   - In the Enter Description dialog, enter High Value Sales Orders, and then click OK.
9. Click New Condition (the dashed circle in the IF area).
10. In the Create Condition dialog, enter order total, wait a moment, and then click Order Total Amount (Order Header > Order Totals).
    Notice that the dialog displays suggestions when you wait a moment after you finish typing.
11. In the Create Condition dialog, set the following values, and then click OK.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Order Total Amount</td>
</tr>
<tr>
<td>Condition</td>
<td>greater than ( &gt; )</td>
</tr>
<tr>
<td>Value</td>
<td>10000</td>
</tr>
</tbody>
</table>

Notice that the IF area now displays the following condition:
   - Order Total Amount > 10,000
12. Click Save.

Create the First AND

Create the first AND statement:

1. Click AND.
2. In the Create Condition dialog, enter primary, wait a moment, and then click Primary (Order Header > Order Totals).
3. In the Create Condition dialog, set the following values, and then click **OK**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Primary</td>
</tr>
<tr>
<td>Condition</td>
<td>Is equal to</td>
</tr>
<tr>
<td>Value</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Notice that the IF area now displays the following condition:

- **Primary is equal to Yes**

4. Click **Save**.

**Create the Second AND**

Create the second AND statement:

1. Below **Primary is equal to Yes**, click **AND**.
2. In the Create Condition dialog, enter **customer**, wait a moment, and then click **Customer (Order Header)**.
3. In the Create Condition dialog, set the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Customer</td>
</tr>
<tr>
<td>Condition</td>
<td>Equals ( =)</td>
</tr>
</tbody>
</table>

4. Click **Search**.
5. In the Search dialog, in the Name field, enter **Computer Service and Rentals**, and then click **Search**.
6. In the search results, click **Computer Service and Rentals**, and then click **OK**.
7. In the Create Condition dialog, click **OK**.

Notice that the IF area now displays the following condition:

- **Customer = Computer Service and Rentals**

8. Click **Save**.

**Create the DO**

Create the DO statement:

1. Click **THEN**, and then click **DO**.
2. Name the DO so that it reflects the meaning of the action:
   - Click **Enter Description**.
   - In the Enter Description dialog, enter **Assign to Order Manager**, and then click **OK**.
3. Click **New Action** (the dashed circle in the DO area), and then click **Perform an Action**.
4. In the Create Action dialog, set the action to **Assign to Job Level**.
5. In the Create Action dialog, verify the following values.
### Field | Value
--- | ---
Action | Assign to Job Level
Required Action | Approval Required
Approval Chain Based On | Task Creator
First Approver | Manager
Minimum Job Level of Top Approver | 1
Utilized Approvers | All Levels

6. Click **Search**.

7. In the Search dialog, next to Users, enter **manager**, and then click **Search**.

8. In the Search for Highest Possible Approver dialog, in the manager row, click the **radio button**, and then click **OK**.

9. Verify that Highest Possible Approver contains **manager**, and then click **OK**.

Notice that the DO area now displays the following condition:

- Assign to Job Level

10. Click **Save and Close**.

#### Activate and Publish Your Rule

Activate and publish your rule:

1. On the Manage Order Approval Rules page, notice that the rule **Get Approval for High Value Sales Orders** is grey.

2. Click **Get Approval for High Value Sales Orders**.

3. In the Get Approval for High Value Sales Orders dialog, add a check mark to the Activate Rule option, and then click **Save and Close**.

Notice that the Active indicator for Get Approval for High Value Sales Orders is green.

4. Click **Publish**.

Order Management publishes the active rules that display on the Manage Order Approval Rules page.

#### Test Your Work

Test your work:

1. Log into Order Management with the Order Entry Specialist job role, create a sales order that includes the following values, and then click **Submit**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>Computer Service and Rentals</td>
</tr>
<tr>
<td>Amount</td>
<td>Add items to an order line so that the Amount exceeds $10,000.</td>
</tr>
</tbody>
</table>
2. Verify that the order status updates to Approval Pending.
3. Click **View Approval Information**, and then verify the approval information reflects the approval rule that you created.

For example:

4. In a separate browser session, log into Order Management with the Order Manager job role.
5. Open your Pending Notifications page, locate the link for the approval request, such as **Action Required: Approval Request for Order 258093**, and then click **Approve**.
6. Navigate to the browser session for the Order Entry Specialist, refresh the page, and then verify that the order status is now Processing.

**Related Topics**
- Getting Approvals for Sales Orders: Explained
- Managing Order Management Parameters: Procedure

**Managing Trade Compliance**
Managing Trade Compliance: Overview

Some governments and companies impose rules and regulations for trade with different countries, people, companies, financial institutions, and so on. You can use Order Management Cloud in conjunction with your trade management solution to create and manage trade transactions that comply with import global trade rules, export global trade rules, and other trade regulations that the country or region requires.

You can use trade compliance to achieve the following results:

- Manage legal, regulatory, and corporate requirements for transactions that cross governmental borders, such as across states, provinces, regions, or countries, according to the unique requirements of each country, region, company, and so on. For example, you can set up trade controls that apply United States rules and Chinese rules on each transaction that occurs between a company that resides in the United States and another company that resides in China.

- Apply trade controls on various items, such as munitions, computer equipment, licenses, license exceptions, documents, registrations, and so on.

- Manage trade compliance policy, hold transactions until they clear trade compliance screening, and so on.

The following diagram illustrates an example of how you can use your trade management solution with Order Management to implement a supply chain that imports raw materials from multiple sources into a factory in the United States, sends a partially completed assembly to a factory in Europe, and then sends the final assembly to distribution centers in multiple countries in Africa.
Each import and export point might require a different set of trade compliance rules for each transaction. For example, you can manage compliance according to the following trade compliance policies:

- Product classification, such as weapons or dangerous chemicals
- Export and import embargoes
- Status on the Denied and Restricted Parties List
- Trade agreements, such as NAFTA (North American Free Trade Agreement)
- Restricted party

-related-topics

- Creating Sales Orders That Require Trade Compliance Screening: Procedure

Managing Trade Compliance: How It Works

Managing a sales order for trade compliance includes multiple job roles, such as Order Entry Specialist, Order Manager, Compliance Manager, and Warehouse Manager.

The following diagram illustrates an example flow that Order Management Cloud can use during trade compliance screening. This example uses Oracle Global Trade Management as the trade compliance solution. You can use your own trade compliance solution. As an option, it also identifies the points where you can screen for trade compliance during order
fulfillment in Oracle Fusion Shipping. For details about Global Trade Management and how to set it up, see https://www.oracle.com/applications/supply-chain-management/solutions/logistics/global-trade-management.html.

**Explanation of Callouts**

Order Management does the following work to manage trade compliance:

1. An Order Entry Specialist creates a sales order in the Order Management work area, and then clicks Submit. Order Management can also import a source order from a source system.

   If the order management parameter named Check for Trade Compliance When User Submits Sales Order is Yes, then Order Management sends a request to Global Trade Management to screen the sales order for trade
compliance. Note that Global Trade Management can screen for a nonconfigured item, configured item, or change order.

2. Global Trade Management does the following work:
   - Applies the trade compliance policies that you define for this integration to the sales order, and then creates a screening result. Global Trade Management supports a variety of compliance screening types. For example, restricted party, sanctioned countries or territories, and trade control.
   - Sets the trade compliance status to Passed, Under Review, or Failed, according to the screening result. You cannot modify this value.
   - If the trade compliance status is Failed or Under Review, then provides the screening failure reason for each order line that fails screening.
   - Sends the screening result to Order Management.

Note that this integration cannot screen a return line.

3. Order Management updates the trade compliance status of the sales order according to the most restrictive trade compliance status of the order lines that the sales order contains.

For example, if the status for one order line is Failed, then the status for the sales order is Failed. Order Management uses the following hierarchy to determine which status is most restrictive, where 1 is least restrictive, and 3 is most restrictive:

1. Passed
2. Under Review
3. Failed

Note the following points:
   - Global Trade Management typically finishes screening and sends the results to Order Management without delay. However, Global Trade Management might require a few minutes to screen a large or complex sales order.
   - If the sales order has not yet moved to order fulfillment, then the Order Entry Specialist can click Actions, and then click Revert to Draft. Order Management will set the order status to Draft and stop the screening.

4. If screening fails, then Order Management sends the sales order back to the Order Entry Specialist because a predefined processing constraint prevents Order Management from submitting a sales order that includes a trade compliance exception.

You can disable this constraint or create a new one that allows the sales order to proceed according to a condition. For details, see Constraining Compliance Screening: Procedure.

5. If screening does not fail, then Order Management sends the sales order to order fulfillment.

If the constraint allows the sales order to proceed according to a condition, then, as an option, you can define an approval rule that uses the trade compliance result of the order header or order line to route the sales order to an Order Manager for approval. For details, see Managing Approval Rules for Sales Orders: How It Works.

Managing Trade Compliance: Procedure

Order Management Cloud comes predefined with trade compliance screening disabled. If your implementation requires trade compliance screening to occur before sending the sales order to order fulfillment, then you must set it up.
For details about how to set up trade compliance screening to occur during order fulfillment, see Managing Trade Compliance During Order Fulfillment: Procedure.

Set up trade compliance screening to occur before sending the sales order to order fulfillment:

1. **Configure an integration between Order Management and your trade management system.**
   
   You must use Integration Cloud Service. For details, see Using Integration Cloud Service to Integrate Order Management with Other Systems: Overview.

2. **Set the order management parameter named Check for Trade Compliance When User Submits Sales Order to Yes.**
   
   For details, see Managing Order Management Parameters: Procedure.

3. **Optional. Perform the following work, according to your business requirements:**
   
   ◦ Constrain compliance screening. For details, see Constraining Compliance Screening: Procedure.
   
   ◦ Require sales order approval. For details, see Managing Approval Rules for Sales Orders: How It Works.

**Related Topics**

- Using Integration Cloud Service to Integrate Order Management with Other Systems: Overview
- Managing Order Management Parameters: Procedure

**Managing Trade Compliance During Order Fulfillment: Procedure**

You can configure an orchestration process so that it screens order lines for trade compliance during order fulfillment, according to your business requirements.
In this example, you create a duplicate of a predefined orchestration process, and then modify this duplicate so that it includes the following steps.

This topic includes example values. You might use different values, depending on your business requirements.

Manage trade compliance in an orchestration process:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Orchestration Process Definitions.
   For details about how to configure an orchestration process, see the Setting Up Orchestration Processes chapter in the guide titled Oracle SCM Cloud, Implementing Order Management on Oracle Help Center.
4. On the Manage Orchestration Process Definitions page, search for the following orchestration process.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name</td>
<td>ShipOrderGenericProcess</td>
</tr>
</tbody>
</table>

5. In the Search Results, click **Actions**, and then click **Duplicate**.
6. On the Edit Orchestration Process Definitions page, set the following values, accept all other default values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name</td>
<td>Orchestration_Process_for_Compliance_Screening</td>
</tr>
</tbody>
</table>

   You can enter any value.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Display Name</td>
<td>Orchestration Process for Compliance Screening</td>
</tr>
<tr>
<td>Description</td>
<td>Screen sales order for trade compliance, and then process it through order fulfillment.</td>
</tr>
</tbody>
</table>

7. In the Process Details area, create the following step.

This step calls the service that performs compliance screening. You can place it at any location in the sequence of steps. For this example, assume you prefer to perform compliance screening after scheduling, so place it immediately after the Schedule step. The values for the Task attribute and the Service attribute are required to call the compliance screening service. Other attribute values are optional.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Request Trade Compliance Screening</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task Type</td>
<td>DOO_TradeCompliance</td>
</tr>
<tr>
<td>Task</td>
<td>DOO_TradeCompliance</td>
</tr>
<tr>
<td>Service</td>
<td>Request Screening for Trade Compliance</td>
</tr>
<tr>
<td>Exit Criteria</td>
<td>Leave empty</td>
</tr>
<tr>
<td>Default Lead Time</td>
<td>1</td>
</tr>
<tr>
<td>Lead-Time UOM</td>
<td>Hour</td>
</tr>
<tr>
<td>Update Service</td>
<td>Update Screening Request for Trade Compliance</td>
</tr>
<tr>
<td>Cancel Service</td>
<td>Cancel Screening Request for Trade Compliance</td>
</tr>
<tr>
<td>Use Transactional Item Attributes</td>
<td>Contains a check mark.</td>
</tr>
<tr>
<td>Use Flexfield Attributes</td>
<td>Contains a check mark.</td>
</tr>
<tr>
<td>Compensation Pattern</td>
<td>Do not define a rule.</td>
</tr>
</tbody>
</table>
8. In the Process Details area, create the following step immediately after the Request Screening for Trade Compliance step. This step pauses the orchestration process so that it waits for the compliance screening service to finish screening. The values for the Task, Service, and Exit Criteria attributes are required. Other attribute values are optional. Note that compliance screening is a long-running task, so you can perform compliance screening only through a long-running task.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Wait for Trade Compliance Screening</td>
</tr>
<tr>
<td></td>
<td>You can use any text for the name. It is recommended that you use a name that describes what the step accomplishes.</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task Type</td>
<td>DOO_TradeCompliance</td>
</tr>
<tr>
<td>Task</td>
<td>DOO_TradeCompliance</td>
</tr>
<tr>
<td>Service</td>
<td>Wait for Trade Compliance Screening</td>
</tr>
<tr>
<td>Exit Criteria</td>
<td>Passed DOO_PASSED Y</td>
</tr>
<tr>
<td></td>
<td>You can configure the exit criteria to continue processing or stop processing, according to your business requirements.</td>
</tr>
<tr>
<td></td>
<td>For example, to require the Order Entry Specialist to modify an order line that does not pass trade compliance screening, use Passed DOO_PASSED Y to allow only Passed status as the exit criteria. If the compliance screening fails or is under review, then the Order Entry Specialist can examine the message details that Order Management displays, and then decide on an action to take, such as revise the sales order.</td>
</tr>
<tr>
<td></td>
<td>For another example, to continue processing order lines while the ship-to-customer is under compliance review, include the Under Review status as part of the exit criteria. Order Management will process each order line through order fulfillment while trade management reviews the ship-to-customer for trade compliance.</td>
</tr>
<tr>
<td>Default Lead Time</td>
<td>1</td>
</tr>
<tr>
<td>Lead-Time UOM</td>
<td>Days</td>
</tr>
</tbody>
</table>

Set the Default Lead Time and Lead-Time UOM according to your business requirements. This example uses a 1 day lead time to allow Global Trade Management to process screening that might depend on an action in the fulfillment system that requires a long time to resolve.

9. In the Process Details area, click **Status Conditions**, click **Orchestration Process Status Values**, and then add the following status values.

<table>
<thead>
<tr>
<th>Status Value</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trade Compliance Screening Is Pending</td>
<td>&quot;DOO_TradeCompliance&quot; = &quot;DOO_AWAITING_COMPLIANCE_RSLT&quot;</td>
</tr>
<tr>
<td>Trade Compliance Screening Passed</td>
<td>&quot;DOO_TradeCompliance&quot; = &quot;DOO_PASSED&quot;</td>
</tr>
</tbody>
</table>
### Status Value | Expression
---|---
Trade Compliance Screening Failed | "DOO_ TradeCompliance" = "DOO_FAILED"
Trade Compliance Under Review | "DOO_ TradeCompliance" = "DOO_ UNDER_REVIEW"

Note that the exit criteria of the Wait for Trade Compliance Screening step is DOO_PASSED in this example.

10. Optional. To pause fulfillment while waiting for compliance review, or while performing more review and approval, add a pause step after the Wait step.

### Constraining Compliance Screening: Procedure

Order Management Cloud comes predefined to use the DOO_GTM_EXCEPTION processing constraint to reject each sales order that does not pass trade compliance screening when the Order Entry Specialist clicks Submit. This constraint prevents Order Management from sending a sales order that is not trade compliant to order fulfillment. If DOO_GTM_EXCEPTION does not meet your business requirements, then you can disable it. You can also create a new constraint. For example, you can create a constraint that rejects a sales order with status Failed but that does not reject a sales order with status Under Review.

In this example, you create a processing constraint that rejects each sales order that does not pass trade compliance screening.

This topic includes example values. You might use different values, depending on your business requirements.

**Summary of the Work**

To constrain compliance screening, do the following work:

1. Prepare your environment.
2. Disable DOO_GTM_EXCEPTION.
3. Create the validation rule set.
4. Create the processing constraint.

**Prepare Your Environment**

Prepare your environment:

1. If you have not enabled trade compliance screening to occur during order submit and during order fulfillment, then disable any constraint that references the trade compliance status.
2. If you have enabled trade compliance screening to occur only during order fulfillment, then disable any constraint that references the trade compliance status.

**Disable DOO_GTM_EXCEPTION**

Disable DOO_GTM_EXCEPTION:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Processing Constraints.
4. On the Manage Processing Constraints page, locate the predefined constraint named DOO_GTM_EXCEPTION.
5. Make sure the Enabled option does not contain a check mark.
The sales order will pass the submit validation and Order Management will send the sales order to order fulfillment even if it does not pass compliance screening.

Create the Validation Rule Set

Create the validation rule set:

1. On the Manage Processing Constraints page, click **Validation Rule Sets**.
2. Click **Actions**, click **Add Row**, and then enter the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Order with Trade Compliance Exception</td>
</tr>
<tr>
<td>Description</td>
<td>Defines a new constraint that prevents a sales order that does not pass trade compliance screening from moving to order fulfillment.</td>
</tr>
<tr>
<td>Short Name</td>
<td>TML</td>
</tr>
<tr>
<td></td>
<td>TML is an abbreviation for trade management line. You can use any value.</td>
</tr>
<tr>
<td>Validation Type</td>
<td>Table</td>
</tr>
<tr>
<td>Entity</td>
<td>Order Header</td>
</tr>
</tbody>
</table>

3. In the Details area, click **Actions**, click **Add Row**, and then set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Name</td>
<td>Trade Compliance Status</td>
</tr>
<tr>
<td>Validation Operation</td>
<td>Equal to</td>
</tr>
<tr>
<td>Value String</td>
<td>ORA_FAILED</td>
</tr>
</tbody>
</table>

4. Click **Save**, and then click **Generate Packages**.
5. In the Confirmation dialog, click **OK**.

Create the Processing Constraint

Create the processing constraint:

1. Click **Constraints**.
2. Click **Actions**, click **Add Row**, and then set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraint Name</td>
<td>GTM_EXCEPTION.ORDER</td>
</tr>
</tbody>
</table>

   If you create a new constraint for trade compliance screening, then you must include the text GTM as part of the constraint name.
### Managing Trade Control Types: Procedure

You can define the trade control types that Order Management Cloud references when it verifies a sales order. Antiterrorism is an example of a trade control type.

In this example, assume you must add the Chemical and Biological Weapons trade control type.

This topic includes example values. You might use different values, depending on your business requirements.

**Summary of the Work**
Manage trade control types:

1. Examine the lookup you must modify.
2. Manage predefined lookups.

Examine the Lookup You Must Modify
Order Management uses a predefined lookup to allow you to set the trade control type. In this section, you examine a predefined lookup.

Examine the lookup you must modify:

1. In the Navigator, click **Order Management**.
2. On the Overview page, click **Tasks**, and then click **Manage Fulfillment Line Exceptions**.
3. Click a row that displays a yellow or red Trade Compliance Status.
4. In the Fulfillment Line details area, click **Trade Compliance**, and then click **View Details** next to a red or yellow trade compliance status.
   The Manage Fulfillment Line Exceptions page only displays the View Details link next to a red or yellow trade compliance status.
5. Examine the predefined lookup that displays the trade control type.
   For example, the following diagram indicates the type of trade control that a fulfillment line might reference, such as Anti-Terrorism, and details about the control, such as Atomic Energy Act.

![Sanctioned Country or Territory: Line 2 – Failed](image)

Manage Predefined Lookups
Manage the predefined lookup that lists the trade control type:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Standard Lookups.
   For details about lookups, see **Lookups: Explained**.
4. On the Manage Standard Lookups page, in the Search area, enter the following value, and then click **Search**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Type</td>
<td>ORA_DOO_TRADE</td>
</tr>
</tbody>
</table>
5. In the Search Results area, click the row that contains the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Type</td>
<td>ORA_DOO_TRADE_CONTROL_CODE</td>
</tr>
</tbody>
</table>

6. In the Lookup Codes section, click **Actions**, click **New**, and then enter the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Code</td>
<td>CHEMICALS_AND_WEAPONS</td>
</tr>
<tr>
<td>Display Sequence</td>
<td>1</td>
</tr>
<tr>
<td>Meaning</td>
<td>Chemical and biological weapons</td>
</tr>
</tbody>
</table>

7. Click **Save**.

**Related Topics**
- Creating Sales Orders That Require Trade Compliance Screening: Procedure
- Lookups: Explained

### Managing Coverages and Subscriptions

#### Setting Up Coverage for Sales Orders: Procedure

You can set up and manage coverage items in Order Management Cloud. For example, you can set up a coverage, such as a warranty for laptop support that includes a daily, weekly, monthly, or yearly duration.

This topic includes example values. You might use different values, depending on your business requirements.

**Summary of the Work**

Manage coverage for sales orders:

1. Administer features and parameters.
2. Create the covered item.
3. Create the coverage item.
4. Set up pricing for the coverage item.
5. Test your work.
6. Add fixed and open-ended coverage.

**Administer Features and Parameters**

Administer the feature and parameter that affect coverage and subscription:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**, click **Actions**, and then click **Change Feature Selection**.
3. On the Edit Features page, in the Enable Coverages and Subscriptions row, add a check mark to the **Enable option**, and then click **Done**.

Order Management will apply the following behavior:

- Perform calculations for coverage items and subscription items.
- Allow the Order Entry Specialist to select a covered item when they add a coverage item to a sales order.
- Display the Service Details tab on the Fulfillment Line page.
- Display sales order attributes that define a coverage or subscription throughout the Order Management work area, such as Duration, Period, Contract Start Date, Contract End Date, Sales Product Type, and Associated Order Line.
- If the Order Entry Specialist chooses a coverage item or subscription, then Order Management disables actions that do not apply to a coverage item or subscription, such as Edit, Schedule, Check Availability, Unschedule, Split, Reserve, Unreserve, and Substitute.
- Display coverage items for each configure option in a configured item.

4. Click **Done**.

5. On the Setup page, search for, and then open Manage Order Management Parameters.

6. On the Manage Order Management Parameters page, set the following value.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage Start Date</td>
<td>Specify whether to set the start date for a coverage item to the delivery date or the shipment date.</td>
</tr>
</tbody>
</table>

Create the Covered Item

You will create and set up an item so that a coverage item can cover it.

Create the covered item:

1. In the Navigator, click **Product Management**, and then click **Product Information Management**.
2. On the Product Information Management page, click **Tasks**, and then click **Create Item**.
3. In the Create Item dialog, set the following values, and then click **OK**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>Vision Operations</td>
</tr>
<tr>
<td>Number of Items</td>
<td>1</td>
</tr>
<tr>
<td>Item Class</td>
<td>Root Item Class</td>
</tr>
<tr>
<td>Template</td>
<td>You can choose any value. For this example, choose Finished Goods.</td>
</tr>
</tbody>
</table>

4. On the Create Item page, enter the following values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>PTO54222 - Sentinel Personal Desktop Computer</td>
</tr>
</tbody>
</table>
Attribute | Value
---|---
Description | Sentinel Desktop computer.
Primary Unit of Measure | Each

5. Click **Specifications**, and then, under Item Organization, click **Sales and Order Management**.

6. Set the following attribute.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales Product Type</td>
<td>Choose one of the following values:</td>
</tr>
<tr>
<td></td>
<td>◦ <strong>Goods</strong>. A covered item that is tangible.</td>
</tr>
<tr>
<td></td>
<td>◦ <strong>Subscription</strong>. An item that provides a product or service that recurs, and that includes a duration and period. For example, a one year subscription to a magazine, a 90 day subscription for cell phone service, and a six month subscription for software usage are each an example of a subscription.</td>
</tr>
<tr>
<td></td>
<td>◦ <strong>Empty</strong>.</td>
</tr>
<tr>
<td></td>
<td>For this example, choose Goods.</td>
</tr>
</tbody>
</table>

7. Under Item Organization, click **Service**.
8. Set the following attribute, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Contract Coverage</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>You must enable contract coverage so that a coverage item can cover this covered item.</td>
</tr>
</tbody>
</table>

**Create the Coverage Item**

You will create a coverage item that covers the item PTO54222 - Sentinel Personal Desktop Computer. You configure the coverage item as an extended warranty, to display default values for duration and period, and to allow the Order Entry Specialist to modify these values.

Create the coverage item:

1. Click **Tasks**, and then click **Create Item**.
2. In the Create Item dialog, set the following values, and then click **OK**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>Vision Operations</td>
</tr>
<tr>
<td>Number of Items</td>
<td>1</td>
</tr>
<tr>
<td>Item Class</td>
<td>Warranty/ Services</td>
</tr>
</tbody>
</table>
You can choose any value. For this example, choose Finished Goods.

3. On the Create Item page, enter the following values, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template</td>
<td>You can choose any value. For this example, choose Finished Goods.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Variable Extended Warranty for Sentinel Desktop</td>
</tr>
<tr>
<td>Description</td>
<td>Variable warranty for the Sentinel Desktop computer.</td>
</tr>
<tr>
<td>Primary Unit of Measure</td>
<td>Year</td>
</tr>
</tbody>
</table>

4. Click **Specifications**, and then, under **Item Organization**, click **Sales and Order Management**.

5. Set the following attribute.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
</table>
| Sales Product Type | Set to one of the following values. You can use only these values for a coverage item:  
|                 | ◦ Extended Warranty  
|                 | ◦ Service Level Agreement  
|                 | ◦ Software Maintenance  
|                 | ◦ Preventive Maintenance  

For this example, choose **Extended Warranty**.

6. Under Item Organization, click **Service**.

7. Set the Service Duration Type attribute to one of the following values. For this example, set it to Variable.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed</td>
<td>The Order Management work area will default the values that it displays for Duration and Period to the values that you set, and then make them read-only.</td>
</tr>
<tr>
<td>Variable</td>
<td>Works the same as Fixed, except the Order Entry Specialist can edit Duration and Period.</td>
</tr>
<tr>
<td>Open Ended</td>
<td>Allows the Order Entry Specialist to edit Duration and Period, but does not display a default value for these attributes. The Order Entry Specialist must set them.</td>
</tr>
</tbody>
</table>

Note that you cannot set the Enable Contract Coverage attribute because this attribute allows a coverage item to cover the item you are defining. However, a coverage item cannot cover another coverage item. For example, you cannot create a 1 Year Sentinel Desktop Warranty that covers a 5 Year Sentinel Desktop Warranty.

8. Set the following values for the duration, click **Save and Close**, and then sign out.
Oracle SCM Cloud
Implementing Order Management
Chapter 6
Setting Up Features

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>5</td>
</tr>
<tr>
<td>Duration Period</td>
<td>Year</td>
</tr>
</tbody>
</table>

Note: It is strongly recommended that you set these values. Order Management displays them in fields that are not labeled on the catalog line. If you do not set these values, then the fields will be empty, and their usage will not be clear.

Set Up Pricing for the Coverage Item
For this example, assume you must allow the Order Entry Specialist to set coverage to recur yearly or monthly, and to use the following calculation for the coverage:

- For yearly coverage, charge 20% of the covered item price.
- For monthly coverage, charge 10% of the covered item price.

You define a charge for each of these durations. For details about setting up Pricing, see the Managing Pricing for Covered Items section in the Oracle SCM Cloud, Administering Pricing guide.

For brevity, assume you already set up pricing for the covered item named PTO54222 - Sentinel Personal Desktop Computer so that it calculates a pricing basis of $500.

Set up pricing for the coverage item:
1. Log into Oracle Fusion Pricing.
2. In the Navigator, click Pricing Administration.
3. On the Overview page, in the Updated Pricing Rules area, click Corporate Segment Price List.
4. In the Search Results, click Action, click Add Row, set the following values, and then click Save.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Variable Extended Warranty for Sentinel Desktop</td>
</tr>
<tr>
<td>Line Type</td>
<td>Buy</td>
</tr>
</tbody>
</table>

If you:
- Do not enter a value for Service Duration, then Pricing will not calculate a value for the Duration Extended Amount attribute.
- Enter a value for Service Duration, then Pricing will calculate a value for the Duration Extended Amount attribute.

If the Service Duration Type is Open Ended for the coverage item in the Product Information Management work area:
- The Search Results area will not display a value for Service Duration Period or Service Duration, and will not allow you to edit these attributes.
And if Pricing prices the sales order instead of your source system pricing it, then the Order Entry Specialist must enter a value for Service Duration Period.

5. In the Associated Items column, click **Manage Covered Item**.
6. On the Manage Covered Items page, in the Search Results area, click **Add Row**, set the following values, and then click **Create Charge**.

<table>
<thead>
<tr>
<th>Pricing UOM</th>
<th>Coverage UOM</th>
<th>Action Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each</td>
<td>Year</td>
<td>Add</td>
</tr>
</tbody>
</table>

7. In the Charge area, set the following values, and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing Charge Definition</td>
<td>OAL Recurring Charge</td>
</tr>
<tr>
<td>Calculation Method</td>
<td>Covered Item Price Percent</td>
</tr>
<tr>
<td>Coverage Basis</td>
<td>Coverage Basis Your Price</td>
</tr>
<tr>
<td>Calculation Amount</td>
<td>20</td>
</tr>
<tr>
<td>Price Periodicity</td>
<td>Year</td>
</tr>
</tbody>
</table>

8. In the Search Results area, click **Add Row**, set the following values, and then click **Create Charge**.

<table>
<thead>
<tr>
<th>Pricing UOM</th>
<th>Coverage UOM</th>
<th>Action Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each</td>
<td>Month</td>
<td>Add</td>
</tr>
</tbody>
</table>

9. In the Charge area, set the following values, and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing Charge Definition</td>
<td>OAL Recurring Charge</td>
</tr>
<tr>
<td>Calculation Method</td>
<td>Covered Item Price Percent</td>
</tr>
<tr>
<td>Coverage Basis</td>
<td>Coverage Basis Your Price</td>
</tr>
<tr>
<td>Calculation Amount</td>
<td>10</td>
</tr>
<tr>
<td>Price Periodicity</td>
<td>Month</td>
</tr>
</tbody>
</table>

10. Click **Save and Close** again, and then sign out of Pricing.

**Specifying Charges for Each Duration**
Note that Duration comes predefined to display the following values:

- Year
- Quarter
- Month
- Week
- Day
- Hours
- Minutes

Note the following points:

- Order Management renders pricing details only for each charge that you specify. For example, if you do not specify a charge for Month, and if the Order Entry Specialist sets Duration to Month, then the order line for the coverage item will not display price details, Order Management will display an error message, and the user cannot submit the sales order.
- If you define a coverage item in Product Information Management, but do not set up pricing for it in the Pricing Administration work area, then the Order Entry Specialist might add the coverage item but not be able to submit the sales order because of a pricing error.

⚠️ Caution: If you define a coverage item in Product Information Management, but incorrectly set up pricing for it in the Pricing Administration work area, then the Order Entry Specialist might add the coverage item and successfully submit it without error, but the pricing values on the order line might not be correct, resulting in overcharging or undercharging the billing.

Test Your Work

Test your work:

1. Log into Order Management, and then create a sales order.
2. On the catalog line, search for the following item, and then click Add.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Item</td>
<td>PTO54222</td>
</tr>
</tbody>
</table>

3. On the catalog line, search for the following item.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Item</td>
<td>Variable Extended Warranty for Sentinel Desktop</td>
</tr>
</tbody>
</table>

4. Wait for the search to finish, and then verify that the catalog line displays the following values, which are the default values you set up in Product Information Management.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>5</td>
</tr>
<tr>
<td>Period</td>
<td>Year</td>
</tr>
</tbody>
</table>
5. Click **Select Covered Item**, choose the order line to cover, and then click **Add**.
6. Click **View**, click **Columns**, and then click **Period**.
7. Verify that the following values display on the coverage line, by default.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td></td>
</tr>
<tr>
<td>Your Price</td>
<td>100</td>
</tr>
<tr>
<td>Amount</td>
<td>100</td>
</tr>
<tr>
<td>Amount for Total Duration</td>
<td></td>
</tr>
</tbody>
</table>

8. Change the quantity on the covered line from 1 to 2, and then verify that the coverage line displays the following values.

<table>
<thead>
<tr>
<th>Quantity</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Your Price</td>
<td>100</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>Amount</td>
<td>100</td>
<td>100</td>
<td>1000</td>
</tr>
<tr>
<td>Amount for Total Duration</td>
<td></td>
<td>1000</td>
<td>5</td>
</tr>
</tbody>
</table>

9. Change the quantity on the covered line from 2 to 1.
10. Change the Duration to 1, and then verify that the coverage line displays the following values.

<table>
<thead>
<tr>
<th>Quantity</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Your Price</td>
<td>100</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Amount</td>
<td>100</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Amount for Total Duration</td>
<td></td>
<td>50</td>
<td>1</td>
</tr>
</tbody>
</table>

11. Change the Period to Month, and then verify that the coverage line displays the following values.

<table>
<thead>
<tr>
<th>Quantity</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Your Price</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Amount</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Amount for Total Duration</td>
<td></td>
<td>50</td>
<td>1</td>
</tr>
</tbody>
</table>

12. Change the Period to Day, verify that the coverage line does not display price details, click **Actions**, click **Reprice Order**, and then notice the error message.

Order Management will display an error for any charge that you do not define. Recall that you did not create a charge for Day.

### Add Fixed and Open-Ended Coverage

The behavior of fixed coverage and open-ended coverage is slightly different from the behavior of variable coverage. As an option, you will create a fixed coverage item, an open-ended coverage item, and then test them.

Add fixed coverage and open-ended coverage:

1. Use the Product Information Management work area to create the following coverage item.
2. Create another coverage item using the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>Vision Operations</td>
</tr>
<tr>
<td>Number of Items</td>
<td>1</td>
</tr>
<tr>
<td>Item Class</td>
<td>Warranty/Services</td>
</tr>
<tr>
<td>Template</td>
<td>Finished Goods</td>
</tr>
<tr>
<td>Item</td>
<td>Open-Ended Extended Warranty for Sentinel Desktop</td>
</tr>
<tr>
<td>Description</td>
<td>Open-ended warranty for the Sentinel Desktop computer.</td>
</tr>
<tr>
<td>Primary Unit of Measure</td>
<td>Year</td>
</tr>
<tr>
<td>Sales Product Type</td>
<td>Extended Warranty</td>
</tr>
<tr>
<td>Service Duration Type</td>
<td>Fixed</td>
</tr>
<tr>
<td>Duration</td>
<td>5</td>
</tr>
<tr>
<td>Duration Period</td>
<td>Year</td>
</tr>
</tbody>
</table>
3. Sign out, and then log into Oracle Fusion Pricing.
4. Define pricing for the fixed coverage item on the Corporate Segment Price List using the following values. Note that Order Management does not allow the Order Entry Specialist to change the Duration or Period for a fixed coverage item, so you only need to define a charge for one period.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Fixed Extended Warranty for Sentinel Desktop</td>
</tr>
<tr>
<td>Line Type</td>
<td>Buy</td>
</tr>
<tr>
<td>Pricing UOM on the Manage Covered Items page</td>
<td>Each</td>
</tr>
<tr>
<td>Coverage UOM</td>
<td>Year</td>
</tr>
<tr>
<td>Action Type</td>
<td>Add</td>
</tr>
<tr>
<td>Pricing Charge Definition</td>
<td>OAL Recurring Charge</td>
</tr>
<tr>
<td>Calculation Method</td>
<td>Covered Item Price Percent</td>
</tr>
<tr>
<td>Coverage Basis</td>
<td>Coverage Basis Your Price</td>
</tr>
<tr>
<td>Calculation Amount</td>
<td>20</td>
</tr>
<tr>
<td>Price Periodicity</td>
<td>Year</td>
</tr>
</tbody>
</table>

5. Add the open-ended coverage item to the Corporate Segment Price List using the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Open-Ended Extended Warranty for Sentinel Desktop</td>
</tr>
<tr>
<td>Line Type</td>
<td>Buy</td>
</tr>
</tbody>
</table>

6. Add the charge for Year. Use the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing UOM</td>
<td>Each</td>
</tr>
<tr>
<td>Coverage UOM</td>
<td>Year</td>
</tr>
<tr>
<td>Action Type</td>
<td>Add</td>
</tr>
<tr>
<td>Pricing Charge Definition</td>
<td>OAL Recurring Charge</td>
</tr>
<tr>
<td>Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Calculation Method</td>
<td>Covered Item Price Percent</td>
</tr>
<tr>
<td>Coverage Basis</td>
<td>Coverage Basis Your Price</td>
</tr>
<tr>
<td>Calculation Amount</td>
<td>20</td>
</tr>
<tr>
<td>Price Periodicity</td>
<td>Year</td>
</tr>
</tbody>
</table>

7. Add the charge for Quarter. Use the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing UOM</td>
<td>Each</td>
</tr>
<tr>
<td>Coverage UOM</td>
<td>Quarter</td>
</tr>
<tr>
<td>Action Type</td>
<td>Add</td>
</tr>
<tr>
<td>Pricing Charge Definition</td>
<td>OAL Recurring Charge</td>
</tr>
<tr>
<td>Calculation Method</td>
<td>Covered Item Price Percent</td>
</tr>
<tr>
<td>Coverage Basis</td>
<td>Coverage Basis Your Price</td>
</tr>
<tr>
<td>Calculation Amount</td>
<td>5</td>
</tr>
<tr>
<td>Price Periodicity</td>
<td>Quarter</td>
</tr>
</tbody>
</table>

8. Add the charge for Month. Use the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing UOM</td>
<td>Each</td>
</tr>
<tr>
<td>Coverage UOM</td>
<td>Month</td>
</tr>
<tr>
<td>Action Type</td>
<td>Add</td>
</tr>
<tr>
<td>Pricing Charge Definition</td>
<td>OAL Recurring Charge</td>
</tr>
<tr>
<td>Calculation Method</td>
<td>Covered Item Price Percent</td>
</tr>
<tr>
<td>Coverage Basis</td>
<td>Coverage Basis Your Price</td>
</tr>
</tbody>
</table>
Oracle SCM Cloud
Implementing Order Management

Chapter 6
Setting Up Features

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculation Amount</td>
<td>10</td>
</tr>
<tr>
<td>Price Periodicity</td>
<td>Month</td>
</tr>
</tbody>
</table>

9. Add the charge for Week. Use the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing UOM</td>
<td>Each</td>
</tr>
<tr>
<td>Coverage UOM</td>
<td>Week</td>
</tr>
<tr>
<td>Action Type</td>
<td>Add</td>
</tr>
<tr>
<td>Pricing Charge Definition</td>
<td>OAL Recurring Charge</td>
</tr>
<tr>
<td>Calculation Method</td>
<td>Covered Item Price Percent</td>
</tr>
<tr>
<td>Coverage Basis</td>
<td>Coverage Basis Your Price</td>
</tr>
<tr>
<td>Calculation Amount</td>
<td>7</td>
</tr>
<tr>
<td>Price Periodicity</td>
<td>Week</td>
</tr>
</tbody>
</table>

10. Save your work, and then sign out of Pricing.

11. Log into Order Management, and then create a sales order.

12. On the catalog line, search for the following item, and then click Add.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Item</td>
<td>PTO54222</td>
</tr>
</tbody>
</table>

13. On the catalog line, search for the following item.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Item</td>
<td>Fixed Extended Warranty for Sentinel Desktop</td>
</tr>
</tbody>
</table>

14. Wait for the search to finish, and then notice that the catalog line displays 5 Year as a read-only value.

15. Click Select Covered Item, choose the order line to cover, and then click Add.

16. Verify that you cannot edit the Duration or Period.

17. On the catalog line, search for the following item.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Item</td>
<td>Open-Ended Extended Warranty for Sentinel Desktop</td>
</tr>
</tbody>
</table>
18. Wait for the search to finish, and then notice that the catalog line displays two empty fields, and that you must set these fields before you continue. If you do not set them, then Order Management displays an error. Note that an open-ended coverage item is similar to a variable coverage item, except that Order Management displays the default values for Duration and Period on the catalog line for a variable coverage item, but does not display any values for an open-ended coverage item. The Order Entry Specialist must set these values for an open-ended coverage item.

19. Set the following values on the catalog line, click Select Covered Item, choose the order line to cover, and then click Add.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>1</td>
</tr>
<tr>
<td>Period</td>
<td>Day</td>
</tr>
</tbody>
</table>

20. Verify that the coverage order line for the open-ended coverage does not contain prices. You did not define a charge for Day, so no pricing will display, and you cannot submit the sales order.

21. Set Period to Year, and then verify that the following values display on the coverage line.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Your Price</th>
<th>Amount</th>
<th>Amount for Total Duration</th>
<th>Duration</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>Year</td>
</tr>
</tbody>
</table>

22. Set Period to Month, and then verify that the following values display on the coverage line.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Your Price</th>
<th>Amount</th>
<th>Amount for Total Duration</th>
<th>Duration</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>1</td>
<td>Year</td>
</tr>
</tbody>
</table>

23. Set Period to Week, and then verify that the following values display on the coverage line.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Your Price</th>
<th>Amount</th>
<th>Amount for Total Duration</th>
<th>Duration</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>1</td>
<td>Year</td>
</tr>
</tbody>
</table>

24. Set Duration to 2, and then verify that the following values display on the coverage line.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Your Price</th>
<th>Amount</th>
<th>Amount for Total Duration</th>
<th>Duration</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>35</td>
<td>35</td>
<td>70</td>
<td>1</td>
<td>Year</td>
</tr>
</tbody>
</table>
Related Topics

- Adding Coverage to Sales Orders: Explained

Setting Up Orchestration Processes for Coverage Items: Explained

If you set up an orchestration process that processes a covered item or a coverage item, then you can use the same orchestration process instance for the covered item and for the coverage item, depending on your business requirements. You can coordinate different instances with each other.

Using the Same Orchestration Process Instance

The following diagram illustrates the recommended way to configure an orchestration process when you use the same orchestration process instance to process the covered item and the coverage item.

If you use the same orchestration process instance, then it is recommended that you define an Assign and Launch rule and specify an expression in the line selection criteria. This orchestration process is only for illustration purposes. You might require different assignment rules and expressions that meet your business requirements.

For details about how to assign an orchestration process, see Assigning Orchestration Processes: Explained.

Define an Assign and Launch Rule

You can define an Assign and Launch rule that makes sure Order Management uses the same orchestration process instance for the covered item and for the coverage item. Using the same orchestration process instance achieves the following results:

- Makes sure the orchestration process uses the same steps, in the same sequence, and at the same time for the covered item and for the coverage item.
• Helps to make sure processing for the covered item and coverage item remain synchronized, and makes sure processing for the covered item does not get too far ahead or behind processing for the coverage item.

If you define an Assign and Launch rule that assigns the same orchestration process to the covered item and to the coverage item, then a single orchestration process instance processes these items. You define this rule one time during setup. The orchestration process instance then runs in the runtime environment and it is different for each sales order or fulfillment line.

**Specify an Expression in the Line Selection Criteria**

You can specify an expression in the Line Selection Criteria for each orchestration process step that must not run so that the step does not call the fulfillment task service. The grey steps in the diagram above must not run. In this example, it is not necessary to ship a coverage item, such as a warranty, so you must specify an expression that prevents the shipping step from processing the coverage item. It is also not necessary to run the contracts step for the covered item, so you must specify an expression that prevents it from processing the covered item.

Consider the following examples:

• You can add a line selection criteria that specifies to process only shippable lines. You add this criteria to a Scheduling task or Shipping task. Note that the predefined orchestration processes already use this line selection criteria.

• A Contracts step does not need to run for a covered item, so you can add a line selection criteria that specifies to run this step only for each line where the Sales Product Type specifies a coverage item.

For details about how to specify expressions, see Selecting Fulfillment Lines for Orchestration Process Steps: Procedure.

**Using Different Orchestration Process Instances**

It might be necessary to use one orchestration process instance to process the covered item and another orchestration process instance to process the coverage item. For example:

• Your business requirements demand that you process them separately.

• An Assign and Launch rule assigns separate process instances when creating the sales order.

• The Order Entry Specialist uses Revise Order to add coverage to an existing covered item, and then Order Management assigns a separate orchestration process instance even if the Assign and Launch rule assigns the same instance that processes the covered item.
The following diagram illustrates an example of how you can use a pause task to coordinate different orchestration process instances.

**Covered Item, Laptop Computer**

Schedule → Reserve → Ship → Install Base → Invoice

Pause exit criteria

**Coverage Item, Warranty for Laptop Computer**

Pause → Contracts → Invoice

Note the following points:

- The pause task makes sure the orchestration process instance that processes the coverage item does not get too far out in front of the instance that processes the covered item. For example, the Contracts step must not process the coverage item until after the covered item ships and the Install Base step runs.
- You can set the pause exit criteria to a condition, such as Shipped Quantity is greater than zero.

This orchestration process is only for illustration purposes. You might require different pause tasks that meet your business requirements.

For details about how to use a pause task, see Pausing Orchestration Processes: Explained.

**Related Topics**

- Assigning Orchestration Processes: Explained
- Selecting Fulfillment Lines for Orchestration Process Steps: Procedure
- Pausing Orchestration Processes: Explained

**Setting Up Subscriptions for Sales Orders: Procedure**

Setting up a subscription for a sales order is similar to setting up coverage for a sales order.

Note the following points:

- You can include a duration and period.
- You can define a subscription as a configured item.
- You can enable a subscription for Contract coverage similar to how you enable a covered item so that it is available to associate with a service.
- You can also specify another recurring price for on-going, periodic billing that bills the subscription for the periods that occur after the first pay period.
- If you import a subscription, then the import must include the start date, end date, duration, duration UOM.

In the example in this topic, you set up an annual subscription for Visions Magazine at a rate of $5 for each week.

To set up subscriptions for sales orders, do the set up described in Setting Up Coverage for Sales Orders: Procedure, but with the following differences:

1. Use the Product Information Management work area to create the following subscription.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>Vision Operations</td>
</tr>
<tr>
<td>Number of Items</td>
<td>1</td>
</tr>
<tr>
<td>Item Class</td>
<td>Warranty/Services</td>
</tr>
<tr>
<td>Template</td>
<td>Finished Goods</td>
</tr>
<tr>
<td>Item</td>
<td>Subscription to Visions Magazine</td>
</tr>
<tr>
<td>Description</td>
<td>Subscription for Visions Magazine, a weekly publication for trade professionals.</td>
</tr>
<tr>
<td>Primary Unit of Measure</td>
<td>Week</td>
</tr>
<tr>
<td>Pricing typically must reference a primary UOM that it can measure numerically, such as number of weeks, number of users, or number of hosted environments. If you cannot use a predefined UOM, then you must define a new one.</td>
<td></td>
</tr>
<tr>
<td>Sales Product Type</td>
<td>Subscription</td>
</tr>
<tr>
<td>Service Duration Type</td>
<td>Variable</td>
</tr>
<tr>
<td>You can use Fixed, Variable, or Open-Ended. It is recommended that you use Variable in most situations because it allows you to suggest a default value, but also provides the Order Entry Specialist the flexibility to adjust or negotiate price for each customer.</td>
<td></td>
</tr>
<tr>
<td>If you use Open Ended, then renewal is not required and the subscription continues until the Order Entry Specialist ends it. For example, assume the customer buys a subscription to an electrical utility that must continue indefinitely until the customer actively ends it.</td>
<td></td>
</tr>
<tr>
<td>Duration</td>
<td>52</td>
</tr>
<tr>
<td>Duration Period</td>
<td>Week</td>
</tr>
</tbody>
</table>

2. Use the Pricing Administration work area to define pricing for the subscription on the Corporate Segment Price List. Use the following values.
### Attribute Table

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Subscription to Visions Magazine</td>
</tr>
<tr>
<td>Line Type</td>
<td>Buy</td>
</tr>
<tr>
<td>Pricing UOM</td>
<td>Week</td>
</tr>
<tr>
<td>Service Duration Period</td>
<td>Week</td>
</tr>
<tr>
<td>Service Duration</td>
<td>52</td>
</tr>
</tbody>
</table>

**3.** Add the charge for Week. Use the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing Charge Definition</td>
<td>OAL Recurring Charge</td>
</tr>
<tr>
<td>Calculation Method</td>
<td>Price</td>
</tr>
<tr>
<td>Base Price</td>
<td>5</td>
</tr>
<tr>
<td>Calculation Amount</td>
<td>20</td>
</tr>
<tr>
<td>Price Periodicity</td>
<td>Week</td>
</tr>
</tbody>
</table>

**4.** Sign out of Pricing Administration, and then log into Order Management.

**5.** Add the item named Subscription to Visions Magazine to the sales order, and then verify that the order line displays the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>UOM</th>
<th>Your Price</th>
<th>Amount</th>
<th>Amount for Total Duration</th>
<th>Duration</th>
<th>Period</th>
<th>Billing Frequency</th>
<th>Number of Billing Periods</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Week</td>
<td>5</td>
<td>5</td>
<td>260</td>
<td>52</td>
<td>Week</td>
<td>One Time Billing</td>
<td>1</td>
</tr>
</tbody>
</table>

**Related Topics**

- Adding Coverage to Sales Orders: Explained

**Setting Up Subscriptions for Covered Items: Procedure**

You can set up a subscription for a covered item. In this example, you define a covered item for cloud up time, and then define a subscription that allows the Order Entry Specialist to add this up time.
Summary of the Work

Set up a subscription for a covered item:

1. Add the User UOM.
2. Define the agreement and the subscription.

Add the User UOM

In this example, you add a User UOM as a way to measure the type of usage that the subscription and the covered item provides.

Add the User UOM:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Product Management**.
3. Search for, and then open Unit of Measure Class.
4. On the Business Object, Unit of Measure Class page, click **Manage Units of Measure**.
5. On the Manage UOM Classes page, click **Manage UOM Classes**.
6. On the Manage UOM Classes page, click **Actions**, click **Add**, set the following values, and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Code</td>
<td>USE</td>
</tr>
<tr>
<td>Class Name</td>
<td>User</td>
</tr>
<tr>
<td>Description</td>
<td>People who use a subscription</td>
</tr>
<tr>
<td>UOM Code</td>
<td>USR</td>
</tr>
<tr>
<td>Base UOM Name</td>
<td>Users</td>
</tr>
<tr>
<td>BASE UOM Description</td>
<td>Someone who uses a subscription</td>
</tr>
</tbody>
</table>

Define the Agreement and the Subscription

Do the steps described in Setting Up Coverage for Sales Orders: Procedure, but with the following differences:

1. Use the Product Information Management work area to define the following agreement.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>Vision Operations</td>
</tr>
<tr>
<td>Number of Items</td>
<td>1</td>
</tr>
<tr>
<td>Item Class</td>
<td>Warranty/ Services</td>
</tr>
<tr>
<td>Template</td>
<td>Finished Goods</td>
</tr>
</tbody>
</table>
2. Define the subscription. Use the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>Vision Operations</td>
</tr>
<tr>
<td>Number of Items</td>
<td>1</td>
</tr>
<tr>
<td>Item Class</td>
<td>Warranty/ Services</td>
</tr>
<tr>
<td>Template</td>
<td>Finished Goods</td>
</tr>
<tr>
<td>Item</td>
<td>Vision Cloud Subscriptions</td>
</tr>
<tr>
<td>Description</td>
<td>Subscriptions for end-user access to the Vision Cloud.</td>
</tr>
<tr>
<td>Primary Unit of Measure</td>
<td>Users</td>
</tr>
<tr>
<td>Sales Product Type</td>
<td>Subscription</td>
</tr>
<tr>
<td>Service Duration Type</td>
<td>Fixed</td>
</tr>
<tr>
<td>Duration</td>
<td>3</td>
</tr>
<tr>
<td>Duration Period</td>
<td>Year</td>
</tr>
</tbody>
</table>

3. Use the Pricing Administration work area to define pricing for the agreement on the Corporate Segment Price List. Use the following values.
4. Add the charge. Use the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing Charge Definition</td>
<td>OAL Recurring Charge</td>
</tr>
<tr>
<td>Calculation Method</td>
<td>Price</td>
</tr>
<tr>
<td>Base Price</td>
<td>30000</td>
</tr>
<tr>
<td>Price Periodicity</td>
<td>Year</td>
</tr>
</tbody>
</table>

5. Use the Pricing Administration work area to define pricing for the subscription on the Corporate Segment Price List. Use the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Vision Cloud Subscriptions</td>
</tr>
<tr>
<td>Line Type</td>
<td>Buy</td>
</tr>
<tr>
<td>Pricing UOM</td>
<td>Users</td>
</tr>
<tr>
<td>Pricing UOM</td>
<td>Users</td>
</tr>
</tbody>
</table>

6. Add the charge. Use the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pricing Charge Definition</td>
<td>OAL Recurring Charge</td>
</tr>
<tr>
<td>Calculation Method</td>
<td>Price</td>
</tr>
</tbody>
</table>
7. Sign out of Pricing Administration, and then log into Order Management.
8. Create a sales order, add the item Vision Cloud Subscriptions to the sales order, and then verify that the order line displays the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Price</td>
<td>1000</td>
</tr>
<tr>
<td>Price Periodicity</td>
<td>Year</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quantity</th>
<th>UOM</th>
<th>Your Price</th>
<th>Amount</th>
<th>Amount for Total Duration</th>
<th>Duration</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Users</td>
<td>1,000</td>
<td>1,000</td>
<td>3,000</td>
<td>3</td>
<td>Year</td>
</tr>
</tbody>
</table>

9. Search for the item Vision Cloud Agreement on the catalog line, click **Select Covered Item**, associate it with the Vision Cloud Subscriptions order line, click **Add**, and then verify that the Vision Cloud Agreement order line displays the following values.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>UOM</th>
<th>Your Price</th>
<th>Amount</th>
<th>Amount for Total Duration</th>
<th>Duration</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Users</td>
<td>1,000</td>
<td>1,000</td>
<td>3,000</td>
<td>3</td>
<td>Year</td>
</tr>
</tbody>
</table>

**Related Topics**

- Adding Coverage to Sales Orders: Explained

**Importing Source Orders That Include Coverage or Subscriptions: Procedure**

You can import a source order that includes covered items, coverage items, or subscriptions into Order Management Cloud. For details about how to import source orders, see the Importing Source Orders section in the Oracle SCM Cloud, Implementing Order Management guide.

This topic describes how to import source orders that include coverage. However, you can also use it to import subscriptions.

Import source orders that include coverage into Order Management:

1. Choose one of the following technologies to import your source orders:
   - Web service in Oracle Application Development Framework (ADF)
   - Web service in Oracle Service-Oriented Architecture (SOA)
   - Business-to-business (B2B) messaging in Oracle Collaboration Messaging Framework
   - File Based Import. For details, see Order Import Template: Explained.

2. Make sure your source order includes the following attributes.
The import uses these attributes to establish a relationship between the coverage item and the covered item. The import process that you use must map each of these attributes in the import payload of the source order to an attribute in a sales order in Order Management.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Reference Type</td>
<td>Creates a relationship between the covered item and the coverage item.</td>
</tr>
<tr>
<td>Document Line Id</td>
<td>Line Id in the source transaction. This Id identifies the coverage item.</td>
</tr>
<tr>
<td>Document Sub Line Id</td>
<td>Line Id in the source transaction. This Id identifies the coverage item. To purchase an item and apply coverage for it:</td>
</tr>
<tr>
<td></td>
<td>◦ At the same time. This subline can reside in the same source order.</td>
</tr>
<tr>
<td></td>
<td>◦ At some later time. This subline can reside in a closed order line from another sales order that you already imported.</td>
</tr>
<tr>
<td>Document Id</td>
<td>Source order Id of the covered line.</td>
</tr>
<tr>
<td>Document Number</td>
<td>Sales order number of the covered line.</td>
</tr>
<tr>
<td>Document Line Additional Id</td>
<td>Line Id from the source transaction. This Id identifies the coverage for the root parent of a configured item.</td>
</tr>
</tbody>
</table>

3. Make sure your source order includes a value for the Number of Billing Periods attribute for the coverage item.
4. Make sure your source order includes values for the following coverage item attributes, depending on whether the source order contains price details.

<table>
<thead>
<tr>
<th>Contains Price Details</th>
<th>Does Not Contain Price Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Fusion Pricing does not price the item. Note the following points:</td>
<td>Oracle Fusion Pricing prices the item.</td>
</tr>
<tr>
<td>◦ The Duration Extended Amount must contain a value, and this value must use the same currency that the order header uses.</td>
<td>Each of the following attributes must contain a value:</td>
</tr>
<tr>
<td>◦ Contract Start Date must contain a value.</td>
<td>◦ Duration</td>
</tr>
<tr>
<td>◦ Values for Duration, Period, and Contract End Date can be empty.</td>
<td>◦ Period</td>
</tr>
<tr>
<td></td>
<td>◦ Contract Start Date</td>
</tr>
</tbody>
</table>

5. Verify the following settings in the Product Management work area:

- Make sure you define the Enable Contract Coverage attribute and the Sales Product Type attribute in Product Management so that they support the covered item. For details, see Managing Coverage for Sales Orders: Procedure.
- To specify a subscription item, set Sales Product Type to Subscription. To use a subscription as a covered item, enable Contract Coverage for the subscription. Note that a subscription is similar to a coverage item. It includes a Duration, Period, Contract Start Date, and Contract End Date. In general, the rules that apply for
importing a coverage item also apply for importing a subscription. For details, see Managing Subscriptions for Sales Orders: Procedure.

6. If you set Service Duration Type in Product Management to:
   - **Fixed**. Make sure the Duration and Period in the source order match the values you set for Duration and Period in Product Management.
   - **Open Ended**. And if you import a source order that includes a coverage item that does not include price details, then make sure the import includes values for Duration and Period.

7. Make sure the value of the coverage quantity equals the value of the covered quantity.

8. Make sure the value of the coverage UOM equals the value of the covered UOM.

9. If the coverage item covers a covered item that Order Management already submitted to order fulfillment, then make sure the status of the covered item is Closed.

10. Examine your import payload. Make sure the document references that identify the coverage line map to only a single covered line.

   One coverage line can cover only one covered line. A coverage line must not cover more than one covered line.

**Importing Coverage with Configured Items That Are Already Priced**

If the source order contains price details for a configured item, and if each of the following attributes in the source order are set to Y, then you must make sure the input payload includes coverage for each configure option that is enabled for contract coverage:

- Freeze Price
- Freeze Shipping Charges
- Freeze Tax

If Enable Contract Coverage equals Y on the Edit Item page of the Product Management work area, then the configure option is enabled for contract coverage.

Make sure your import performs the following validations for each configure option that is enabled for contract coverage:

- If the source order does not contain a coverage for each configure option, then reject the source order.
- If the coverage quantity and the UOM for each configure option do not match the line quantity and UOM for the covered item, then reject the source order.

For details about pricing, see Pricing Covered Items: Overview.

**Importing Coverage with Configured Items That Are Not Priced**

If the source order does not contain price details, and if any of the following attributes in the source order is set to N, then Oracle Fusion Pricing will price it:

- Freeze Price
- Freeze Shipping Charges
- Freeze Tax

Note the following points:

- The source order must include coverage only for the configured item that is the root parent, and not for each configure option.
- If you enable the configure option for contract coverage, then your import process must create a coverage line for each configure option. Each configure option can be an option class or option item.
If the input payload contains a coverage for the root parent and also for one or more configure options, then your import must reject the source order.

Importing Coverage That Includes Returns

You can import a covered item that includes a return. Note the following points:

- You can keep the covered item and return only the coverage item. Make sure your import payload includes only the return lines for the coverage item.
- If the Contract End Date of the coverage line occurs before the return date of the coverage line, then you cannot return a coverage line.

Make sure your return includes the following details:

- Status of the covered line and the coverage line are each Closed.
- Return quantity is greater than zero.
- Return quantity does not exceed the quantity that is available to return. For example, if the sales order included a quantity of 10, and if you already returned a quantity of six, then the return quantity cannot exceed four.
- A document reference type of Original_Sales_Order for the return coverage line.
- A document reference type of Source_Coverage_Covered_Association for the covered line.

Note the following points when returning a covered item, depending on whether the source order contains price details.

<table>
<thead>
<tr>
<th>Contains Price Details</th>
<th>Does Not Contain Price Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Fusion Pricing did not price the item.</td>
<td>Oracle Fusion Pricing priced the item.</td>
</tr>
<tr>
<td>You can return the coverage item that covers this covered item.</td>
<td>The import payload can specify to return the covered item and the coverage lines.</td>
</tr>
<tr>
<td>Make sure your import payload includes the following details:</td>
<td>Make sure your import payload includes the return line for the covered item. Oracle Fusion Applications will create the return lines for the coverage lines.</td>
</tr>
<tr>
<td>- The quantity and UOM of the coverage line equal the quantity and UOM of the covered line. For example, assume a sales order includes a quantity of 5 for the covered item and a quantity of 5 for the coverage item. If the return includes a quantity of 3 for the covered item, then make sure the coverage line in the return includes a quantity of 3.</td>
<td></td>
</tr>
<tr>
<td>- Adds the returned quantity to the quantity that is available.</td>
<td></td>
</tr>
<tr>
<td>- Includes return charges.</td>
<td></td>
</tr>
</tbody>
</table>

If your import payload returns only the coverage item, then note the following points:

- The quantity for the coverage line must equal the total returnable quantity of the covered line. Your import payload cannot return only part of the returnable quantity.
- The coverage return UOM in the import payload must equal the covered line UOM.
• If the source order contains price details, then the import payload must include the return charges.

Related Topics

• Order Import Template: Explained
7 Setting Up Orchestration Processes

Defining Orchestration Processes

Defining Orchestration Processes: Explained

You can set up an orchestration process so that it meets the requirements of your organization. You specify the rules that determine how Order Management Cloud creates the orchestration process at run time, and then assigns objects to the fulfillment lines that Order Management creates as part of this order processing.

You can use the predefined ShipOrderGenericProcess or ReturnOrderGenericProcess orchestration processes. You must create them and then deploy them. It is not necessary to release them.

Specifying How to Plan and Replan Orchestration Processes

You can specify how Order Management plans and replans each orchestration process. To specify planning and replanning, see Defining Orchestration Process Steps: Points to Consider.

<table>
<thead>
<tr>
<th>Type of Planning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process planning</td>
<td>Sets and helps meet the completion date for each orchestration process step and task in an orchestration process.</td>
</tr>
<tr>
<td></td>
<td>Process planning allows you to specify how to use the transformation rules that an orchestration process references to transform each source order, including planning for each step after it receives a source order from an order capture system.</td>
</tr>
<tr>
<td>Process replanning</td>
<td>Updates completion dates when a change occurs to a sales order at any point in an orchestration process. You can configure replanning in the following ways:</td>
</tr>
<tr>
<td></td>
<td>• Replan an orchestration process immediately after each orchestration process step finishes. For details, see Defining Orchestration Processes: Points to Consider.</td>
</tr>
<tr>
<td></td>
<td>• Replan an orchestration process according to some event, such as every time it receives an update from the order capture system. To control when it does this replanning, you can use a scheduled process named Plan Orchestration Processes to schedule an update at regular intervals according to the frequency that your deployment requires. For example, if your orchestration process requires planning data that is current, then you can administer this scheduled process to run the orchestration process and update the planning data each day. To view an example that uses a scheduled process, see Using Scheduled Processes to Fix Errors in Multiple Sales Orders: Procedure.</td>
</tr>
</tbody>
</table>

Related Topics

• Fixing Errors in All Sales Orders: Procedure
Defining Orchestration Processes: Examples

An orchestration process provides a way for you to define a business process. It can include steps that call the task layer service, get planning details, modify change management settings, set status conditions, and so on. The following examples describe some of the ways you can use an orchestration process to define a business process.

Call the Task Layer Service

Assume your company sells widgets and requires an orchestration process that incorporates the following company policy:

- If an invoice exceeds $100,000, then a representative must phone the customer.

You must identify the steps that your company requires to fulfill a sales order, and then define an orchestration process that includes these steps so the orchestration process mirrors your business process. To get started, use the following pseudocode to specify the sequence of calls to the Task Layer service:

1. Plant Acknowledgment.
2. Assemble.
3. Wait for status to equal COMPLETE.
4. Ship.
5. Wait for status to equal SHIPPED.
6. Call Customer.
7. Wait for Call Customer to equal COMPLETE.
8. Test for the following conditions:
   - If the invoice exceeds $100,000, then do the following work:
     - Send High Value Invoice.
     - Wait for the status for High Value Invoice to equal BILLED.
   - If the invoice does not exceed $100,000, then do the following work:
     - Invoice.
     - Wait for the status to equal BILLED.
9. End the condition.

Get Planning Details

Assume your company sells carpet and has established the following lead-times to allow the sales representative to monitor order fulfillment:

<table>
<thead>
<tr>
<th>Step</th>
<th>Lead Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule</td>
<td>Two days</td>
</tr>
<tr>
<td>Reservation</td>
<td>One day</td>
</tr>
<tr>
<td>Shipment</td>
<td>Six days</td>
</tr>
<tr>
<td>Invoice</td>
<td>Two days</td>
</tr>
</tbody>
</table>
To define an orchestration process that includes these details, you add the default lead time to each orchestration process step. If a step is delayed, then a process runs in the background that replans the process and resets the expected completion dates.

Set a Status
Assume your company sells carpet and you have an important customer who requires that you notify the receiving clerk one day before the shipping system ships the carpet. You define an orchestration process that handles the sales orders for this customer. You define an orchestration process class named Carpet Process for this process. You determine that this class must include the following statuses:

- SHIPPED
- RESERVED
- READY TO SHIP
- SHIPPED
- INVOICED

You use the Orchestration Process Status tab to define the status conditions. For example:

- If the status of the Create Shipment step is PRESHIP READY, then use the READY TO SHIP status to indicate the orchestration process status.

The order manager can use the Order Management work area to determine whether or not the orchestration process status is READY TO SHIP.

Related Topics
- Order Management Statuses: Explained

Defining Orchestration Processes: Points to Consider
This topic describes some important points to consider when you define an orchestration process.

⚠️ Caution: If you use your implementation project to migrate an orchestration process instance from a development environment to a production environment, then do not modify the process name in either environment. Modifying the name might prevent Order Management Cloud from updating references to other data in the orchestration process. For details about using an implementation project, see Implementing Order-to-Cash: Points to Consider.

Use Predefined Orchestration Processes
It is recommended that you use one of the following predefined orchestration processes before you create a new orchestration process. Using a predefined orchestration process and other predefined objects will help to reduce your development and maintenance effort. If you find that these predefined orchestration processes do not meet your business requirements, then create a new orchestration process.

<table>
<thead>
<tr>
<th>Predefined Orchestration Process</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShipOrderGenericProcess</td>
<td>Includes the following tasks:</td>
</tr>
<tr>
<td></td>
<td>• Schedule</td>
</tr>
</tbody>
</table>
### Predefined Orchestration Process

<table>
<thead>
<tr>
<th>Predefined Orchestration Process</th>
<th>Description</th>
</tr>
</thead>
</table>
| Reservation                      | - Reservation  
- Shipment  
- Invoice |
| ReturnOrderGenericProcess        | Includes the following tasks:  
- Return Receipt  
- Invoice |
| OrderFulfillmentGenericProcess   | Includes the following tasks. You can use it for various fulfillment requirements, such as back-to-back shipments, drop shipments, and so on:  
- Schedule Conditional. Starts the branching.  
- Request Supply. Starts the back-to-back branch.  
- Pause.  
- Create Back to Back Shipment Request.  
- Wait for Back to Back Shipment Request.  
- Create Purchase Request. Starts the drop ship branch.  
- Wait for Procurement.  
- Create Reservation. Starts the warehouse shipment branch.  
- Create Shipment Request.  
- Wait for Shipment Advice.  
- Merge. Ends the branching.  
- Invoice.  
- Wait for Invoice. |

### Perform the Required Configurations

To define an orchestration process that accurately represents your desired business process, it is recommended that you configure the following items, and that you configure them in the same order that the following table lists them.

<table>
<thead>
<tr>
<th>Item That Requires Configuration</th>
<th>Work That You Perform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task type</td>
<td>Define the task types that arrange fulfillment tasks in groups. Each task type references services that communicate with a type of fulfillment system. For example, a billing system. For details, see Task Types: Explained.</td>
</tr>
<tr>
<td>Orchestration process</td>
<td>Define the orchestration process that represents your business process that fulfills a fulfillment line. It includes the sequence of calls to the task layer service and planning details, change management settings, and status conditions. For details, see Defining Orchestration Processes: Procedure.</td>
</tr>
<tr>
<td>Change logic</td>
<td>Define the set of rules that determine how the orchestration process handles changes to sales orders. For details, see Processing Changes That Occur During Order Fulfillment: How It Works.</td>
</tr>
<tr>
<td>Process planning</td>
<td>Define the schedule that displays the completion date of each task and of the orchestration process. For details, see Order Management Process Planning: How It Works.</td>
</tr>
<tr>
<td>Jeopardy threshold and jeopardy priority</td>
<td>Define the level of risk that is associated with the delay of an orchestration process task as low, medium, or high. For details, see Jeopardy Threshold: Explained.</td>
</tr>
<tr>
<td>Statuses and status conditions</td>
<td>Define the status of a sales order or an object, including a task, orchestration process, fulfillment line, or order line. For details, see Orchestration Process Status: Explained.</td>
</tr>
</tbody>
</table>
Configure Replanning
You can enable the Replan Instantly feature when you define an orchestration process. It replans the orchestration process immediately after this process completes the orchestration process step, and then displays the revised order data according to the replanning work that it does. It is recommended that you use Replan Instantly only for high priority sales orders, or with sales orders that include a jeopardy threshold of less than one day. For performance reasons, it is recommended that you do not use it with an orchestration process step that is long or complex. If Replan Instantly does not contain a check mark, then Order Management updates the planning data only during the scheduled replanning.

For details about how to configure replanning for each orchestration process step, see Defining Orchestration Process Steps: Points to Consider.

Defining Orchestration Process Steps: Points to Consider
An orchestration process includes the orchestration process steps that Order Management Cloud requires to process a fulfillment line. Each step includes details that Order Management uses when it runs the step, such as the task type, task, service, dependencies, planning, change management, and so on.

To define an orchestration process step, you access the Edit Orchestration Process Definition page, and then use the Step Definition list in the Process Details area.

Specify the Orchestration Process Step
Consider the following attributes you can set when you define an orchestration process step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
</tbody>
</table>
| Step Type       | Specify the behavior of the orchestration process step. You can view these steps in an orchestration process diagram. You can set this attribute to one of the following values:  
  • **Conditional**. Runs a path in an orchestration process according to the results of a condition. You must specify a branching condition on a step immediately after the conditional step.  
  • **Merge**. Identifies the point where two or more orchestration paths reunite.  
  • **Service**. Allows this step to reference a service.  
  • **Subprocess**. Allows this step to reference an orchestration subprocess. |
| Task Type       | Specify the task type. Each task type includes services that you can use to communicate with a fulfillment system, such as a billing system. You can use the following predefined task types:  
  • Schedule  
  • Reservation  
  • Shipment  
  • Invoice  
  • Return |
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Specify the task to run. A task can include more than one step. For example, the Ship task calls the Create Shipment service. It also calls the Wait for Shipment service to wait for different status values to occur in the fulfillment system updates.</td>
</tr>
<tr>
<td>Caution:</td>
<td>If you use your implementation project to migrate an orchestration process instance from a development environment to a production environment, then do not modify the task name in either environment. Modifying this name might prevent Order Management from updating references to other data in the orchestration process. For details about your implementation project, see Implementing Order-to-Cash: Points to Consider.</td>
</tr>
<tr>
<td>Service</td>
<td>Identifies the fulfillment task service that this step calls.</td>
</tr>
<tr>
<td>Manual</td>
<td>Specify whether to wait for user input. If you define a manual task, then the orchestration process waits until the user manually completes the task in the Order Management work area. It is recommended that you define some tasks as manual tasks. For example, you can define the Schedule task as a manual task so an order manager can manually schedule all fulfillment lines at the end of the day.</td>
</tr>
<tr>
<td>Exit Criteria</td>
<td>Specify the task status that determines when to exit a wait step. For example, if the status of a shipment task changes to Shipped, then exit the wait step. If your task includes multiple wait steps, then make sure these steps do not include the same set of exit criteria. Use the Manage Task Status Conditions page to make sure only the last step or wait step uses the exit criterion.</td>
</tr>
<tr>
<td>Line Selection Criteria</td>
<td>Add a business rule that selects fulfillment lines. The orchestration process step will then process only these fulfillment lines.</td>
</tr>
<tr>
<td></td>
<td>An orchestration process might not require all of the order lines or fulfillment lines when it calls the fulfillment task service. For example, assume the item on an order line is a warranty. You typically do not ship a warranty, so you can create a business rule that specifies not to ship items that do not ship. Note the following points:</td>
</tr>
<tr>
<td></td>
<td>• The rule populates the result with the fulfillment line Ids that identify the fulfillment lines to select.</td>
</tr>
<tr>
<td></td>
<td>• The rule runs for each fulfillment line that the orchestration process is processing.</td>
</tr>
<tr>
<td></td>
<td>• If the rule does not select any fulfillment line, then the orchestration process skips the step.</td>
</tr>
<tr>
<td></td>
<td>For details, see Selecting Fulfillment Lines for Orchestration Process Steps: Procedure.</td>
</tr>
<tr>
<td>Start-After Condition</td>
<td>Specify when to pause processing before calling the next step, or when to resume processing. For details, see Pausing Orchestration Processes: Explained.</td>
</tr>
</tbody>
</table>

308
Specify Branching

An orchestration process can be linear, where step occurs in sequence with no branching, or it can contain a branch where flow travels along different paths depending on a condition. A branch can contain steps that an item might or might not require to complete fulfillment. You can use the following set ups to achieve branching logic:

- Use an assignment rule that assign one orchestration process from more than one possible orchestration process candidates.
- Use a branching condition in a single orchestration process.

The technique you use depends at least in part on the complexity of your implementation. For example, you can create several simple, linear orchestration processes, and then use an assignment rule to determine which orchestration process to run. Alternatively, you can combine these orchestration processes into a single orchestration process that uses branching conditions, and then not define an assignment rule. A more complex set up might require assignment rules and branching conditions. For an example that uses branching, see Adding Branches to Orchestration Processes: Procedure.

Consider the following attributes you can set when you specify an orchestration process that includes branching.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branching Condition</td>
<td>Specify the criteria that the condition must meet to run the steps in a branch. Note the following points:</td>
</tr>
<tr>
<td></td>
<td>• You add the branching condition on the first step of the branch, which is the first step immediately after the conditional step.</td>
</tr>
<tr>
<td></td>
<td>• If you do not add a check mark to the Otherwise option on the conditional step, then you must include a branching condition.</td>
</tr>
<tr>
<td>Evaluation Sequence</td>
<td>Specify the sequence that the orchestration process uses when it evaluates branch conditions.</td>
</tr>
<tr>
<td>Otherwise</td>
<td>If you add a check mark to the Otherwise option on the conditional step, and if the branching condition does meet the criteria you specify for the branch, then the orchestration process runs the branch that does not meet the branching condition.</td>
</tr>
</tbody>
</table>

Specify Orchestration Planning

Consider the following attributes you can set when you specify orchestration planning for an orchestration process step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning Default Branch</td>
<td>Specify the default path that Order Management uses for planning. Order Management uses this setting only if the orchestration process includes one or more conditional branches.</td>
</tr>
<tr>
<td>Fulfillment Completion Step</td>
<td>Add a check mark to this attribute to indicate that the fulfillment lines are fulfilled when this step finishes. Order Management uses this setting when it plans an orchestration process to make sure it meets the request date. The last step that occurs in chronological order in the orchestration process is not necessarily the Fulfillment Completion Step. For example, an orchestration process might use the requested ship date to indicate the completion date as the last step instead of using the shipped date to indicate the completion date.</td>
</tr>
<tr>
<td>Default Lead Time</td>
<td>Specify the duration that Order Management requires to complete the step.</td>
</tr>
</tbody>
</table>
Note the following points:

- The orchestration process uses lead-time to plan and to predict the completion date.
- If you do not define a lead-time expression for a step, then the orchestration process uses the value you set for Default Lead Time.
- If actual completion dates are available, then the orchestration process uses actual dates instead of estimated dates.
- The Gantt chart in the Order Management work area displays the planned orchestration process. Order Management uses the number of days that are past the lead time when it calculates jeopardy.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead Time UOM</td>
<td>The unit of measure for the lead-time, such as days, hours, or minutes.</td>
</tr>
<tr>
<td>Lead-Time Expression</td>
<td>A lead-time expression is a business rule that determines the amount of time you expect the orchestration process step requires to finish. You can use it to calculate planning for the orchestration process. For example, an item that uses complex packing might require a longer lead-time for shipping. The rule populates the result with a numeric value that represents the lead time. You must use BigDecimal with your lead-time expression. For example, the following expression determines the difference between the current date and the scheduled ship date:</td>
</tr>
</tbody>
</table>
|                       | \[
|                       | \text{BigDecimal.valueOf}((\text{DooSeededOrchestrationRules.DOOLHeader} / \text{childFLines.scheduleShipDate.time} - \text{CurrentDate.date.timeInMillis}) / (1000*60*60*24))
|                       | \]
|                       | where:                                                                      |
|                       | BigDecimal is a public Java class that specifies an arbitrary-precision, signed, decimal number. |
|                       | DooSeededOrchestrationRules is a method that contains a set of predefined business rules for an orchestration process. |
|                       | DOOHHeader/childFLines specifies a relationship between the parent order head and the child fulfillment lines. The forward slash (/) separates the parent from the child. |
|                       | scheduleShipDate is a sales order attribute. |
|                       | CurrentDate is a function that returns the current date. |
|                       | timeInMillis converts the current date to milliseconds. |
|                       | 1000 is the number of milliseconds in one second. |
|                       | 60 is the number of minutes in one hour. |
|                       | 60 is the number of seconds in one hour. |
|                       | 24 is the number of hours in one day. |
|                       | For details, see Defining Lead-Times for Orchestration Process Steps: Procedure. |

Specify Change Management

Consider the following attributes that you can set when you specify change management for an orchestration process step.
## Attribute | Description
--- | ---
Hold on wait | Sends a message to the fulfillment system for each active step when the orchestration process receives a change order.

Use Transactional Item Attributes | If you enable transactional item attributes, then Order Management examines the transactional item attributes to help it determine the differences that exists between the change order and the previous version of this order.

Use Flexfield Attributes | If you enable flexfield attributes, then Order Management examines them to help it determine the differences that exists between the change order and the previous version of this order.

Compensation Pattern | Specify the set of rules that determine how to handle each step that ran before Order Management received a change order.

For example, assume a change order requests a change from carpet to tiles. In this example, Order Management must cancel a number of the previous steps that it ran. If it already scheduled carpet for shipping, then Order Management must cancel the steps that scheduled the shipping. Order Management creates a different set of fulfillment lines for tiles when it processes the change order, so it must use a different orchestration process because a tile order requires more time to fulfill and it uses a different contractor. Order Management must cancel most of the previous steps, but it does not cancel the Measure step because the room dimensions are still accurate.

If you do not specify a compensation pattern, then Order Management might process a step as an update, depending on the context of the operation. It might rerun some steps. If Order Management cannot compensate a step for some reason, then it bypasses the step, and then compensates the next step that it encounters.

### Related Topics
- Implementing Order-to-Cash: Points to Consider

## Defining Orchestration Processes: Procedure

This topic describes how to define an orchestration process. It uses an example that calls a scheduling service.

### Summary of the Work

To define an orchestration process, do the following work:

1. Prepare to define the orchestration process.
2. Define the orchestration process.
3. Define the orchestration process steps.
4. Deploy the orchestration process. For details, see Deploying Orchestration Processes: Procedure.

This topic includes example values. You might use different values, depending on your business requirements.

### Prepare to Define the Orchestration Process

Depending on the complexity of your orchestration process, you might need to perform one or more of the following steps:

1. Do the tasks in the Setup and Maintenance work area that allow you to define an orchestration process.
2. Define the task types that the orchestration process requires. Also define the tasks and services that these task types reference.
3. Define the status codes that the orchestration process requires and specify how the task types, fulfillment lines, and your orchestration process will use them.

4. Define the orchestration subprocesses that the orchestration process will reference.

5. Define the status catalogs that the orchestration process will use for status conditions. Define these catalogs in one of the following items:
   - Oracle Product Model
   - Oracle Product and Catalog Management
   - Oracle Product Hub

**Define the Orchestration Process**

Define the orchestration process:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Orchestration Process Definitions.
4. On the Manage Orchestration Process Definitions page, click **Actions**, and then click **Create**.
5. On the Create Orchestration Process Definition page, set the following attributes, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name</td>
<td>Enter an orchestration process name that does not include any spaces, such as OrchestrationProcess1.</td>
</tr>
<tr>
<td>Process Display Name</td>
<td>Enter any value. Do not include spaces.</td>
</tr>
<tr>
<td>Process Class</td>
<td>Select an orchestration process class that includes the statuses that you require for the orchestration process. For this example, select <strong>Ship Order Class</strong>.</td>
</tr>
<tr>
<td>Set</td>
<td>Select a set that allows the business unit to use this orchestration process. For this example, select <strong>Common Set</strong>. A set is a collection of business units. It provides a way to organize business units and control the business units that can access an orchestration process. The predefined Common Set allows you to provide access to many business units. You must add your business unit to the set that you specify.</td>
</tr>
</tbody>
</table>

**Define the Orchestration Process Steps**

Define the orchestration process steps:

1. On the Step Definition tab, click **Actions**, and then click **Add Row**.
2. Set the following attributes, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Enter text that describes the purpose of the step. For this example, enter Schedule Product.</td>
</tr>
<tr>
<td>Step Type</td>
<td>Select a value that indicates the type of processing that this step performs. For example, if this step must do conditional branching, then choose Conditional. This example calls a service, so select <strong>Service</strong>.</td>
</tr>
</tbody>
</table>
Attribute | Description
---|---
Task Type | Select the group of services that Order Management Cloud uses to perform a fulfillment task. For this example, select Schedule.

Task | Select the task that this step must perform. For this example, select Schedule.

Service | Identify the service that this step must call. For this example, select Create Scheduling.

3. Repeat steps 1 and 2, using the following values.

**Tip:** To maintain the correct sequence when you add each step in this procedure, click the step that you most recently added, and then click Add Row. To make sure you do not lose any work, click Save after you add each step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Reserve Product</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task Type</td>
<td>Reservation</td>
</tr>
<tr>
<td>Task</td>
<td>Reserve</td>
</tr>
<tr>
<td>Service</td>
<td>Create Inventory Reservation</td>
</tr>
</tbody>
</table>

4. Repeat steps 1 and 2, using the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Ship Product</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task Type</td>
<td>Shipment</td>
</tr>
<tr>
<td>Task</td>
<td>Ship</td>
</tr>
<tr>
<td>Service</td>
<td>Create Shipping</td>
</tr>
</tbody>
</table>

5. Repeat steps 1 and 2, using the following values. Note that you also specify the exit criteria in this step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Wait for Product Shipment</td>
</tr>
</tbody>
</table>
### Attribute | Value
--- | ---
Step Type | Service
Task Type | Shipment
Task | Ship
Service | Wait for Shipment
Exit Criteria | Shipped

6. Repeat steps 1 and 2, using the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Create Invoice</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task Type</td>
<td>Invoice</td>
</tr>
<tr>
<td>Task</td>
<td>Invoice</td>
</tr>
<tr>
<td>Service</td>
<td>Create Billing Lines</td>
</tr>
</tbody>
</table>

7. Repeat steps 1 and 2, using the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Wait for Invoice</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task Type</td>
<td>Invoice</td>
</tr>
<tr>
<td>Task</td>
<td>Invoice</td>
</tr>
<tr>
<td>Service</td>
<td>Wait for Billing</td>
</tr>
</tbody>
</table>

**Deploying Orchestration Processes: Procedure**

You must release, and then deploy an orchestration process that you define so that it is available throughout Order Management Cloud. You deploy it on an instance of Order Management.
If you use the ShipOrderGenericProcess or the ReturnOrderGenericProcess predefined orchestration process, then you must deploy it, but it is not necessary to release it.

Summary of the Work
To deploy an orchestration process that you have defined, do the following work:

1. Release the orchestration process.
2. Deploy the orchestration process.

Release the Orchestration Process
Release the orchestration process:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup page, click Order Management.
3. Search for, and then open Manage Orchestration Process Definitions.
4. On the Manage Orchestration Process Definitions page, locate the orchestration process definition that you must release, click Actions, and then click Edit.
5. On the Edit Orchestration Process Definition page, click Actions, and then click Release.

Order Management does the following work when you release an orchestration process:

1. Validates the orchestration process logic to make sure that it correctly created this process. If it:
   a. Identifies any errors during validation, then it stops the release, and then displays an error icon next to the orchestration process name. It keeps these errors until the next time it runs the validation. You must resolve these errors before you can continue the release.
   b. Does not identify any errors during validation, then it continues the release. An orchestration process that does not contain any errors is valid. A valid orchestration process might include warning messages, but these messages do not stop the release from proceeding.
2. Finishes the validation.
3. Updates the status of the orchestration process to Released.
4. Makes the orchestration process definition read-only.
5. Creates, and then saves the BPEL artifacts that Order Management uses to deploy and run the orchestration process in a production environment.

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

Deploy the Orchestration Process
Deploy the orchestration process:

1. Click Actions, and then click Deploy Process.
2. In the Deploy Process dialog, enter your password, and then click Deploy.

If you cannot access the Manage Orchestration Process Definition pages to deploy an orchestration process for some reason, then you can use Oracle Middleware to deploy it. For details, see Using Oracle Middleware to Deploy Orchestration Processes: Procedure.
Using Oracle Middleware to Deploy Orchestration Processes: Procedure

If you cannot use the Manage Orchestration Process Definition page to deploy an orchestration process for some reason, then you can use Oracle Middleware to deploy it.

Summary of the Work

Use Oracle Middleware to deploy an orchestration process that you have defined:

1. Release the orchestration process. For details, including important guidelines, see Deploying Orchestration Processes: Procedure.
2. Download the orchestration process.
3. Modify the SOA configuration plan.
4. Deploy the JAR file.

Download the Orchestration Process

You deploy the downloaded artifacts to the server. You can use Oracle Setup Manager to export the artifacts, and Oracle Middleware to deploy them.

Download the orchestration process:

1. On the Manage Orchestration Process Definitions page, select the orchestration process that you must deploy, and then click Edit.
2. In the Download Generated Process dialog, click Download.
3. Save the archive file that Order Management Cloud displays to a local directory.
4. Open the archive file that you saved.

Modify the SOA Configuration Plan

You can use an Oracle Service-Oriented Architecture (SOA) configuration plan to define the URL and properties to use for different environments. Order Management uses this plan to search the SOA project for values that it must replace so that the project supports each environment.

Modify the following SOA configuration plan. Replace each host name with the Oracle Application Development Framework (ADF) server and replace each port with the port that your organization uses for Order Management and the server and port that Supply Chain Management uses for SOA. Use the external-facing URLs of the servers:

```xml
<?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">
```
```
Deploy the JAR File

Deploy the JAR file. You can use one of the following tools:

- Oracle Enterprise Manager Middleware Control
- Ant command line tool
- Oracle WebLogic Scripting Tool

The JAR file resides in a Deploy folder that resides in a folder that uses the name of the orchestration process that you downloaded.

For information about how to deploy an SOA composite application, see Oracle Fusion Middleware Administrator’s Guide for Oracle SOA Suite and Oracle Business Process Management Suite.

Defining Lead-Times for Orchestration Process Steps: Procedure

This topic describes how to define a rule that sets the lead-time for an orchestration process step according to a set of conditions.

Assume you are the order administrator of a company that sells glass. The lead-time required to ship the item varies depending on where the inventory organization is located, so you implement the following business rules:

- If the inventory organization is located in Denver, then use a two day lead-time
- If the inventory organization is not located in Denver, then use a four day lead-time

Summary of the Work

To define the lead-time for an orchestration process step, do the following work:

1. Create the If statement for the first rule.
2. Create the Then statement for the first rule.
3. Create the If statement for the second rule.
4. Create the Then statement for the second rule.

This topic includes example values. You might use different values, depending on your business requirements.

For a demonstration that creates a business rule, see Creating Business Rules: Demo.
Create the If Statement for the First Rule

Create the If statement for the first rule:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Orchestration Process Definitions.
4. On the Manage Orchestration Process Definitions page, locate the CallCustomerWhenLargeInvoice orchestration process, click **Actions**, and then click **Edit**.

   For this example, you edit an orchestration process named CallCustomerWhenLargeInvoice. For details about how to create this process, see Adding Branches to Orchestration Processes: Procedure.

5. On the Edit Orchestration Process Definitions page, in the Process Details area, in the Step Definition list, click **Planning**.
6. In the Ship Product row, set the following value.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead-Time UOM</td>
<td>Days</td>
</tr>
</tbody>
</table>

7. In the Lead-Time Expression column, click **Click for Rule**.
8. In the Lead-Time Expression dialog, click **Add Rule**, and then click **Expand**.

   If you write a rule for an orchestration process that includes multiple fulfillment lines, then it is recommended that you use Advanced Mode. However, in this example, all lines are treated the same way, so Advanced Mode is not required. To use Advanced Mode, click **Show Advanced Settings**, add a check mark to **Advanced Mode**, and then edit the rule.

9. In the If area, click **Left Value**, expand **DOOSeededOrchestrationRules**, expand **DOOFline**, click **InventoryOrganizationId**, and then click **OK**.
10. In the Right Value field, enter **1234440**.

   For this example, assume 1234440 is the inventory organization ID for Denver.

Create the Then Statement for the First Rule

Create the Then statement for the first rule:

1. In the Then area, click **Add Action**, and then click **Assert New**.
2. Click **Select a Target**, and then click **DooSeededOrchestrationRules.Result**.
3. Click **Edit Properties**.
4. In the Edit Properties dialog, in the ResultObjKey row, enter the following value, and then click **OK**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>BigDecimal.valueOf(2) why BigDecimal.valueOf why not only 2?</td>
</tr>
</tbody>
</table>

Create the If Statement for the Second Rule

Create the If statement for the second rule:

1. In the Lead-Time Expression dialog, in the Lead-Time Expression Set area, click **Add Rule**.
2. In the lower portion of the dialog where the application added Rule 2, click **Expand**.
3. In the If area for Rule 2, click Left Value, expand DOOSeededOrchestrationRules, expand DOOFline, select InventoryOrganizationId, and then click OK.

Tip: In some fields, you can copy the value from one field to another instead of using the drop-down list. For example, you can copy DooSeededOrchestrationRules.DOOFline.inventoryOrganizationId from the Left Value field of the first rule, and then paste it into the Left Value field of the second rule.

4. Click the down arrow for the operator, and then click isn't.

5. In the Right Value field, enter 1234440.

Create the Then Statement for the Second Rule

Create the Then statement for the second rule:

1. In the Then area for Rule 2, click Add Action, and then click Assert New.
2. Click Select a Target, and then click DooSeededOrchestrationRules.Result.
3. Click Edit Properties.
4. In the Edit Properties dialog, in the ResultObjKey row, enter the following value, and then click OK.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>BigDecimal.valueOf(4)</td>
</tr>
</tbody>
</table>

5. Click Save.

6. On the Edit Orchestration Process Definition page, click Save.

Related Topics

- Creating Business Rules: Demo

Selecting Fulfillment Lines for Orchestration Process Steps: Procedure

You can create a business rule that selects fulfillment lines, and then specify whether the orchestration process step will process these fulfillment lines. The example in this topic describes how to create a line selection rule that makes sure Order Management Cloud does not attempt to ship a nonshippable item.

Assume you work for a company that sells DVRs. The sales order includes multiple fulfillment lines for each of the following items:

- DVR
- Remote control
- Instruction manual
- Extended warranty

Your customers can purchase the extended warranty as a contract online, but it is not a shippable item, so Order Management must not attempt to send it to the fulfillment system during the Shipment task. You create the following business rule:

- If Order Management cannot ship the item, then do not attempt to ship the item.
This example includes a business rule that requires you to use a dictionary, fact, and other objects in a business rule. It is strongly recommended that you familiarize yourself with these objects before you proceed. For details, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then open the attachment that describes how to use business rules.

Summary of the Work
Set up an orchestration process step so it processes fulfillment lines according to a business rule:

1. Create the If statement.
2. Create the Then statement.

You will create the following line selection rule.

**Line-Selection Criteria Set**

![Image of Line-Selection Criteria Set](image)

**IF**

- DooSeededOrchestrationRules.DOOFLine: is "Y"

**THEN**

- assert new DooSeededOrchestrationRules.Result (resultObjKey)

where:

- DooSeededOrchestrationRules is a dictionary that contains a set of predefined functions, variables, objects, and other data you can use to define behavior for an orchestration process.
- DOOFLine contains fulfillment line attributes, such as orderedQty, customerPONumber, creationDate, and so on. You can reference these attributes and use their values in your rule.
- shippableFlag is a fulfillment line attribute.
- Y is one possible value of shippableFlag.
Create the If Statement

Create the If statement:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Orchestration Process Definitions.
4. On the Manage Orchestration Process Definitions page, locate the orchestration process named CallCustomerWhenLargeInvoice, click **Actions**, and then click **Edit**.
   
   For this example, you edit the CallCustomerWhenLargeInvoice orchestration process. For details about how to create this process, see Adding Branches to Orchestration Processes: Procedure.
5. On the Edit Orchestration Process Definitions page, in the Process Details area, in the Step Definition list, locate the Ship Item row.
6. In the Ship Item row, in the Line Selection Criteria column, click **Click for Rule**.
7. In the Line Selection Criteria dialog, click **Add Rule**, and then click **Expand**.
8. In the If area, click **Left Value**, expand **DOOSeededOrchestrationRules**, expand **DOOFline**, click **ShippableFlag**, and then click **OK**.
9. In the Right Value field, enter “Y”. You must include the double quotation marks.

Create the Then Statement

You will create the following Then statement:

- **assert new** resultObjKey:DOOSeededOrchestrationRules.DOOFLine.fulfillLineId

where:

- **resultObjKey** is a property of the Result fact. It specifies the fulfillment line to examine.
- **DooSeededOrchestrationRules.DOOFLine.fulfillLineId** specifies to use the value of the fulfillLineId attribute in the DOOFLine fact of the DooSeededOrchestrationRules dictionary.

Create the Then statement:

1. In the Then area, click **Add Action**, and then click **Assert New**.
2. Click **Select a Target**, and then click **DooSeededOrchestrationRules.Result**.
3. Click **Edit Properties**.
4. In the Properties dialog, in the ResultObjKey row, click **Value**.
5. In the Condition Browser, enter **DOOFline**, wait a moment, expand **DooSeededOrchestrationRules.DOOFLine**, click **fulfillLineId**, click **OK**, and then click **OK** again.
6. In the Line Selection Criteria dialog, click **Validate**, make sure the Validation Log that displays does not contain any errors, and then click **Save**.
7. On the Edit Orchestration Process Definition page, click **Save**.

Defining Subprocesses In Orchestration Processes

Orchestration Subprocesses: Explained

You can use an orchestration subprocess to increase the efficiency of your design. You can add a subprocess to an orchestration process, and then reuse this subprocess in another orchestration process. You can also use a subprocess
to manage a large orchestration process. In general, it is easier to manage a large orchestration process that includes sub-processes than it is to manage a large orchestration process that does not include sub-processes.

**Example of a Subprocess**

Assume your organization uses different steps when it processes an order, depending on the item. However, all orders use the following billing steps:

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

So, you create an orchestration subprocess that includes these steps, and then add this subprocess to an orchestration process that you define.

**Adding Branches to Orchestration Processes: Procedure**

You can create a branching condition that determines whether or not to run an orchestration process branch.

For example, assume you are the order administrator of a company that sells commercial computer systems, and you must implement the following business rule:

- If the sales order is valued at $50,000.00 or more, then make sure a representative calls the customer before sending an invoice for this order.
The following diagram illustrates the orchestration process flow for this example. Each step in this flow includes the step number, task name, task type, and service name. 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.** The orchestration process runs the steps on the branch that notify the representative.
- **Not met.** The orchestration process runs the steps on the branch that do not notify the representative.

Order Management Cloud adds an empty default branch when it runs the orchestration process. If the orchestration process includes only one branch, then it is not necessary to set an Otherwise condition.

This example uses a rule that includes only one fulfillment line. To write a rule for an orchestration process that includes multiple fulfillment lines, it is recommended that you use advanced rules. For details, search for Doc ID 2051639.1 (White...
Summary of the Work

Create a branch in an orchestration process:

1. Set up routing that notifies the representative.
2. Define the orchestration process.
3. Define the orchestration process steps.
4. Create the If statement.
5. Create the Then statement.
6. Test your work.

This topic includes example values. You might use different values, depending on your business requirements.

Set Up Routing That Notifies the Representative

This example sends a notification to a representative to call the customer if the invoice is valued at $50,000.00 or more. You must set up the routing that enables this notification. This setup is specific to this example. Other branching usages might not require this setup, or they might require some other setup.

Set up routing that notifies the representative:

1. Create the routing rules that enable the send. For details, see Creating Routing Rules That Send Notifications: Procedure.
2. Create the connector that you referenced in the routing rules in step 1.
3. Use the Manage Web Service Details page to create the connector.

Define the Orchestration Process

Define a new orchestration process. For this example, use the following values in the header of this orchestration process. For details about how to define an orchestration process and orchestration process steps, see Defining Orchestration Processes: Procedure.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name</td>
<td>CallCustomerWhenLargeInvoice</td>
</tr>
<tr>
<td>Process Display Name</td>
<td>Call the Customer</td>
</tr>
<tr>
<td>Process Class</td>
<td>Ship Order Class</td>
</tr>
<tr>
<td>Set</td>
<td>Common Set</td>
</tr>
</tbody>
</table>

Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then open the attachment that describes how to use business rules.

For a general demonstration about how to create a business rule, see Creating Business Rules: Demo.
Define the Orchestration Process Steps

You will define the following steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Task Name</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Schedule Item</td>
<td>Service Schedule</td>
</tr>
<tr>
<td>200</td>
<td>Reserve Item</td>
<td>Service Reserve</td>
</tr>
<tr>
<td>300</td>
<td>Ship Item</td>
<td>Service Ship</td>
</tr>
<tr>
<td>400</td>
<td>Wait for Item to Ship</td>
<td>Task Ship</td>
</tr>
<tr>
<td>500</td>
<td>Conditional Node</td>
<td>Service Ship</td>
</tr>
<tr>
<td>600</td>
<td>Otherwise Step - Create Invoice</td>
<td>Service Schedule</td>
</tr>
<tr>
<td>700</td>
<td>Otherwise Step - Wait for Invoice</td>
<td>Task Invoice</td>
</tr>
<tr>
<td>800</td>
<td>Phone Customer</td>
<td>Service Activity</td>
</tr>
<tr>
<td>900</td>
<td>Wait to Phone Customer</td>
<td>Task Activity</td>
</tr>
<tr>
<td>1000</td>
<td>Create Invoice</td>
<td>Service Invoice</td>
</tr>
<tr>
<td>1100</td>
<td>Wait for Invoice</td>
<td>Task Invoice</td>
</tr>
<tr>
<td>1200</td>
<td>Merge</td>
<td>Merge</td>
</tr>
</tbody>
</table>

Define the orchestration process steps:

1. Add the following step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Schedule Item</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task</td>
<td>Schedule</td>
</tr>
<tr>
<td>Service</td>
<td>Create Scheduling</td>
</tr>
</tbody>
</table>

2. Add the following step.

Tip: To maintain the correct sequence when you add each step in this procedure, click the step that you most recently added, and then click Add Row.
3. Add the following step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Reserve Item</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task</td>
<td>Reserve</td>
</tr>
<tr>
<td>Service</td>
<td>Create Inventory Reservation</td>
</tr>
</tbody>
</table>

4. Add the following step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Ship Item</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task</td>
<td>Ship</td>
</tr>
<tr>
<td>Service</td>
<td>Create Shipping</td>
</tr>
</tbody>
</table>

5. Add the following step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Wait for Item to Ship</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task</td>
<td>Ship</td>
</tr>
<tr>
<td>Service</td>
<td>Wait for Shipment</td>
</tr>
<tr>
<td>Exit Criteria</td>
<td>Shipped</td>
</tr>
<tr>
<td>Fulfillment Completion Step</td>
<td>Contains a check mark.</td>
</tr>
</tbody>
</table>

5. Add the following step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Conditional Node</td>
</tr>
<tr>
<td>Step Type</td>
<td>Conditional</td>
</tr>
</tbody>
</table>
6. Add the following step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Phone Customer</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task</td>
<td>Activity</td>
</tr>
<tr>
<td>Service</td>
<td>Create Activity</td>
</tr>
<tr>
<td>Evaluation Sequence</td>
<td>1</td>
</tr>
</tbody>
</table>

7. Add the following step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Wait for Phone Customer</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task</td>
<td>Activity</td>
</tr>
<tr>
<td>Service</td>
<td>Wait for Activity</td>
</tr>
<tr>
<td>Exit Criteria</td>
<td>Completed</td>
</tr>
</tbody>
</table>

8. Add the following step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Create Invoice</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task</td>
<td>Invoice</td>
</tr>
<tr>
<td>Service</td>
<td>Create Billing Lines</td>
</tr>
</tbody>
</table>

9. Add the following step.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Wait for Invoice</td>
</tr>
<tr>
<td>Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task</td>
<td>Invoice</td>
</tr>
<tr>
<td>Service</td>
<td>Wait for Billing</td>
</tr>
<tr>
<td>Exit Criteria</td>
<td>Billed</td>
</tr>
</tbody>
</table>

10. In the Step Definition list, click the Conditional Node step, and then add the following step. This step creates the Otherwise branch.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Otherwise Step - Create Invoice</td>
</tr>
<tr>
<td>Task Type</td>
<td>Invoice</td>
</tr>
<tr>
<td>Task</td>
<td>Otherwise Create Invoice</td>
</tr>
</tbody>
</table>

Each task name that you associate with a task type must be unique. You associated the Invoice task with the Invoice task type earlier in this procedure, so you must create a new task name. To do this, in the Task attribute, click the Search down arrow, and then click Create. In the Create Task Name dialog, enter the following values, and then click Save and Close:

- Code: 1
- Name: Otherwise_Create_Invoice
- Display Name: Otherwise Create Invoice
- Task Type: Invoice

<table>
<thead>
<tr>
<th>Service</th>
<th>Create Billing Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation Sequence</td>
<td>2</td>
</tr>
<tr>
<td>Otherwise</td>
<td>Contains a check mark.</td>
</tr>
<tr>
<td>Planning Default Branch</td>
<td>Contains a check mark.</td>
</tr>
</tbody>
</table>

11. In the Step Definition list, click the step named Otherwise Step - Create Invoice, and then add the following step. This step creates the Wait for Invoice Step on the Otherwise branch.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Otherwise Step - Wait for Invoice</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
</tbody>
</table>
### Attribute | Value
--- | ---
Task | Otherwise Create Invoice
Service | Wait for Billing
Exit Criteria | Billed

12. In the Step Definition list, click the Wait for Invoice step, and then add the following step. This step merges the branch back to the main flow.

| Attribute | Value |
--- | --- |
Step Name | Merge |
Step Type | Merge |

**Create the If Statement**

You will create the following business rule in the remaining part of this procedure:

**Branching Condition Rule Set**

**IF**

- DooSeededOrchestrationRules.DOOFLine_.extendedAmount
  - more than
  - 50000

**THEN**

- assert new
  - DooSeededOrchestrationRules.Result
  - (resultObj:DooSeededOrchestrationRules)

You will start by creating the following If statement:

- If the extendedAmount attribute on the fulfillment line is more than 50000
Create the If statement:

1. In the Step Definition list, in the Phone Customer step, in the Branching Condition column, click Click for Rule.
2. In the Branching Condition Rules dialog, click Add Rule, and then click Expand.
3. Delete the value Rule 1, and then enter Condition for invoices valued at more than 50000 dollars.
4. Click Left Value.
5. In the Condition Browser dialog, expand DooSeededOrchestrationRules, expand DOOFLine, click extendedAmount, and then click OK.

where:
- DooSeededOrchestrationRules is a dictionary that includes predefined rule sets, facts, functions, variables, bucket sets, links, and functions for order orchestration.
- DOO means distributed order orchestration, which is an earlier name for order orchestration.
- FLine means fulfillment line
- extendedAmount is a fulfillment line attribute that stores the total value of the sales order.

6. Click Is, and then click More Than.
7. Click Right Value.
8. In the Condition Browser dialog, enter 50000, and then click OK. Do not include commas.

Create the Then Statement

You will create the following Then statement:

- ( resultObj:DooSeededOrchestrationRules.Boolean.TRUE )

where:
- resultObj is a variable in the DooSeededOrchestrationRules dictionary. You use it to store the result of the business rule.
- Boolean sets the value of resultObj to TRUE.

Create the Then statement:

1. In the Then area, click Add Action, and then click Assert New.
2. Click Select a Target, and then click DooSeededOrchestrationRules.Result.
3. Click Edit Properties.
4. In the Properties dialog, in the ResultObj row, click Value.
5. In the Condition Browser dialog, expand DooSeededOrchestrationRules, expand Boolean, click True, and click OK.
6. Make sure the Properties dialog contains the following value, and then click **OK**.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>resultObj</td>
<td>Object</td>
<td>DocSeededOrchestrationRules.Boolean.TR</td>
<td></td>
</tr>
<tr>
<td>resultObjKey</td>
<td>Object</td>
<td></td>
<td></td>
</tr>
<tr>
<td>viewRowImp1</td>
<td>oracle.jbo.server.ViewRo...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. In the Branching Condition Rules dialog, click **Save**.
8. On the Edit Orchestration Process Definition page, click **Save**.

**Test Your Work**

Test your work:

1. Verify that you correctly defined the orchestration process steps and flow:
   - In the header of the Edit Orchestration Process Definition page, click **Actions**, and then click **Generate Process Diagram**.
   - Make sure the diagram that this page displays includes the same steps and logic that the diagram at the beginning of this topic displays.

2. Test the nonbranching flow:
   - In the Navigator, click **Order Management**.
   - In the Order Management work area, create a sales order that is valued at less than $50,000.00.
   - Verify that Order Management Cloud ships the item without requesting that the user call the customer.

3. Test the branching flow:
   - In the Order Management work area, create a sales order that is valued at more than $50,000.00.
   - Verify that Order Management does not ship the item until the activity that the Wait for Phone Customer step references reaches a Completed state.

**Related Topics**

- Creating Business Rules: Demo

**Parallel Processing: Explained**

Order Management Cloud can simultaneously process more than one task. You can use this parallel processing to process different sets of fulfillment lines that each require a different set of processing tasks.
For example, assume a customer purchases a laptop computer that includes a service contract, and that your business flow requires that the fulfillment system ship the laptop at the same time that it sends the service contract through email. You can use parallel processing to ship the laptop and send the email at the same time.

The following figure illustrates part of an orchestration process that includes branches that run in parallel with one another. Step 300 begins the parallel branching. Order Management runs steps 400 and 500 in one branch at the same time that it runs steps 600 and 700 in a parallel branch. The Create Invoice and the Wait for Billing steps occur after step 800. This figure does not include these final steps.

Order Management plans the orchestration process after it finishes each task. However, if a task requires several days to finish, then planning data might not be current because Order Management cannot replan the branch while it waits for a long-running task to finish. For details about how to use a scheduled process named Plan Orchestration Processes to schedule an update that occurs at regular intervals, see Defining Orchestration Processes: Explained.
Guidelines for Using Parallel Processing
You set up an orchestration process so that it uses parallel processing in the same way that you set up an orchestration process to use a conditional branch, with the following differences:

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

Note the following:

- It is recommended that you use parallel processing when multiple, long-running tasks exist that must run at the same time. For example, to send simultaneous requests to two different fulfillment systems that typically require elapsed time to reply.
- It is recommended that you do not use parallel processing when the reply from a task request occurs almost immediately.
- You can set up parallel processing so that more than one task can run at the same time.
- You can filter the fulfillment lines of a group so that the tasks for some lines in this group run at the same time as tasks for the other lines.
- 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.
- Order Management does not allow you or a fulfillment system user to split a task that resides in a parallel branch.

Assigning Orchestration Processes

Assigning Orchestration Processes: Explained
You can define an assignment rule that assigns an orchestration process to one or more fulfillment lines of a sales order.

Note the following points:

- Order Management Cloud does not version assignment rules so the changes you release take effect immediately. You can save rules without releasing them.
- You do not need to specify versions or effectivity dates in the assignment rules because the orchestration process controls these versions and dates.
- You must define orchestration processes or at least know the names that you will use for them before you create an assignment rule. You add the orchestration process name to bucket sets that allow you to select them when you create the assignment rule.
- You can assign an orchestration process according to each set of unique conditions.
- You can use Otherwise logic to set up a default orchestration process for each orchestration group.
- You can create an assignment rule that references data from a different product model. For details, see Getting Data from Different Product Models When Using Oracle Business Rules: Procedure.

Orchestration Groups
You create assignment rules according to orchestration groups. An orchestration group is a collection of fulfillment lines that Order Management processes together. Each orchestration group is a subset of a sales order.
Note the following points:

- You define an orchestration group as a shipment set, configured item, or set of order lines.
- An assignment rule processes each orchestration group at run time, so you must add a test that links each fulfillment line with the group. The predefined assignment rules already include this test.

You can use the following predefined orchestration groups.

<table>
<thead>
<tr>
<th>Orchestration Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>Groups all fulfillment lines that fulfill items that are not configured or are finished items to the same orchestration process.</td>
</tr>
<tr>
<td>Configured Item or Kit</td>
<td>Groups all fulfillment lines that fulfill a configured item or kit to the same orchestration process.</td>
</tr>
<tr>
<td>Shipment Set</td>
<td>Groups all fulfillment lines that fulfill a shipment set to the same orchestration process.</td>
</tr>
</tbody>
</table>

Assigning Orchestration Processes: Examples

The following examples assign an orchestration process.

<table>
<thead>
<tr>
<th>Assign an Orchestration Process According to...</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>Each sales order for ceramic tile must use the same processing steps. You write a process assignment rule that assigns an orchestration process named Tile Processing to each order line that includes tile as the product.</td>
</tr>
<tr>
<td>Customer</td>
<td>Customer A requires an inspection step for each sales order. You write a process assignment rule that assigns an orchestration process named Customer A Process to each order line that includes Customer A in the order header.</td>
</tr>
<tr>
<td>Destination</td>
<td>Each sales order that the fulfillment system must ship to a country that resides outside of your current location requires different handling, such as completing customs forms. You write a process assignment rule that assigns an orchestration process named International Orders to each order line that includes a foreign country in the ship-to address in the header.</td>
</tr>
</tbody>
</table>

Assigning Orchestration Processes: Procedure

This topic describes how create an assignment rule that assigns an orchestration process to a fulfillment line. For details about how to use a simplified rule builder, see Using Visual Information Builder to Create Rules: Explained.
Assume you are an order administrator for a company that receives a large quantity of sales orders each day for laptop computers. You must create an assignment rule that assigns an orchestration process for each bulk order according to the ordered quantity. You create the following rule:

- If the ordered quantity is large, then assign the orchestration process named CallCustomerWhenLargeInvoice to the fulfillment line.

**Summary of the Work**

Assign an orchestration process:

1. Add a rule to a decision table.
2. Add an assignment condition.
3. Add the buckets.
4. Add the bucket set to the assignment condition.
5. Specify the action to take.

This topic describes how to use decision tables and bucket sets. It is strongly recommended that you get details about and how to use them. Search for document ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, download the attachment that describes how to use business rules, and then see the chapter that describes decision tables and bucket sets.

This topic includes example values. You might use different values, depending on your business requirements.

**Add a Rule to a Decision Table**

Add a rule to a decision table:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Orchestration Process Assignment Rules.
4. On the Manage Orchestration Process Assignment Rules page, in the View list, click **AssignProcess**.

AssignProcess is a predefined decision table that you set up in this topic. It uses Advanced Mode, which allows you to do the following work:

- Set a variable and an alias for each object.
- Define a variable so you can use a simple name to access it.
- Define a hierarchy, which can improve performance.
- Arrange fulfillment lines in a group.

AssignProcess comes predefined with the following aliases. An alias simplifies the text that Order Management displays. For example, the Group alias references the assignLaunchRules.ResponseGL object. So, you can specify Group instead of assignLaunchRules.ResponseGL in some fields when you define an assignment rule. You can also add new aliases. Take a moment to examine the following predefined aliases, and then click **Collapse Pane** so that you can more easily view the decision table.

<table>
<thead>
<tr>
<th>Alias</th>
<th>Object That the Alias References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>assignLaunchRules. ResponseGL</td>
</tr>
<tr>
<td>fline</td>
<td>assignLaunchRules. FulfillLineVO</td>
</tr>
</tbody>
</table>
5. In the decision table, immediately above the Conditions section, click Add, and then click Add Rule.

Notice that the decision table includes a new rule in the R1 column.

Add an Assignment Condition

You will create the following part of the If statement. This statement determines whether the ordered quantity is equal to quantity. Note that fline.Ordered Quantity already contains a value. You will add the value later in this topic.

Add an assignment condition:

1. In the decision table, immediately above the Conditions section, click Add Rule, and then click Add Condition.

Notice that the Condition Browser includes the Group, fline, and Header aliases that you noticed earlier.

2. In the Condition Browser, expand fline, click Ordered Quantity, and then click OK.

Notice that the Conditions section of the decision table includes a new fline.OrderedQty row.
Add Buckets

You will create the buckets that you can select to complete the If statement. You create the following buckets.

Add the buckets:

1. In the decision table, in the Conditions area, click the `fline.OrderedQty` cell.

   When you click `fline.OrderedQty`, notice that the Local List of Ranges list and a pencil icon display immediately above the Condition table.

2. Click `Edit Bucketset` immediately to the right of Local List of Ranges.

3. In the Bucketset Editor dialog, specify the following values for the bucket set.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Bulk Ranges</td>
</tr>
<tr>
<td>Description</td>
<td>Set of ranges to use for bulk orders.</td>
</tr>
<tr>
<td>Data Type</td>
<td>int</td>
</tr>
</tbody>
</table>

4. Create the small bucket. Click `Add Bucket`, and then set the following value in the row that contains an End Point value of 1.
5. Create the medium bucket. Click **Add Bucket**, and then set the following values in the row that contains an End Point value of 2. You modify this End Point value to 100. Notice that the Bucket set Editor updates the values in the Range and Alias fields when you modify the End Point.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Point</td>
<td>100</td>
<td>Medium</td>
</tr>
</tbody>
</table>

6. Create the large bucket. Click **Add Bucket**, and then set the following values in the row that contains an End Point value of 101. You modify this End Point value to 1000.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Point</td>
<td>1000</td>
<td>Large</td>
</tr>
</tbody>
</table>

7. Click **OK**, and then click **Save**.

**Add Bucket Set to Assignment Condition**

You will add the following details in the decision table.

Add the bucket set to the assignment condition:

1. In the Conditions section of the decision table, in the R1 column, set the following values. Double-click each cell to add a value.

<table>
<thead>
<tr>
<th>Row</th>
<th>Value in R1 Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group. groupType</td>
<td>Standard</td>
</tr>
<tr>
<td>line. CategoryCode</td>
<td>Order</td>
</tr>
<tr>
<td>Header. value</td>
<td>True</td>
</tr>
</tbody>
</table>

The Header row might contain different details, depending on your configuration. For example, it might contain...
2. Click Save.

Specify the Action to Take

Specify the action to take when the condition is true:

1. In the decision table, scroll down to the Actions area.
2. In the R1 column, set the following values. Double-click each cell to add a value.

<table>
<thead>
<tr>
<th>Row</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value in R1 Column</td>
</tr>
<tr>
<td>fline.OrderedQty</td>
<td>&gt;=1000</td>
</tr>
</tbody>
</table>

3. Click Save.

Related Topics

- Using Visual Information Builder to Create Rules: Explained
- Creating Business Rules: Demo

Pausing Orchestration Processes
Pausing Orchestration Processes: Explained

A pause task temporarily stops an orchestration process from running so it can wait to meet a condition. When it meets the condition, the orchestration process releases the pause task, and then proceeds to the next orchestration process step. You can use a pause task to wait between tasks, or to specify when to release the pause and begin the next orchestration process step.

The following examples describe where you can use a pause task.

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
</table>
| Pause an orchestration process until an event occurs. | Assume your company provides a layaway program that allows a customer to reserve an item, and then pay for it in installments. You can use a pause task to pause the orchestration process. When the customer pays in full, the orchestration process calls a service named Release Paused Tasks, releases the pause task, and then ships the item to the customer.  
In another example, assume your company must release each sales order to shipping only if inventory is available at the fulfillment location. You can use a pause task that pauses the orchestration process until the inventory is available. When the inventory is available, the orchestration process calls Release Paused Tasks, releases the pause task, and then ships the item to the customer.  
For details, see Pausing Orchestration Processes Until an Event Occurs: Procedure. |
| Pause an orchestration process until time elapses. | Assume a customer cannot receive the goods that they order too early. You could use the requested ship date to prevent early delivery, but the warehouse might not recognize this date. You can set up a pause task that pauses the orchestration process until two days before the scheduled ship date occurs.  
In another example, assume your company sells a video game up to seven days before the release date of the game. This lead time allows your company to ship the game as soon as it is available. Your company allows seven days because it estimates that it requires this number of days to schedule and process the sales order before it ships this order. You can set up a pause task that pauses the orchestration process before the schedule step occurs, and continues this pause until the publisher releases the game. You can create a rule that releases the pause task according to an extensible flexfield that includes a value of the release date minus seven days. Order Management releases the pause task, schedules the sales order, and then resumes the orchestration process.  
For details, see Pausing Orchestration Processes Until Time Elapses: Procedure. |
| Pause an orchestration process until a dependency resolves. | Consider the following examples:  
• A customer requests to receive a shipment that includes coffee, paper cups, sugar, and creamer at the same time. You can use a pause task that pauses each order line until the orchestration process finishes scheduling each of these lines. The orchestration process periodically evaluates the sales order. It finishes scheduling all lines, and then ships them to the customer at the same time.  
• A customer must receive invoices for all order lines of a sales order at the same time, regardless of when the orchestration process ships each of these lines. You can use a pause task that pauses the orchestration process until it ships all items. It then sends the invoices for all lines to the customer at the same time.  
• A shipping system cannot handle an early request. It can only handle a request for a sales order that Order Management schedules to ship within two days. You can pause the orchestration process so it does not send a shipping request to the shipping system until two days before the ship date occurs.  
• Your company policy requires manual review and approval of any sales order where the scheduled date occurs after the requested date. You can use a pause task to pause the orchestration process until the review and approval finishes. |
Attributes You Can Set for the Pause Task

You can set the following attributes for a pause task.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reevaluateFlag</td>
<td>The pause task evaluates the condition the first time the rule runs. If the condition is True, then the pause task pauses the orchestration process. You can use one of the following values in reevaluateFlag to determine whether to evaluate the condition again:</td>
</tr>
<tr>
<td></td>
<td>• Y. Evaluate the condition every time the orchestration process completes a step. If the condition does not evaluate to true, then release the pause task. For example, assume the orchestration process is processing more than one order line in a sales order, and that you must evaluate the condition after it processes each order line. You can use reevaluateFlag to do the evaluation for each order line.</td>
</tr>
<tr>
<td></td>
<td>• N. Do not evaluate the condition every time the orchestration process completes a task. The orchestration process will remain paused until you explicitly release the pause task.</td>
</tr>
<tr>
<td>sacType</td>
<td>Specify the action to take when the If statement in a business rule evaluates to True. You can set sacType to the following values:</td>
</tr>
<tr>
<td></td>
<td>• SAC_TYPE_EVENT. Pause the pause task until an an event occurs.</td>
</tr>
<tr>
<td></td>
<td>• SAC_TYPE_TIMER. Pause the pause task until time elapses.</td>
</tr>
<tr>
<td></td>
<td>• SAC_TYPE_IMMEDIATE. Release the pause task.</td>
</tr>
</tbody>
</table>

sac is an abbreviation for start after condition

Resuming a Paused Orchestration Process

It is important that you resume a paused orchestration process.

Note: You must make sure to release a paused orchestration process. If you do not, then the orchestration process will remain in a paused state, which might result in failed sales orders, failed order fulfillment, and so on.

You can do one of the following to release the pause task that is pausing an orchestration process:

- **Release automatically.** Use the Release Pause Tasks scheduled process to release the pause task.
- **Release manually.** Navigate to the Orchestration Process details page, and then click Release Pause Task. You must use the Order Orchestration Error Recovery Manager role.

Pausing Orchestration Processes Until an Event Occurs: Procedure

You can use a pause task to temporarily stop an orchestration process from running so it can wait for an event to occur. Assume you work for a publisher who will release a new book at some point in the future, and that you need to provide your Gold customers an opportunity to order the book before your company releases it to the public. You can define an
An orchestration process that allows a Gold customer to order the book but pauses processing so it does not attempt to ship the book before the release date occurs.

You create the following rules:

1. If the customer is Gold, then pause the orchestration process.
2. If the customer is not Gold, then skip the pause task.

Summary of the Work

Pause an orchestration process until an event occurs:

1. Define the orchestration process.
2. Define the orchestration process steps.
3. Create the If statement for the first rule.
4. Create the Then statement for the first rule.
5. Create the If statement for the second rule.
6. Create the Then statement for the second rule.
7. Deploy the orchestration process. For details, see Deploying Orchestration Processes: Procedure.

This topic includes example values. You might use different values, depending on your business requirements.

For background details about:

- The properties you set in this topic for the pause task, see Pausing Orchestration Processes: Explained.
- The Business rule you create in this topic, search for Document ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then download the attachment that describes how to use business rules.
- A generic demonstration of how to create a rule, see Creating Business Rules: Demo.

Define the Orchestration Process

Define the orchestration process:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Orchestration Process Definitions.
4. On the Manage Orchestration Process Definitions page, click **Actions**, and then click **Create**.
5. On the Create Orchestration Process Definition page, set the following attributes, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name</td>
<td>Pause_for_Gold</td>
</tr>
<tr>
<td>Process Display Name</td>
<td>Pause_for_Gold</td>
</tr>
<tr>
<td>Process Class</td>
<td>Ship Order Class</td>
</tr>
<tr>
<td>Set</td>
<td>Common Set</td>
</tr>
</tbody>
</table>
Define the Orchestration Process Steps

You will define the following steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Step Name</th>
<th>Task Type</th>
<th>Task</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Schedule Goods</td>
<td>Schedule</td>
<td>Schedule</td>
<td>Create Scheduling</td>
</tr>
<tr>
<td>200</td>
<td>Create Reservation</td>
<td>Reservation</td>
<td>Reserve</td>
<td>Create Inventory Reservation</td>
</tr>
<tr>
<td>300</td>
<td>Pause</td>
<td>Pause</td>
<td>Pause</td>
<td>Pause Process</td>
</tr>
<tr>
<td>400</td>
<td>Create Shipment</td>
<td>Shipment</td>
<td>Ship</td>
<td>Create Shipping</td>
</tr>
<tr>
<td>500</td>
<td>Wait for Shipment Advice</td>
<td>Shipment</td>
<td>Ship</td>
<td>Wait for Shipment</td>
</tr>
<tr>
<td>600</td>
<td>Create Invoice</td>
<td>Invoice</td>
<td>Invoice</td>
<td>Create Billing Lines</td>
</tr>
<tr>
<td>700</td>
<td>Wait for Invoice</td>
<td>Invoice</td>
<td>Invoice</td>
<td>Wait for Billing</td>
</tr>
</tbody>
</table>

Define the orchestration process steps:

1. On the Step Definition tab, click **Actions**, and then click **Add Row**.
2. Set the following attributes, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Schedule Goods</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task</td>
<td>Schedule</td>
</tr>
<tr>
<td>Service</td>
<td>Create Scheduling</td>
</tr>
</tbody>
</table>

3. Add more steps. Repeat steps 1 and 2, using the following values.

   - Create Reservation
   - Create Shipment
   - Wait for Shipment Advice
   - Create Invoice
   - Wait for Invoice

Note the following:

   - Set the Step Type for all steps to Service.
   - Set the other attributes so they reflect the step behavior.
For example, for the Create Reservation step, set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task</td>
<td>Reserve</td>
</tr>
<tr>
<td>Service</td>
<td>Create Inventory Reservation</td>
</tr>
</tbody>
</table>

- Make sure you set the Exit Criteria attribute for each wait step. For example, set the Exit Criteria for the Wait for Shipment Advice step to Shipped.

4. Add the pause step. Click the Create Reservation step, click **Add Row**, set the following values in the new row, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Name</td>
<td>Pause</td>
</tr>
<tr>
<td>Step Type</td>
<td>Service</td>
</tr>
<tr>
<td>Task</td>
<td>Pause</td>
</tr>
<tr>
<td>Service</td>
<td>Pause Process</td>
</tr>
</tbody>
</table>

**Create the If Statement for the First Rule**

In this section you create the start-after condition, which is a condition on a Pause task in an orchestration process step that specifies when to pause processing before calling the next step, or when to resume processing. This condition determines whether the customer is Gold.
You will create the following If statement.

Start After Condition Set

Create the If statement for the first rule:

1. On the Pause step, in the Start-After Condition column, click Click for Rule.
2. In the Start After Condition dialog, click Add Rule, and then click Expand.
3. Click Left Value.
4. In the Condition Browser dialog, expand DooSeededOrchestrationRules, expand DOOFLine, click demandClassCode, and then click OK.
5. Click Right Value.
6. In the Condition Browser dialog, enter "Gold", click OK, and then click OK again. You must include the double quotation marks (").
Create the Then Statement for the First Rule

You will create the following Then statement.

Start After Condition Set

For brevity, this screen print does not include the entire value for the property. Here's the entire value:

```plaintext
( eventName:"Gold Preorders",
  reevaluateFlag:"N", sacType:DooSeededOrchestrationRules.SacResult.SAC_TYPE_EVENT )
```

where:

- `eventName` is a variable that you can use to help track the status of the pause task. The Order Management work area uses the text you enter wherever it displays status details, such as in Gantt chart in the Orchestration Plan tab of the Orchestration Process page that the user can access from the Manage Orchestration Process page.
- `reevaluateFlag` specifies not to evaluate this rule again.
- `sac` is an abbreviation for start after condition.
- `DooSeededOrchestrationRules` is a dictionary that contains predefined rules you can use in your business rule.
- `SacResult` is a rule set in the DooSeededOrchestrationRules dictionary. It contains events and variables you can use to specify how to handle the result of a start after condition.
- `EVENT` is an event in the SacResult rule set. It specifies to raise a business event.

Create the Then statement for the first rule:

1. In the Then area, click Add Action, and then click Assert New.
   You will assert a new fact into the DooSeededOrchestrationRules dictionary.
2. Click **Select a Target**, and then click `DooSeededOrchestrationRules.SacResult`.

3. Click **Edit Properties**.

4. In the Properties dialog, in the reevaluateFlag row, enter "N". You must include double quotation marks.

5. In the sacType row, click **Value**.

6. In the Condition Browser dialog, expand `DooSeededOrchestrationRules`, expand `SacResult`, click **SAC_TYPE_EVENT**, and then click **OK**.

   A pause type of EVENT causes the orchestration process to pause until an event occurs.

7. In the eventName row, in the Value attribute, enter "Gold Preorders". You must include double quotation marks.

   Note that an attribute on a scheduled process references this event, but you can also use it in a public service. You can also use other parameters to release a pause task. This event is not related to the events that the event framework in Oracle Middleware uses.

8. Make sure your properties resemble the following values.

   ![Properties Table](image)

9. Click **OK**, click **Save**, and then click **Save** again.
Create the If Statement for the Second Rule
You will create the following If statement for the second rule.

You create the following If statement:

- DooSeededOrchestrationRules.DOOFLine.demandClassCode Is Not "Gold"

where:

- DOOFLine is a function in the DooSeededOrchestrationRules dictionary.
- DOO is an abbreviation for distributed order orchestration, which is an earlier name for order orchestration.
- FLine is an abbreviation for fulfillment line.
- demandClassCode is a fulfillment line attribute. This attribute stores the customer class, such as Gold, Silver, or Bronze.
- Is Not determines whether the demandClassCode attribute contains Gold.

Create the If statement for the second rule:

1. On the Pause step, in the Start-After Condition column, click Click for Rule.
2. In the Start After Condition dialog, collapse Rule 1, if necessary, and then click Add Rule.
   Notice that the dialog displays a new rule and enters a value of Rule 2 in the name attribute. Make sure you do the rest of this procedure in this rule.
3. Click Expand next to Rule 2.
4. Click Left Value.
5. In the Condition Browser dialog, expand DooSeededOrchestrationRules, expand DOOFLine, click demandClassCode, and then click OK.
6. Click Is, and then click Isn't.
7. Click Right Value.
8. In the Condition Browser dialog, in the window, enter “Gold”, and then click OK. You must include the double quotation marks.

Create the Then Statement for the Second Rule

You will create the following Then statement for the second rule.

For brevity, this screen print does not include the entire value for the property. Here’s the entire value:

```java
(eventName:"Gold Preorders", ( reevaluateFlag:"N",
    sacType:DooSeededOrchestrationRules.SacResult.SAC_TYPE_IMMEDIATE )
```

where:

- The IMMEDIATE start after condition ends the pause task and allows the orchestration process to continue to the next step.

Create the Then statement for the second rule:

1. In the Then area, click Add Action, and then click Assert New.
2. Click Select a Target, and then click DooSeededOrchestrationRules.SacResult.
3. Click Edit Properties.
4. In the Properties dialog, in the reevaluateFlag row, enter "N". You must include double quotation marks.
5. In the sacType row, click Value.
6. In the Condition Browser dialog, expand DooSeededOrchestrationRules, expand SacResult, click SAC_TYPE_IMMEDIATE, and then click OK.
7. Make sure your properties resemble the following settings.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
<th>Consta</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventName</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reevaluateFlag</td>
<td>String</td>
<td>&quot;N&quot;</td>
<td></td>
</tr>
<tr>
<td>sacType</td>
<td>String</td>
<td>DooSeededOrchestrationRules.SacResult.$</td>
<td></td>
</tr>
<tr>
<td>viewRowImpl</td>
<td>oracle.jbo.server.ViewRow...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>waitDateTime</td>
<td>DooSeededOrchestration...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. Click OK, and then click Save.

Related Topics
• Creating Business Rules: Demo

Pausing Orchestration Processes Until Time Elapses: Procedure

You can use a pause task to temporarily stop an orchestration process from running until time elapses.

In this example, assume your supply chain analysis determined that it typically requires two days to ship goods from the warehouse to the customer when your customer orders through your legacy system, and that these customers cannot receive the sales order before the scheduled ship date. You set up a pause task that pauses the orchestration process until two days before the scheduled ship date occurs. You create the following business rule:

• If the source system is Legacy, then release the pause on ShipDate minus two.

Summary of the Work
Pause an orchestration process until time elapses:

1. Define the orchestration process.
2. Create the If statement for the first rule.
3. Create the Then statement for the first rule.
4. Create the second rule.
5. Deploy the orchestration process. For details, see Deploying Orchestration Processes: Procedure.

This topic includes example values. You might use different values, depending on your business requirements.
For background details about:

- The properties you set in this topic for the pause task, see Pausing Orchestration Processes: Explained.
- The Business rule you create in this topic, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then download the attachment that describes how to use business rules.
- A generic demonstration of how to create a rule, see Creating Business Rules: Demo.

### Define the Orchestration Process

Define the orchestration process:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Orchestration Process Definitions.
4. On the Manage Orchestration Process Definitions page, click **Actions**, and then click **Create**.
5. On the Create Orchestration Process Definition page, set the following attributes, and then click **Save**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name</td>
<td>Pause_for_Time</td>
</tr>
<tr>
<td>Process Display Name</td>
<td>Pause_for_Time</td>
</tr>
<tr>
<td>Process Class</td>
<td>Ship Order Class</td>
</tr>
<tr>
<td>Set</td>
<td>Common Set</td>
</tr>
</tbody>
</table>

6. Define the orchestration process steps.

Follow the instructions in the Define the Orchestration Process Steps section of the topic Pausing Orchestration Processes Until an Event Occurs: Procedure on Oracle Help Center at https://docs.oracle.com/cloud/latest/scmcs_gs/FAIOM/FAIOM2061951.htm#FAIOM1761954.

### Create the If Statement for the First Rule

Create the If statement that determines whether the source system is Legacy.

---

**Oracle**

351
You will create the following If statement.

Create the If statement for the first rule.

1. On the Pause step, in the Start-After Condition column, click **Click for Rule**.
2. In the Start After Condition dialog, click **Expand**, and then click **Show Advanced Settings**.
3. Add a check mark to **Advanced Mode**, and then add a check mark to **Tree Mode**.

   Tree Mode helps your rule maintain the following sales order hierarchy when the orchestration process evaluates the rule. For details about creating rules, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on
My Oracle Support at https://support.oracle.com, and then download the attachment that describes how to create business rules.

4. Set the following value.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>DooSeededOrchestrationRules.DOODHeader</td>
</tr>
</tbody>
</table>

5. In the IF area, in the Left Value attribute, delete `DooSeededOrchestrationRules.DOODHeader`, and then enter `Header`.

6. In the IF area, click the down arrow that is located next to Right Value, and then click Delete Test.

7. Click Add Pattern.

8. In the left field, under Add Test, enter `Line`.

9. Set the field that is located to the right of Is A to `DooSeededOrchestrationRules.DOOFLine`.

10. Immediately under the left field that contains the value Line, click Add Test, and then click Simple Test.

Make sure you click Add Test that is located immediately below the attribute you edited in step 7.

11. Click Left Value.

12. In the Condition Browser, expand `Header`.

Notice the list displays order header attributes you can use to specify which attribute to examine. For this example, this If statement creates a filter that causes the rule to only consider source orders where the Source Order System attribute contains a value of LEG, where LEG is an abbreviation for the term legacy.

13. Click `sourceOrderSystem`, and then click OK.

14. In the Right Value field, enter “LEG”.

Assume your deployment uses the text LEG to indicate a source order from a legacy system. You must include the double quotation marks when you use a string.
Create the Then Statement for the First Rule

You will create the following Then statement. This statement adds the pause task and specifies to release it two days before the scheduled ship date occurs.

1. In the Then area, click Add Action, and then click Assign New.
2. Click Select a Target, and then click DooSeededOrchestrationRules.Timestamp.
3. In the empty field immediately to the right of the Select a Target field, enter DateTime.
4. Do not include quotation marks. Adding DateTime in this field creates the variable that this example uses for the date calculation.
5. Click Expression Value.
6. In the Condition Browser dialog, expand Line.
7. The dialog displays a list of fulfillment line attributes. You can set up you rule so it considers the value of one these attributes.
8. Click scheduleShipDate, and then click OK.
9. Make sure the field where you added DateTime in step 3 still contains DateTime.
10. Click Add Action, and then click Modify.
11. In the Properties dialog, in the Time row, click Value.
12. In the Condition Browser dialog, expand DateTime, and then click Time.
Note that the Condition Browser adds a value of `DateTime.time` to the Expression field.

13. In the Expression field, append `-(2*24*60*60*1000)` to the end of `DateTime.time`.

This value calculates the number of milliseconds so the pause task ends two days before the orchestration process schedules the order lines for shipping, where:

- 2 is the number of days.
- 24 is the number of hours in one day.
- 60 is the number of minutes in one hour.
- 60 is the number of seconds in one hour.
- 1000 is the number of milliseconds in one second.

14. Make sure the Expression field includes the following value, and then click **OK**: `DateTime.time-(2*24*60*60*1000)`.

For example.
15. Make sure the Properties dialog includes the following value, and then click OK.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hours</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>minutes</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>month</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nanos</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seconds</td>
<td>int</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time</td>
<td>long</td>
<td>DateTime.time-(2<em>24</em>60<em>60</em>1000)</td>
<td></td>
</tr>
<tr>
<td>year</td>
<td>int</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

16. Click Add Action, and then click Assert New.
17. Click Select a Target, and then click DooSeededOrchestrationRules.SacResult.
18. Click Edit Properties.
19. In the Properties dialog, in the reevaluateFlag row, in the Value field, enter “N”. You must include the double quotation marks.
20. In the sacType row, click Value.
21. In the Condition Browser dialog, expand DooSeededOrchestrationRules, expand SacResult, click SAC_TYPE_TIMER, and then click OK.

A pause type of TIMER causes the orchestration process to pause until time elapses.

22. In the Properties dialog, in the waitDateTime row, click Value.
23. In the Condition Browser, click DateTime, click OK, and then click OK again.

Create the Second Rule
You will create the following business rule:

- If the customer is not Gold, then immediately release the pause task.
For example:

```
( reevaluateFlag:"N", sacType:DooSeededOrchestrationRules.SacResult.SAC_TYPE_IMMEDIATE )
```

where:
- `reevaluateFlag` is a property of the `SacResult` object. It specifies to evaluate the result of the start after condition again after the rule runs. In this example, you don’t want this reevaluation, so you set `reevaluateFlag` to `N`.
- `sacType` is a property of `SacResult`. It stores the value that the rule uses to determine whether to pause the orchestration process or release it.
- `DooSeededOrchestrationRules` is a predefined dictionary that contains objects you can use with Order Management.
- `SAC_TYPE_IMMEDIATE` is a value of the `sacType` property. It specifies to immediately release the pause task that is pausing the orchestration process.

Create the second rule:

1. In the Start After Condition dialog, near the top of the dialog, add a check mark in the option next to Rule 1, click **Collapse**, click **Cut**, click **Copy**, and then click **Paste**.
2. In the Name attribute, Replace Rule 1 with Rule 2.
3. Click **Expand**.
4. In the If area, next to `Header.sourceOrderSystem`, change `is` to `isn’t`.
5. In the Then area, delete the rules.
   - In the Then area, click **Add Action**, and then click **Assert New**.
   - Click **Select a Target**, and then click `DooSeededOrchestrationRules.SacResult`.
   - Click **Edit Properties**.
   - In the Properties dialog, in the reevaluateFlag row, in the Value field, enter “N”.  

You must include the double quotation marks. This example uses reevaluateFlag to specify whether or not to evaluate the rule after the calculated date occurs.

9. In the sacType row, click Value.
10. In the Condition Browser dialog, expand `DooSeededOrchestrationRules`, expand `SacResult`, click `SAC_TYPE_IMMEDIATE`, and then click OK.
11. Make sure your dialog includes the following values, and then click OK.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
<th>Consta</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventName</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reevaluateFlag</td>
<td>String</td>
<td>&quot;N&quot;</td>
<td></td>
</tr>
<tr>
<td>sacType</td>
<td>String</td>
<td><code>DooSeededOrchestrationRules.SacResult.</code></td>
<td></td>
</tr>
<tr>
<td>viewRowImpl</td>
<td></td>
<td><code>oracle.jbo.server.ViewRo...</code></td>
<td></td>
</tr>
<tr>
<td>waitDateTime</td>
<td></td>
<td><code>DooSeededOrchestration...</code></td>
<td></td>
</tr>
</tbody>
</table>

12. In the Start-After Condition dialog, click Save.

**Related Topics**
- Creating Business Rules: Demo

Pausing Orchestration Processes Until a Dependency Resolves: Procedure

This example describes how to coordinate orchestration processes to make sure that Order Management Cloud invoices all the order lines that a sales order contains at the same time, even if the shipment dates vary. The orchestration process uses the same sequence of steps to process each fulfillment line, but these lines might not be synchronized.

Assume a school district places a sales order for 600 new history books and requires that they receive the invoice for this sales order only after Order Management ships all the books.

Assume that the orchestration process must get the books from the following warehouses:

- Fulfillment line 1 gets 80 books from the Seattle warehouse
- Fulfillment line 2 gets 400 books from the Denver warehouse
- Fulfillment line 3 gets 120 books from the Chicago warehouse
You define an orchestration process that processes the order lines, pauses them until all of these lines are shipped, and then sends them to billing.

**Summary of the Work**

Pause an orchestration process until items are ready to ship:

1. Define the orchestration process.
2. Create the first rule.
3. Create the If statement for the second rule.
4. Create the Then statement for the second rule.

This topic includes example values. You might use different values, depending on your business requirements.

For background details about:

- The properties you set in this topic for the pause task, see Pausing Orchestration Processes: Explained.
- The business rules you create in this topic, including how to set sacResult, how to use Advanced Mode, and how to use Tree Mode, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then download the attachment that describes how to use business rules.
- A generic demonstration of how to create a rule, see Creating Business Rules: Demo.

**Define the Orchestration Process**

Define the orchestration process. For an example that describes how to do this work, see Pausing Orchestration Processes Until an Event Occurs: Procedure:

1. Create a new orchestration process and define the header.
2. Add the following orchestration process steps:
   - Schedule
   - Reserve
   - Ship
   - Wait for Shipment
   - Pause
   - Invoice
   - Wait for Billing
Create the First Rule
You will create the following business rule.

```
IF
  Header is a DooSeededOrchestrationRules.DOOLayer
  Add Test
THEN
  assign new DooSeededOrchestrationRules.SacResult = new DooSeededOrchestrationRules.SacResult
  assign DooSeededOrchestrationRules.HeaderSacResult = SAC
  assign DooSeededOrchestrationRules.SacResult SacType = DooSeededOrchestrationRules.SacResult
```

where:

- **DooSeededOrchestrationRules.** A dictionary that contains predefined rules you can use in your business rule.
- **DOOLayer.** An object in the DooSeededOrchestrationRules dictionary that stores details about the sales order header.
- **Assign New.** Specifies to assign DOOLayer as a new object in the DooSeededOrchestrationRules dictionary. This assignment makes the object available throughout your rule.
- **SacResult.** A rule set in the DooSeededOrchestrationRules dictionary. It contains events and variables you can use to specify how to handle the result of a start after condition.
- **SAC.** A value you can set for SacResult. sac is an abbreviation for start after condition.
- **The first Assign.** Specifies to set the value of SacResult to SAC.
- **sacType.** A property of SacResult. sacType stores the value that the rule uses to determine whether to pause the orchestration process or release it.
- **SAC_TYPE_IMMEDIATE.** A value of sacType. If sacType contains IMMEDIATE, then the rule will immediately release the pause task.
- **The second Assign.** Specifies to set sacType to SAC_TYPE_IMMEDIATE.

Create the first rule:

1. On the Pause step, in the Start-After Condition column, click Add Rule.
   In the Start After Condition dialog, click Add Rule, click Expand, click Show Advanced Settings, and then set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Default Rule for Dependent Pause</td>
</tr>
<tr>
<td>Description</td>
<td>Default rule that pauses orchestration process until a dependency resolves.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>Priority</td>
<td>Highest</td>
</tr>
<tr>
<td>Advanced Mode</td>
<td>Contains a check mark.</td>
</tr>
</tbody>
</table>

2. In the field below If, enter an alias, such as `Header`.
3. In the field to the right of Is A, select `DooSeededOrchestrationRules.DOOHeader`.
4. Click the down arrow next to Right Value, and then click Delete Test.

   If you remove the test, then the rule always applies the actions that the Then area contains.
5. In the Then area, click Add Action, and then click Assign New.
6. Click Select a Target, and then click `DooSeededOrchestrationRules.SacResult`.
7. In the field to the left of the equal sign (=), enter SAC.
8. In the field to the right of the equal sign (=), click Expression Value.
9. In the Condition Browser dialog, expand `DooSeededOrchestrationRules`, expand SacResult, click New, and then click OK.
10. Click Add Action, and then click Assign.
11. Click Select a Target, and then click Header.sacResult.
12. In the field to the right of the equal sign (=), enter SAC.
13. Click Add Action, and then click Assign.
14. Click Select a Target, and then click Header.sacResult.sacType.
15. In the field to the right of the equal sign (=), click Expression Value.
16. In the Condition Browser dialog, expand `DooSeededOrchestrationRules`, expand SacResult, click SAC_TYPE_IMMEDIATE, and then click OK.
17. Click Validate, and then click Save.
Create the If Statement for the Second Rule

You will create the following rule. It states that if the actual ship date is not null on all fulfillment lines, then release the pause task.

First, you will create the following If statement.
where:

- **There is a Case Where.** Specifies to pause all shipped lines even if the fulfillment system shipped all lines except one line.
- **OrderHeader/allFLinesInTheOrder.** Allows Order Management to coordinate across multiple orchestration processes where more than one orchestration process processes the fulfillment lines for this sales order.
- **actualShipDate is null.** Determines whether any of the fulfillment lines have not shipped. actualShipDate is null when orchestration starts. Orchestration sets actualShipDate to a value when it ships the item.

Create the If statement for the second rule:

1. On the Pause step, in the Start-After Condition column, click **Add Rule**.
   In the Start After Condition dialog, collapse the rule you just created.
2. click **Add Rule**, click **Expand**, click **Show Advanced Settings**, and then set the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Rule for Pause Dependency</td>
</tr>
<tr>
<td>Description</td>
<td>Rule that pauses orchestration process until a dependency resolves.</td>
</tr>
<tr>
<td>Priority</td>
<td>Medium</td>
</tr>
<tr>
<td>Advanced Mode</td>
<td>Contains a check mark.</td>
</tr>
<tr>
<td>Tree Mode</td>
<td>Contains a check mark.</td>
</tr>
</tbody>
</table>

3. Set Root to **DooSeededOrchestrationRules.DO0Header**.
4. In the If area, in the field to the left of Is A, replace the default value with an alias, such as **OrderHeader**.
5. Click the down arrow to the right of Right Value, and then click Delete Test.
6. Click Add Pattern to the right of the If statement.
7. To the right of the empty fields, click Surround Pattern with Parenthesis, and then click Surround.
8. In the list of values under And, click There is a Case Where.
9. In the field on the left, under There is a Case Where, enter FLine.
10. In the field to the right of Is A, select OrderHeader/allFLinesInTheOrder.
11. Click the down arrow next to Add Test, and then click Simple Test.
12. Click Left Value.
13. In the Condition Browser, expand FLine, click actualShipDate, and then click OK.
14. Click Right Value.
15. In the Condition Browser, click null, and then click OK.

Create the Then Statement for the Second Rule

You will create the following Then statement.

This step configures Order Management to evaluate the order lines until it ships all of them, and then to release them.

1. In the Then area, click the down arrow next to Add Action, and then click Assign.
2. Click Select a Target, and then click OrderHeader.sacResult.eventName.
3. Click Expression Value.
4. In the Condition Browser, expand DooSeededOrchestrationRules, expand SacResult, select SAC_SYSTEM_EVENT_IPC_PAUSE, and then click OK.

You must include the double quotation marks.

5. Click the down arrow next to Add Action, and then click Assign.
6. Click Select a Target, and then click OrderHeader.sacResult.reevaluateFlag.
7. In the field to the right of the equal sign (=), enter “Y”.
8. Click the down arrow next to Add Action, and then click Assign.
9. Click Select a Target, and then click OrderHeader.sacResult.sacType.
10. Click Expression Value on the same line.
11. In the Condition Browser, expand DooSeededOrchestrationRules, expand SacResult, click SAC_TYPE_EVENT, and then click OK.
12. Click Validate, and then click Save.
Automatically Resuming Paused Orchestration Processes: Procedure

You can use a scheduled process or a public service to release a pause task so that an orchestration process can automatically resume processing sales orders.

**Using a Scheduled Process to Release Pause Tasks**

Use a scheduled process to release a pause task:

1. In the Navigation, click **Scheduled Processes**.
2. On the Scheduled Processes page, click **Schedule New Process**.
3. In the Schedule New Process dialog, set the following value, and then click **OK**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Release Pause Tasks</td>
</tr>
</tbody>
</table>

4. Use one of the Parameters to specify the orchestration process you must resume.

   For example, if you set up a pause task on an orchestration process that is waiting for event Gold Preorders, then set the event parameter to Gold Preorders.

5. Click Submit.

For an example that uses this scheduled process, see Pausing Orchestration Processes Until an Event Occurs: Procedure.

**Using a Public Service to Release Paused Tasks**

An external application can call a public service named Release Paused Tasks to release a pause task. You can use the settings of this service to identify the pause tasks to release. The event handler calls the service to release the pause tasks according to the settings that you specify.

To get the attributes that the sales order references, you can use the order capture service that requests the order details.

**Managing Hold Codes: Procedure**

You can set up a hold code that allows your users to place a hold on a sales order. This hold pauses an action that is currently running on a business object or business service that an orchestration process references.

You can define a hold code only for a hold that originates in Order Management. You cannot use it for a hold that originates in some other application.
Summary of the Work

For this example, assume you are an order administrator for Vision Corporation, and your sales engineers have informed you that a problem might exist with an item. You must create a hold code that your users can use to temporarily hold all sales orders that reference this item while your sales engineers investigate the problem. Do the following work:

1. Create a hold code.
2. Create a release reason.
3. Test your work.

This topic includes example values. You might use different values, depending on your business requirements.

Create a Hold Code

Create a hold code:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Hold Codes.
4. On the Manage Hold Definitions page, click **Actions**, click **Add Row**, and then set the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>HOLD_PRODUCT_ISSUE</td>
</tr>
<tr>
<td>Name</td>
<td>Hold to Investigate Product</td>
</tr>
<tr>
<td>Description</td>
<td>This hold allows us to investigate a problem that occurred with the item.</td>
</tr>
<tr>
<td>Hold All Services</td>
<td>Contains a check mark.</td>
</tr>
<tr>
<td></td>
<td>This value specifies to apply a hold on all services. Task layer services determine whether or not Order Management attaches a hold code to the fulfillment line, or to the sales order for one or more fulfillment tasks that the orchestration process references.</td>
</tr>
<tr>
<td>Set</td>
<td>COMMON</td>
</tr>
<tr>
<td></td>
<td>You can use Set to specify the set that Order Management uses with a hold code. You can assign a hold code to a single business unit, or you can assign multiple hold codes to sets of business units. These sets can share hold codes.</td>
</tr>
</tbody>
</table>

5. Optional. Specify each service where Order Management must apply the hold.

For example, do the following work where Order Management must apply the hold:

- Make sure Hold All Services does not contain a check mark.
- In the Hold to Investigate Product: Services area, click **Actions**, and then click **Select and Add**.
- the Select and Add: Services dialog, leave all fields empty, and then click **Search**.
- Click a **row** that displays Shipment in the Task Type column.
- For this example, you must apply the hold to all services, so click **Cancel**, and then add a check mark to Hold All Services.
6. Click **Save and Close**.
Create a Release Reason

Create a reason that your users can choose to indicate why they released the hold:

1. On the Setup page, search for, and then open Manage Standard Lookups.
2. On the Manage Standard Lookups page, enter the following value, and then click Search.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Type</td>
<td>DOO</td>
</tr>
</tbody>
</table>

3. In the Search Results, click the row that displays DOO_HLD_RELEASE_REASON in the Lookup Type column.
4. In the DOO_HLD_RELEASE_REASON: Lookup Codes area, click Actions, click New, and then set the following values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Code</td>
<td>DOO_HLDREL_PRODUCT_OK</td>
</tr>
<tr>
<td>Display Sequence</td>
<td>1</td>
</tr>
<tr>
<td>Start Date</td>
<td>The current date.</td>
</tr>
<tr>
<td>End Date</td>
<td>One week from today.</td>
</tr>
<tr>
<td>Meaning</td>
<td>Fixed the Product Problem</td>
</tr>
<tr>
<td>Description</td>
<td>Fixed the possible problem with item.</td>
</tr>
</tbody>
</table>

5. Click Save and Close.

Test Your Work

Test your work:

1. In the Navigator, click Order Management.
2. On the Overview page, click Actions, and then click Manage Orders.
3. On the Manage Orders page, enter the following value, and then click Search.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Equals Processing</td>
</tr>
</tbody>
</table>

4. In the Search Results, click a link in the Order column.
5. On the Order page, in the Order Lines area, click Apply Hold.
6. In the Apply Hold dialog, set the following value, and then click Save and Close.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold Name</td>
<td>Hold to Investigate Product</td>
</tr>
</tbody>
</table>
7. In the Order Lines area, to verify that Order Management placed a hold on the order line, make sure it displays the following On Hold icon.

![Order Lines](image)

8. Click the **arrow** next to Apply Hold, and then click **Release Hold**.

9. In the Release Hold dialog, set the following values, and then click **Save and Close**. Note that you defined these values earlier in this topic.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold Name</td>
<td>Hold to Investigate Product</td>
</tr>
<tr>
<td>Release Reason</td>
<td>Fixed the Product Problem</td>
</tr>
</tbody>
</table>

**Related Topics**

- Using Holds to Temporarily Stop Processing: Explained

---

### Setting Up Task Types

**Task Types: Explained**

You can use a task type to determine how to process a sales order. 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

Note the following:

- Each task type is read-only. You cannot modify or delete it.
- You can create a new task type. For details, see Creating Task Types: Procedure.
- 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 or a 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 task type that you create 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 Management work area displays Ship Goods when a Shipment service runs, regardless of whether or not it calls the Create Shipment service or the Update Shipment service. The Order Management 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.

Order Management Cloud comes predefined with the following task types. For details about how Order Management uses these task types, see Task Layer: Explained.

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>Set of services that communicate with a fulfillment system to cause a human activity, such as an installation.</td>
</tr>
<tr>
<td>Fulfillment Order</td>
<td>Set of services that integrate between Order Management and an Enterprise Resource Planning (ERP) system. Note that the fulfillment task layer can run multiple fulfillment actions through a single request, such as shipment and invoice.</td>
</tr>
<tr>
<td>Invoice</td>
<td>Set of services that communicate with a billing system to create invoices for an order line.</td>
</tr>
<tr>
<td>Pause</td>
<td>Set of services that temporarily pauses processing to wait until a date or event occurs before proceeding to the next orchestration process step. You can use the pause task to coordinate processing across multiple fulfillment lines in a single sales order.</td>
</tr>
<tr>
<td>Procurement</td>
<td>Set of services that source and ship the items that a fulfillment line references when these items reside in an outside organization.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Set of services that schedules a fulfillment line.</td>
</tr>
<tr>
<td>Supply</td>
<td>Set of services that communicate with Oracle Fusion Supply Chain Orchestration so that you can use more complex logic when you source the items that a fulfillment line references.</td>
</tr>
<tr>
<td>Reservation</td>
<td>Set of services that reserve the items that a fulfillment line references.</td>
</tr>
<tr>
<td>Return</td>
<td>Set of services that communicate with a fulfillment system to return the items that a fulfillment line references.</td>
</tr>
<tr>
<td>Shipment</td>
<td>Set of services that communicate with a shipping fulfillment system to ship the items that a fulfillment line references.</td>
</tr>
</tbody>
</table>
Task Layer: Explained

The task layer that Order Management Cloud uses is a logical component that contains services that fulfill fulfillment requests. Some other Order Management component or an external component can request one of these services.

The following diagram illustrates an example of how Order Management uses the Order Orchestration task layer to fulfill a shipment request during order fulfillment.

Explanation of Callouts

1. Orchestration receives a request to call a task layer service. For example, Create Shipment Request. Orchestration then runs an orchestration process that processes shipment requests. At some point during processing, this process sends a request to the Task Layer to run a task layer service named Create Shipment Request.

2. The Task Layer runs the Create Shipment Request service. This service sends a request to an external interface layer named Create Shipment Request.

3. The external interface layer named Create Shipment Request sends a request to the Shipping fulfillment system.

4. The Shipping fulfillment system fulfills the shipment request.
Order Management typically uses the same process to fulfill other requests. For example, to complete an invoice, Orchestration runs an orchestration process that processes invoice requests, the Task Layer runs a service that processes a request to complete an invoice, and so on.

**Activity Service**

The Activity service sends an activity request to the activity fulfillment system. This system creates and fulfills the activity, and then sends replies and updates to the activity service. The activity service interprets these replies and updates.

Note the following:

- Each activity includes attributes, such as subject, activity type, earliest start date, due date, scheduled duration, actual duration, percent completion, and assignee.
- Order Management can associate an activity with one or more fulfillment lines.
- The activity service does not typically immediately fulfill an activity task, so a wait service allows the orchestration process to wait for the activity to complete.
- Order Management does not support partial fulfillment. An activity step must finish before the orchestration process can proceed to the next orchestration process step.

The Activity service provides the following features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send requests to the activities management system</td>
<td>Sends a Create Activity request to the activities management system that creates the activity. If the task layer service receives a change order, then it changes or cancels the activity, as necessary. If Order Management applies a hold on a sales order, then the activity task layer service sends a request to the activities management system to hold the activity that is currently in progress.</td>
</tr>
<tr>
<td>Receive activity status updates</td>
<td>You can schedule an orchestration process that periodically gets the updated activity status.</td>
</tr>
<tr>
<td>Modify activities</td>
<td>An activity is a predefined task type. You can create a new activity task type. You can also use an extensible flexfield to modify an activity. You can enable an activity default in a task type or orchestration process step. Defaulting the activity type allows you to categorize activities so that the activity management system can perform the business logic and do validation according to the activity type. If you do not set the default value for the task type or the orchestration process step, then Order Management sets the subject of the activity to the name of this step. Some activity management systems allow you to create an activity template that supports a human task. To use an activity template in an activity management system, you can specify the activity template for a task type or an orchestration process step that creates the activity according to the predefined template when you define the task type and the orchestration process.</td>
</tr>
</tbody>
</table>

**Reserve, Schedule, and Hold Services**

The Reserve service, Schedule service, and Hold service provides the following features.
Service Name | Description
--- | ---
Reserve Service | Sends a reservation request to an inventory fulfillment system. A reservation reserves the supply for a sales order so that no other sales order or system can use this supply.

Schedule Service | Sends a scheduling request to order promising. For example, to schedule a sales order, unschedule a sales order, or to check the availability of the item. This scheduling applies to fulfillment lines that are waiting for manual scheduling and for fulfillment lines that fail scheduling in the automated or manual process. This scheduling works only for unscheduled fulfillment lines. Order Management does not allow automatic rescheduling from the Order Management work area.

Hold Service | Sends a hold request to a fulfillment system. For example, the HoldShipmentRequest service can send a hold request from Order Management to a shipping fulfillment system. In this example, the Shipping task layer service contains the hold.

Shipment Service
The Shipment service sends a shipment request to a shipping system. Most shipping services start in orchestration. The Shipment service provides the following features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send a request to the fulfillment system that creates the shipment request</td>
<td>If Order Management receives a change order, then the shipment service changes or cancels the shipment request, if necessary. If someone applies a shipping hold, and if the Hold on Running Task feature is selected, and if the task that is running is a Shipping task, then the shipment service sends a request to the shipping fulfillment system to hold the shipment request that it is currently processing.</td>
</tr>
</tbody>
</table>

Consolidate fulfillment lines

Consolidates the fulfillment lines of:

- A shipment set, and then sends them to the shipping fulfillment system together.
- A configured item, and then sends them to the shipping fulfillment system together.

Receive fulfillment line details and status updates from the shipping system and update business objects in Order Management.

The shipping system might send fulfillment line details to Order Management at any time when a status update occurs, including before it confirms a shipment. These details might include freight cost, tracking number, way bill number, and so on.

Order Management interprets the update that it gets from the shipping system, and then uses one of the following predefined values to update the status:

- Picked
- Packed
- Shipped
- Backordered

Note the following:

- The shipment service continues to interpret the updates that it receives from the shipping system even after the shipping system ships the fulfillment lines. It sends details about these updates to Order Management.
- If the shipping system uses multiple currencies to represent cost, then the shipment service converts them before it sends the update to Order Management.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the shipping system uses a unit of measure to represent shipping</td>
<td>If the shipping system uses a unit of measure to represent shipping that is different from the unit of measure that Order Management uses in the sales order, then the fulfillment system converts this unit of measure back to the unit of measure that the sales order uses, and then communicates the shipped quantity to Order Management.</td>
</tr>
<tr>
<td>Split a fulfillment line, a shipment set, or a configured item when</td>
<td>If only part of a fulfillment line ships, then the shipment service splits this line into the following lines:</td>
</tr>
<tr>
<td>a configured item when only part of a</td>
<td>• One fulfillment line that includes the quantity that was shipped</td>
</tr>
<tr>
<td>shipment ships</td>
<td>• Another fulfillment line that includes the quantity that was not shipped</td>
</tr>
<tr>
<td></td>
<td>If only some fulfillment lines of a shipment set ship, then the shipment service removes the unshipped lines from the shipment set.</td>
</tr>
<tr>
<td></td>
<td>If half of the fulfillment lines of a configured item ships, then the task layer service splits the configured item into a shipped item and an unshipped item. If the split is not proportional, then a remnant results.</td>
</tr>
</tbody>
</table>

**Invoice Service**

The Invoice service sends requests to a billing system and interprets the replies that it receives from this system. The billing system creates the invoice and the credit transactions. Order Management does not allow you to modify a fulfillment line after the invoice service sends this request. If your deployment uses a source system, then the source system provides most of the data that the billing system requires to complete billing. Order Management stores these details, and then routes the billing request to the billing system.

> **Note:** A change order from Order Management cannot update an invoice. A change to an invoice is typically a credit from a return order or a prepayment. It is not typically a cancellation.

The Invoice service sends the following details to the billing system. The Description column describes the details that the Invoice service sends to the billing system.

<table>
<thead>
<tr>
<th>Details</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fulfillment line details from a</td>
<td>The invoice service uses the Create Billing Lines request to send billing details from the sales order or the return order each time the fulfillment line is eligible for billing. For example, it sends unit list price, unit selling price, discounts, charges, tax attributes, sales credits, and fulfillment details.</td>
</tr>
<tr>
<td>fulfillment system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note the following:</td>
</tr>
<tr>
<td></td>
<td>• The invoice service sends charges that occur in the order header with the first fulfillment line that Order Management fulfills for an order.</td>
</tr>
<tr>
<td></td>
<td>• If the fulfillment line does not include payment details or sales credits, then the invoice service sends these details from the order header.</td>
</tr>
<tr>
<td></td>
<td>• The invoice service sends prepayment details from the order header for every fulfillment line.</td>
</tr>
<tr>
<td></td>
<td>• Order Management does not support discounts that occur in the order header.</td>
</tr>
<tr>
<td>Return lines</td>
<td>The invoice service sends the reference to the original sales order line, the return reason, received quantity, and delivered quantity.</td>
</tr>
<tr>
<td>Shipment set or configured item</td>
<td>The invoice service sends the fulfillment lines that the shipment set or the configured item contains together. If only some lines in the shipment set or configured item are fulfilled, then it sends only the fulfilled lines.</td>
</tr>
</tbody>
</table>
The billing system processes the details that it receives, and then sends the following details to the Invoice service:

- Invoice details
- Credit memo details
- Billing amount
- Billing date
- Invoice date or credit memo date
- Number
- Status
- Legal entity details

It sends one of the following statuses:

- Await Billing
- Billed

**Return Service**

The Return service sends a request to a receiving system and interprets the reply and update that it receives from the receiving system. Note the following:

- The Return service creates a change receipt advice or a cancel receipt advice when Order Management receives a return request.
- The request might include one or more attribute updates, such as to increase the receipt quantity.
- If Order Management receives a request to change the original copy of the sales order that the customer returned, then the return service sends a request to the fulfillment system that creates this receipt advice.
- If Order Management receives a request to cancel the original copy of the sales order line that the customer returned, then the task layer service cancels the receipt advice. A cancel is typically allowed up until the fulfillment system receives the returned items.
- If the ordered quantity is greater than the delivered quantity on the receipt advice, and if the customer does not require the ordered quantity, then the Return service can request to cancel the remaining quantity.

**How the Return Service Handles Partial Returns**

The Return service can process a partial receipt, such as the return of only some items of a configured item.

If the customer returns only part of the return, then the Return service splits the fulfillment line into the following lines:

- One line that includes a status of Delivered for the items that the customer returned
- One line that includes the items that the customer did not return

If the customer returns only part of the return, and if the return includes a configured item or kit, then the Return service splits the fulfillment line into the following orchestration groups:

- One group that includes the fulfillment lines that the customer returned
- One group that includes the fulfillment lines that the customer did not return or that are not returnable

**How the Return Service Handles Events**
The Return service might receive a status update from the receiving system that starts an event. For example, if the receiving desk receives goods, then the Return service might receive a status update that starts an event that creates a receipt advice. The following events in the receiving system might start a status update in Order Management:

- Receive goods on the receiving deck when the receipt is created.
- Deliver goods into inventory.
- Return goods to a customer.
- Correct an order after a receipt transaction occurs. For example, a customer cannot return a deliver transaction.

### Fulfillment Order Service

The Fulfillment Order service sends a request to an enterprise resource planning (ERP) system. It provides the following features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| Send a fulfillment request to an enterprise resource planning system and receive a status update. | The Fulfillment Order service can send a request that does the following modification on a sale order that resides in Order Management, and that the enterprise resource planning system uses:  
  - Create.  
  - Update.  
  - Place a hold.  
  - Release a hold.  
  - Update the status.  
  - Cancel.  
  
  The Fulfillment Order service also does the following:  
  - Sends an update to the enterprise resource planning system each time Order Management accepts a change order that affects sales order fulfillment.  
  - Receives interim and final status updates from the enterprise resource planning system. Replies are not delivered immediately. They are delivered when the fulfillment activities run.  

Aggregate the requests that it sends to the enterprise resource planning system to help minimize problems that might occur with the timing of the requests. | An aggregator sends the requests that it has aggregated as a single request when a time limit expires or when it has aggregated a number of fulfillment lines for a sales order. If a sales order includes multiple fulfillment lines, and if these lines complete the task before the timeout occurs, then it sends all requests when the task completes. Note the following:  
  - The aggregator can aggregate only one time for each system for each sales order. If Order Management receives more fulfillment lines for a sales order after the aggregator sends a request, then it sends each of these lines individually.  
  - The default timeout is five minutes. You can create a business rule that modifies this timeout.  
  
  You can use the following parameters when you use the aggregator with an external interface routing rule. If you specify both parameters, then the fulfillment order service calls the external system when the rule meets either condition:  
  - **AGGREGATOR_MAX_FLINES.** Specify the number of pending requests that must aggregate before calling the external fulfillment system.  
  - **AGGREGATOR_WAIT_TIME.** Specify the amount of time to wait before calling the external fulfillment system. This time starts when the aggregator receives the first request. |
Related Topics

- Architectural Layers in Order Management: How They Work

Creating Task Types: Explained

The template task layer is a web service wrapper that you can use to create a new task type while maintaining data integrity in Order Management Cloud.

You can use the template task layer to do the following work:

- Define a new fulfillment task type, fulfillment task, or service that an orchestration process can call.
- Validate that the service data object includes data for each required attribute.
- Add processing logic to the outbound request or to the inbound reply for an action that comes predefined with the template task layer service. You can add logic that defaults data onto the outbound request or that validates the data. For the reply, you can add logic that interprets attributes or messages that the fulfillment system sends that might require split processing.
- Determine the transaction data to update as a result of the external service call. This data resides in the transaction tables that Order Management uses.

Maintaining data integrity makes sure the task type you create displays correctly throughout the Order Management work area. Using the template task layer also makes sure the following functionality works correctly:

- Status update
- Wait step
- Forward planning
- Jeopardy
- Hold processing
- Split processing
- Change management
- Error recovery

A new service that you create with the template task layer works in the same way that a predefined task layer service works. For details about how to implement the template task layer, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com.

Creating Task Types: Procedure

This topic describes the required and optional setup work that you must do to create a new task type.

For details about using task types, see White Papers for Order Management Cloud (Doc ID 2051639.1) on My Oracle Support at https://support.oracle.com.
Create a task type:

1. Add a task type:
   - In the Navigator, click **Setup and Maintenance**.
   - On the Setup page, click **Order Management**.
   - Search for, and then open Manage Task Types.
   - On the Manage Task Types page, click **Actions**, and then click **Add Custom**.

Order Management Cloud adds the following services in the Services list:
   - One service that references the outbound Create operation code
   - One service that references the Inbound operation code

You can specify a name for each service, and you can add a service that references some other operation code, such as Change, Get Status, Apply Hold, Release Hold, or Cancel. You must create at least one task for each new task type.

2. Assign a status code to the task type.

   Order Management sets a default value for some system status codes, such as Pending, Change Pending, Cancel, or Canceled. The status code that each task type references also controls the values for the exit criteria for a wait step that uses this task type, and the value of the task status in the next orchestration process step. You can create a new status code, or you can assign an existing status code to the new task type.

3. Connect Order Management to the fulfillment system that will perform the tasks and services that the new task type references. For details, see Connecting Order Management to Fulfillment Systems: Procedure.

4. Reference the new task type when you define the orchestration process.

   You can reference your new task type in the same way that you reference a predefined task type. Note that Order Management can use these services only in an orchestration process branch that allows split services.

For details about exit criteria, branches, wait steps, and using a task type in an orchestration process, see the Setting Up Orchestration Processes section in the guide titled Oracle SCM Cloud, Implementing Order Management.

Optional Setup for New Task Types

The following setups are optional.

<table>
<thead>
<tr>
<th>Type of Setup</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing service</td>
<td>Add preprocessing logic to the actions that come predefined with the template task layer service. For example:</td>
</tr>
<tr>
<td></td>
<td>- Default data onto the outbound request.</td>
</tr>
<tr>
<td></td>
<td>- Validate data on the outbound request.</td>
</tr>
<tr>
<td>Postprocessing service</td>
<td>Add postprocessing logic to the actions that come predefined with the template task layer service. For example:</td>
</tr>
<tr>
<td></td>
<td>- Default data onto the inbound request.</td>
</tr>
<tr>
<td></td>
<td>- Validate data on the inbound request.</td>
</tr>
</tbody>
</table>
### Type of Setup

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpret attributes or messages that the fulfillment system returns that might require Order Management to split processing across multiple orchestration processes.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To use change management on an orchestration process step that references your new task type, specify the attributes for the task type on the Manage Order Attributes That Identify Change page. Make sure your task type references the Update service and the Cancel service, and the connectors that these services require.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To apply a hold to your new service, create hold codes for it. Hold All applies to the new services and to the existing services.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To include a jeopardy score for your new task, define a jeopardy threshold for it.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define a processing constraint that controls when to use your new task. For example, you can use a processing constraint that specifies the attributes that are required in the outbound request or on the inbound reply.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The template task layer uses a complete data set to communicate Order Management attributes between the following items:</td>
</tr>
<tr>
<td>- GetValidFLineData</td>
</tr>
<tr>
<td>- The preprocessing service</td>
</tr>
<tr>
<td>- Routing rules for the external interface layer</td>
</tr>
<tr>
<td>You can reduce the data set to make processing more efficient.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If your external fulfillment system sends error messages to Order Management that you prefer to process and display in the messaging framework that Order Management uses, then you must register these messages.</td>
</tr>
</tbody>
</table>

### Related Topics

- Connecting Order Management to Fulfillment Systems: Procedure
8 Setting Up Business Rules

Using Business Rules With Order Management: Overview

You can use a business rule in Oracle Fusion Order Management Cloud to implement a dynamic decision at run time that automates a company policy, computation, or logic.

A business rule is a statement that describes how to implement a business policy or make a business decision. A business rule can implement the following logic:

- Enforce a spending policy.
- Constrain a process so it meets a regulatory requirement.
- Compute a discount or premium.
- Provide an offer according to customer value.

A business rule keeps rule logic separate from the underlying application code, which results in efficient rule maintenance and allows a business analyst to modify rule logic without using programmatic code and without interrupting your business process.

Here’s an example of business rule:

- If the sales order is valued at $50,000.00 or more, then make sure a representative calls the customer before sending an invoice.

For details about using a business rule in Order Management, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, and then download the attachment that describes how to use business rules.

Getting Data from Different Product Models When Using Oracle Business Rules: Procedure

This topic describes how to get data from the Product Information Management work area starting in Release 12.

In Release 11 you use the getItemData or getFusionAppsData methods. Starting in Release 12, you use the following objects in your business rule to populate a predefined object in the rule dictionary so it will match an entity from the Product Information Management work area:

- Item Definition
- Item Category
- Related Item

In this example, assume you must assign an orchestration process that includes some extra steps so order fulfillment safely handles hazardous material. You will define the following assignment rule:

- If shipping a hazardous material, then use the Controlled Shipping Process.
You will declare and use variables in your rule, such as var0, to maintain the following object hierarchy:

```
Order Header
  var0
     |    |
Order Line 
  var1
  |
Fulfillment Line
  var2
  |
Item Definition
  var3
```

This topic describes how to create a business rule where you must maintain the sales order hierarchy. For details, search for Doc ID 2051639.1 (White Papers for Order Management Cloud) on My Oracle Support at https://support.oracle.com, download the attachment that describes how to use business rules, and then see the chapter that describes how to maintain the object hierarchy when using only Advanced Mode.

You will create the following rule.
Get data from different product models when using Oracle Business Rules:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage Orchestration Process Assignment Rules.
4. Create a new rule using the following values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Get_Hazardous_Details</td>
</tr>
<tr>
<td>Effective Date</td>
<td>Always</td>
</tr>
</tbody>
</table>
Attribute | Value  
---|---  
Active | Contains a check mark  
Advanced Mode | Contains a check mark  

5. Declare the var0 variable:

where:

- **var0**. Declares a variable. You can enter any text. You then reference this variable elsewhere in the rule.
- **Order Header**. Specifies to store order header attributes in var0.

6. Declare the var1 variable immediately below the var0 variable:

where:

- **var1**. Declares a variable.
- **Order Line**. Specifies to store order line attributes in var1.
- var0.Order Line. Maintains a hierarchical relationship between the order header and the order line, where var0 represents the order header. It specifies to examine the order line.
- Isn't Null. Is a test that makes sure the order line contains a value. It is recommended that you make sure your data contains a value before proceeding so that you can avoid a null pointer error.
- RL.contains. Is a test that makes sure your rules language (RL) correctly declared a variable named var1. It is recommended that you perform this test for your variables as a way to make sure the declaration is correct before proceeding. If you do not do this test, if and if the declaration is not correct, then the rule might fail in a subsequent step.

7. Declare the var2 variable immediately below the var1 variable:

where:
- var2. Declares a variable.
- Order Fulfill Line. Specifies to store fulfillment line attributes in var2.
- var1.Order Fulfill Line. Maintains a hierarchical relationship between the order line and the fulfillment line, where var1 represents the order line. It specifies to examine the fulfillment line.
8. Declare the var3 variable immediately below the var2 variable:

where:

- **var3.** Declares a variable. It will store the item definition.
- **Item Definition.** Specifies to get the definition of the item from the Product Information Management work area. This definition includes the item attributes.
- **var2.Item Definition.** Maintains a hierarchical relationship between the fulfillment line and the item definition, where var2 represents the fulfillment line. It specifies to examine the item definition.
- **VIBStringToLowerCase(var3.Hazardous Material).** Specifies to convert the value of the Hazardous Material attribute to lower case, and then store it in var3.
- **VIBStringToLowerCase("Y").** Specifies to examine the lower case value of the Hazardous Material attribute to determine whether it contains the value Y.

9. Specify the Then statement:

where:

- **Call.** Calls the method named evb_isSetToADFbc_Modify.
○ \( ( \text{"processName"}, \text{"CustomDOO_ControlledShippingProcess"}, \text{var2.ViewRowImpl} ) \). Identifies the orchestration process to run, which is named CustomDOO_ControlledShippingProcess.

○ \( \text{var2.ViewRowImpl} \). Specifies to get the value of \text{ViewRowImpl} from var2, which is the variable you declared earlier to store fulfillment line attributes. The orchestration process uses the value of \text{ViewRowImpl} to identify the fulfillment line it must process.

**Related Topics**

- Assigning Orchestration Processes: Explained
- Creating Business Rules: Demo

---

### Using Visual Information Builder to Create Business Rules

### Using Visual Information Builder to Create Rules: Explained

Prior to Release 11, you use Oracle Business Rules to create rules in Order Management Cloud. Starting with release 12, you can use Visual Information Builder, which is a rule editor that supports a simplified drag-and-drop interface. It helps you to visualize data, visualize your business processes, implement your business logic, and implement your business rule sets.

You can use either editor. To access them, use the following pages.

<table>
<thead>
<tr>
<th>Editor for Oracle Business Rules</th>
<th>Editor for Visual Information Builder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage External Interface Routing Rules</td>
<td>Manage External Integration Routing Rules for Sales Orders</td>
</tr>
<tr>
<td>Manage Pretransformation Defaulting Rules</td>
<td>Manage Pretransformation Rules for Sales Orders</td>
</tr>
<tr>
<td>Manage Orchestration Process Assignment Rules</td>
<td>Manage Process Assignment Rules for Sales Orders</td>
</tr>
</tbody>
</table>

Note the following limitations when using Visual Information Builder:

- You cannot use a decision table. If you created a rule in an earlier version of Order Management that uses a decision table, and if you prefer to use Visual Information Builder to replace this rule, then you can use the IF-THEN-ELSE statement in Visual Information Builder to replace the decision table logic. For an example that uses the IF-THEN-ELSE statement instead of a decision table, see Using Visual Information Builder to Manage Process Assignment Rules: Procedure.

- It is recommended that you use Visual Information Builder or Oracle Business Rules in your implementation. It is recommended that you do not use Visual Information Builder and Oracle Business Rules at the same time in the same implementation.

- You cannot define a global variable for use in a rule.

- You cannot define collections of objects in conditions.
• Do not reference an object that includes the text _Obsolete when you create your rule. _Obsolete indicates that business rules might no longer support the object in the current or future release.

Examine the Behavior of a Predefined Rule

In this section, you observe how the rule editor establishes the following condition:

• If task type is shipment, then set the connector to Connect to Oracle Fusion Shipping System

Examine the behavior of a predefined rule:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup page, click Order Management.
3. Search for, and then open Manage External Integration Routing Rules for Sales Orders.
   
   You can use this page to do the following work:
   
   - Activate or inactivate a rule.
   - Create a new rule.
   - Change the rule priority.
   - View, modify, duplicate, or delete a rule.
   - Publish active rules.

5. Right-click Shipping Task Routing, click Actions, and then click Edit.
6. Click IF, and then notice the phrase Task Type is Equal to Shipment.
7. Click DO, and then notice the phrase Set Connector Name, Connector Name: Connect to Oracle Fusion Shipping System.
8. Click Close.


You can use Visual Information Builder to create routing rules.

Assume you must create a routing rule that implements the following routing rule:

• If the task type is Compliance Check, and if the compliance value is less than 100, then set the connector to ComplianceCheckConnector, and set the interface type to SDO
Summary of the Work

In this example, you create the following rule:

1. Create the routing rule and the If condition.
2. Create the Then statement.
3. Activate and publish your rule.

This topic includes example values. You might use different values, depending on your business requirements.

Create the Routing Rule and IF Condition

Create the routing rule and If condition:

1. In the Navigator, click **Setup and Maintenance**.
2. On the Setup page, click **Order Management**.
3. Search for, and then open Manage External Integration Routing Rules for Sales Orders.
4. On the Manage External Interface Routing Rules page, click **Create New Rule**.
5. Change the rule name.

<table>
<thead>
<tr>
<th>Current Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Rule</td>
<td>Routing Rule for Compliance Check</td>
</tr>
</tbody>
</table>

6. Name the If statement so that it reflects the meaning of the condition:
   - Click **Enter Description**.
   - In the Enter Description dialog, enter **Task Type = Compliance Check and Compliance Value Is Less Than 100**, and then click **OK**.
7. Click **New Condition** (the dashed circle in the IF area).
8. In the Create Condition dialog, enter Task, wait a moment, and then click Task Type (Order Header).

Notice that the dialog displays suggestions when you wait a moment after you finish typing.

9. In the Create Condition dialog, click Search.

10. In the Search dialog, enter Compliance, and then click Search.

11. Choose Compliance Check, click OK, and then click OK again.

Notice that the IF area now displays the following condition:

   ◦ Task Type is Equal to Compliance Check

12. Click AND.

13. In the Create Condition dialog, enter Compli, and then wait a moment.

Notice that the dialog displays a number of attributes, such as Compliance Date (Header Compliance Details).

Extensible flexfields that you define on the order header provide the values for these attributes. The order header displays them as typical attributes in the Order Management work area.

14. Click Compliance Value (Header Compliance Details).

15. Change the operator from = (equal), to < (less than).

16. Change the value from 0.0, to 100, and then click OK.

Notice that the IF area now displays the following condition:

   ◦ Task Type is equal to Compliance Check AND Compliance Value < 100

Create the THEN Statement

Create the Then statement:

1. Click THEN, and then click DO.

2. Name the DO so that it reflects the meaning of the action:

   ◦ Click Enter Description.

   ◦ In the Enter Description dialog, enter Set connector to ComplianceCheckConnector and interface type to SDO, and then click OK.

3. Click New Action (the dashed circle in the DO area), and then click Perform an Action.

4. In the Create Action dialog, set the action to Set Connector Name, and then click Search.

5. In the Search dialog, enter %Compli%, and then click Search.

The percentage symbols (%) are wildcards that search for all values before and after the text Compli. This search returns the services that are defined on the Web Services Detail page that contain the text Compli.

6. In the Description area, click Compliance check system conn. . ., and then click OK.

7. On the Create Action dialog, click OK.

8. In the DO area, click AND, and then click Perform an Action.

9. In the Create Action dialog, set the action to Set Interaction Interface Type.

10. Set Interaction Interface Type to SDO.

If you are a new customer, then you must set Interaction Interface Type to SDO for any interaction interface that you define. Use EBM only for backward compatibility to a prior release.

11. Click Save and Close.
Activate and Publish Your Rule

Activate and publish your rule:

1. On the Manage External Interface Routing Rules page, notice that the Active indicator for Routing Rule for Compliance Check is grey.
2. Right-click **Routing Rule for Compliance Check**, click **Actions**, and then click **Edit**.
3. In the Routing Rule for Compliance Check dialog, add a check mark to the **Activate Rule** option, and then click **Save and Close**.
   
   Notice that the Active indicator for Routing Rule for Compliance Check is green.
4. Click **Publish**.

Order Management publishes each of the active rules that display on the Manage External Interface Routing Rules page.

Related Topics

- Routing Requests to Fulfillment Systems: Explained


You can use Visual Information Builder to manage a pretransformation rule that sets the default value of an attribute according to the value of another attribute.

In the example in this topic, you create the following rule:

- If the order type is equal to Government, then set the warehouse to Denver Manufacturing
Summary of the Work
In this example, you create the following business rule:

![Diagram of IF statement and Then action]

Use Visual Information Builder to create a pretransformation rule:

1. Create the routing rule and the If condition.
2. Create the Then statement.
3. Activate and publish your rule.

This topic includes example values. You might use different values, depending on your business requirements.

Create the Routing Rule and IF Condition
Create the routing rule and If condition:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup page, click Order Management.
3. Search for, and then open Manage Pretransformation Rules for Sales Orders.
5. Change the rule name.

<table>
<thead>
<tr>
<th>Current Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Rule</td>
<td>Pretransformation Default Rule for Warehouse</td>
</tr>
</tbody>
</table>

6. Name the If statement so that it reflects the meaning of the condition:
   - Click Enter Description.
   - In the Enter Description dialog, enter Order Type = Government, and then click OK.
7. Click New Condition (the dashed circle in the IF area).
8. In the Create Condition dialog, enter order type, wait a moment, and then click Order Type (Order Header).
   Notice that the dialog displays suggestions when you wait a moment after you finish typing.
9. Accept **is equal to** as the condition, set the order type to Government, and then click **OK**.

   Notice that the IF area now displays the following condition:

   - **Order Type is equal to Government**

Create the THEN Statement

Create the Then statement:

1. Click **THEN**, and then click **DO**.
2. Name the DO so that it reflects the meaning of the action:
   - Click **Enter Description**.
   - In the Enter Description dialog, enter **Set warehouse to Denver**, and then click **OK**.
3. Click **New Action** (the dashed circle in the DO area).
4. In the Create Action dialog, enter **ware**, wait a moment, and then choose **Warehouse (Order Fulfill Line)**.
5. In the Create Action dialog, click **Search**.
6. In the Search dialog, enter **Denver**, and then click **Search**.
7. Under Warehouse Name, click **Denver Manufacturing**, and then click **OK**.
8. On the Create Action dialog, click **OK**.

   Notice that the DO area now displays the following condition:

   - **Warehouse is set to Denver Manufacturing**

9. Click **Save and Close**.

Activate and Publish Your Rule

Activate and publish your rule:

1. On the Manage External Interface Routing Rules page, notice that the Active indicator for Pretransformation Default Rule for Warehouse is grey.
2. Right-click **Pretransformation Default Rule for Warehouse**, click **Actions**, and then click **Edit**.
3. In the Pretransformation Default Rule for Warehouse dialog, add a check mark to the **Activate Rule** option, and then click **Save and Close**.

   Notice that the Active indicator for Pretransformation Default Rule for Warehouse is green.
4. Click **Publish**.

   Order Management publishes each of the active rules that display on the Manage Pretransformation Defaulting Rules page.

Related Topics

- Transformation Rules: Explained
- Transforming Source Orders to Sales Orders: How It Works
- Transforming Order Lines to Fulfillment Lines: How it Works

You can use Visual Information Builder to assign an orchestration process according to various conditions. For example, you can assign different orchestration processes to handle source orders that Order Management Cloud receives from different source systems, assign them differently depending on whether the order is a new sales order or a return, or assign them depending on the value of an attribute, such as the line type is a credit.

Summary of the Work

In this example, you create the following rule:

1. Create the routing rule and the first If condition.
2. Create the Then for the first If.
3. Create the second If.
4. Create the Then for the Second If.
5. Create the third If.
6. Create the Then for the Third If.
7. Create the Then for the Third If.
8. Create the Then for the Third If.
9. Create the Do.

Use Visual Information Builder to manage process assignment rules:

1. Create the routing rule and the first If condition.
2. Create the Then for the first If.
3. Create the second If.
4. Create the Then for the Second If.
5. Create the third If.
6. Create the Then for the Third If.
7. Create the Do that assigns the orchestration process when receiving returns for credit.
8. Create the Do that assigns the orchestration process for all other returns.
9. Create the Do that assigns the orchestration process for all other sales orders.
10. Activate and publish your rule.

This topic includes example values that demonstrate how you can use Visual Information Builder to create a complex process assignment rule. You might use different values, depending on your business requirements.

Part One Work Instructions
This section includes the detailed steps to perform for the first part of this example.

Create the Routing Rule and First IF Condition
Create the routing rule and first If condition:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup page, click Order Management.
3. Search for, and then open Manage Process Assignment Rules for Sales Orders.
5. Change the rule name.

<table>
<thead>
<tr>
<th>Current Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Rule</td>
<td>Default Process Assignment Rule</td>
</tr>
</tbody>
</table>

6. Name the if statement so that it reflects the meaning of the condition:
   - Click Enter Description.
   - In the Enter Description dialog, enter source system = BM-CPQ, and then click OK.

   This IF statement looks for source orders that are coming in from one channel in an integration, named BM-CPQ (Big Machines - Configure, Price, and Quote).

7. In the Attributes tab, in the Search field, enter Source order, and then wait a moment.

   Notice that the dialog displays suggestions when you wait a moment after you finish typing.

8. Click Source Order System (Order Header).

   Notice that the object hierarchy in the Attributes tab highlights the attribute that the search results returned, such as Source Order System.

9. Drag the Source Order System attribute from the Attributes tab, and then drop it into the IF area.
10. In the Create Condition dialog that displays after you release the mouse button, accept is equal to as the condition, and then click the down arrow in the list that displays immediately below is equal to.

   Notice that the list displays the source systems that are configured for this deployment.

11. Set the source system to BM_CPQ, and then click OK.

   Notice that the IF area now displays the following condition:

   - Source Order System is equal to BM_CPQ

Create the THEN for the First IF
Create the Then statement for the first If:

1. Click THEN, and then click DO.
2. Name the DO so that it reflects the meaning of the action:
   - Click Enter Description.
   - In the Enter Description dialog, enter Set process name to DOO_Billonly, and then click OK.
3. Click New Action (the dashed circle in the DO area), and then click Perform an Action.
4. In the Create Action dialog, enter process name, wait a moment, click Process Name (Order Fulfill Line), and then click Search.
5. In the Search dialog, enter DOO_Bill, and then click Search.
6. In the ProcessName area, click DOO_BillOnlyGenericProcess, and then click OK.
7. In the Create Action dialog, click OK.
   - Process Name is set to DOO_BillOnlyGenericProcess
8. Click Save.

Create the Second IF

Create the second If statement:

1. Below the IF area, click ELSE, and then click IF.
2. Name the If statement so that it reflects the meaning of the condition:
   - Click Enter Description.
   - In the Enter Description dialog, enter category = ORDER, and then click OK.
3. In the Attributes tab, notice that the object hierarchy displays as a navigation tree.
4. In the Attributes tab, browse through the object hierarchy:
   - Click Attributes, and then click Order Header.
   - In the Order Header branch, click Order Line.
   - In the Order Line branch, click Order Fulfill Line.
   - In the Order Fulfill Line branch, click Category.
   - Drag the Category attribute from the Attributes tab, and then drop it into the IF area.
5. In the Create Condition dialog that displays, notice that the condition defaults to Category is equal to Order, and then click OK.

Create the THEN for the Second IF

Create the Then statement for the second If:

1. In the area to the right of the IF category = ORDER condition, Click THEN, and then click DO.
2. Name the DO so that it reflects the meaning of the action:
   - Click Enter Description.
   - In the Enter Description dialog, enter Set process name to DOO_GenericFulfillmentProcess, and then click OK.
3. Click New Action (the dashed circle in the DO area), and then click Perform an Action.
4. In the Create Action dialog, enter process name, wait a moment, click Process Name (Order Fulfill Line), and then click Search.
5. In the Search dialog, enter `DOO_`, and then click **Search**.
6. In the ProcessName area, click **DOO_OrderFulfillmentGenericProcess**, and then click **OK**.
7. In the Create Action dialog, click **OK**.
   Notice that the DO area displays the following value:
   - Process Name is set to DOO_OrderFulfillmentGenericProcess
8. Click **Save**.

**Create the Third IF**

Assign the orchestration process to use when handling a return.

Create the third If statement:

1. Below the **IF category = ORDER** area, click **ELSE**, and then click **IF**.
2. Name the If statement so that it reflects the meaning of the condition:
   - Click **Enter Description**.
   - In the Enter Description dialog, enter `category = RETURN`, and then click **OK**.
3. Click **New Condition** (the dashed circle in the IF area).
4. In the Create Condition dialog, enter `category`, wait a moment, and then click **Category (Order Fulfill Line)**.
5. Set the condition to **is equal to Return goods**, and then click **OK**.
   Notice that the IF area now displays the following condition:
   - Category is equal to Return goods

**Create the THEN for the Third IF**

Returns for different types of fulfillment lines might require Order Management to use different fulfillment processing. In this example, you configure the rule to consider only returns for credit. You create the following statement:

- If Line Type is not empty, and if Line type is equal to Return for credit

Create the THEN statement for the third IF:

1. In the area to the right of the **IF category = RETURN** condition, click **THEN**, and then click **IF**.
2. Name the IF so that it reflects the meaning of the action:
   - Click **Enter Description**.
   - In the Enter Description dialog, enter `line type = credit only`, and then click **OK**.
3. Click **New Condition** (the dashed circle in the IF area).
4. In the Create Condition dialog, enter `line type`, wait a moment, and then click **Line Type (Order Fulfill Line)**.
5. Configure the rule to handle an empty line type value:
   - Set the operator to **is blank**, and then click **OK**.
   - It is important to include a condition that can handle a situation where the value of the line type is empty. If you do not create this condition, then the rule might fail and cause an error.
   - Notice that the IF area displays the following value:
   - Line Type is blank
   - In the upper right corner of the **Line Type is blank** area, click **Select** (the circle with the solid line), and then wait for it to display a blue check mark.
In the upper left corner of the page, next to the trash can, click **Group**.

Notice that the **IF line type = credit only** area now displays a **this condition is true** area around the **Line Type is blank** area. This grouping is equivalent to adding parenthesis inside your statement. In this example, you group **Line Type is blank** to test whether this statement is true or false. To group a set of statements, you select each statement, and then click **Group**.

- Click **true**.

Notice that **true** changes to **false**. In this example, when you run the rule, you must make sure the condition **line type is blank** is false because you want Order Management to assign this orchestration process only if the line type contains a value.

6. In the **IF line type = credit only** area, click the **AND** that resides outside of the **this condition is false** area, as indicated by the following arrow:

   ![Diagram](image)

7. In the Create Condition dialog, enter **line type**, wait a moment, and then click **Line Type (Order Fulfill Line)**.

8. Set the condition to **is equal to Return for credit**, and then click **OK**.

   Notice that the IF area now displays the following condition:

   - **Line type is equal to Return for credit**

9. Click **Save**.

**Part Two Work Instructions**

This section includes the detailed steps to perform for the second part of this example.

**Create the DO That Assigns the Orchestration Process When Receiving Returns for Credit**

Create the Do statement that assigns the orchestration process to use when receiving a return for credit:

1. In the area to the right of the **IF line type = credit only** condition, click **THEN**, and then click **DO**.

2. Name the DO so that it reflects the meaning of the action:
   - Click **Enter Description**.
3. In the Enter Description dialog, enter *Set process to BillOnly*, and then click OK.

4. Click New Action (the dashed circle in the DO area).

5. In the Create Action dialog, enter process name, wait a moment, click Process Name (Order Fulfill Line), and then click Search.

6. In the Search dialog, enter %Bill%, and then click Search.

   The percentage symbols (%) are wildcards that search for all values before and after the text Bill. This search returns the orchestration process names that contain the text Bill.

7. In the ProcessName area, click DOO_BillOnlyGenericProcess, and then click OK.

8. In the Create Action dialog, click OK.

   Notice that the DO area displays the following value:

   • Process Name is set to DOO_BillOnlyGenericProcess

Create the DO That Assigns the Orchestration Process for All Other Returns

Create the Do statement that assigns the orchestration process for all other returns:

1. Immediately below the IF line type = credit only condition, click ELSE, and then click DO.

2. Name the DO so that it reflects the meaning of the action:

   • Click Enter Description.

   • In the Enter Description dialog, enter Set process to Return Process, and then click OK.

3. Click New Action (the dashed circle in the DO area).

4. In the Create Action dialog, enter process name, wait a moment, click Process Name (Order Fulfill Line), and then click Search.

5. In the Search dialog, enter %Return%, and then click Search.

6. In the ProcessName area, click ReturnOrderGenericProcess, and then click OK.

7. In the Create Action dialog, click OK.

   Notice that the DO area displays the following value:

   • Process Name is set to ReturnOrderGenericProcess

8. Click Save.

Create the DO That Assigns the Orchestration Process for All Other Sales Orders

In this section, you assign the orchestration process to use for the sales orders that do not meet the other IF conditions.

Create the Do that assigns the orchestration process for all other sales orders:

1. Immediately below the IF category = RETURN condition, click ELSE, and then click DO.

2. Name the DO so that it reflects the meaning of the action:

   • Click Enter Description.

   • In the Enter Description dialog, enter Set process to ShipOrder Generic, and then click OK.

3. Click New Action (the dashed circle in the DO area).

4. In the Create Action dialog, enter process name, wait a moment, click Process Name (Order Fulfill Line), and then click Search.

5. In the Search dialog, enter Ship, and then click Search.

6. In the ProcessName area, click ShipOrderGenericProcess, and then click OK.
7. In the Create Action dialog, click OK.

   Notice that the DO area displays the following value:
   - Process Name is set to ShipOrderGenericProcess

8. Click Save.

Activate and Publish Your Rule

Activate and publish your rule:

1. On the Manage Orchestration Process Assignment Rules page, notice that the Active indicator for Default Process Assignment Rule is grey.

2. Right-click Default Process Assignment Rule, click Actions, and then click Edit.

3. In the Default Process Assignment Rule dialog, add a check mark to the Activate Rule option, and then click Save and Close.

   Notice that the Active indicator for Default Process Assignment Rule is green.

4. Click Publish.

   Order Management publishes each of the active rules that display on the Manage Orchestration Process Assignment Rules page.

Related Topics

- Assigning Orchestration Processes: Explained
- Assigning Orchestration Processes: Examples
Performing Other Implementation Work

Administering Messaging and Social Networking

Administering Email

Administering Email Notification in Order Management: Procedure

You can configure Order Management Cloud to send an email message when a business event occurs, such as when the sales order status changes to Shipped.

The following example illustrates an email format that your deployment can send when the sales order status changes to Shipped. You can modify this format, such as changing the From address, adding your company name and logo, or changing the template message text. For details, see Administering Email Formats in Order Management: Explained.

Administer email notification:

1. In the Navigator, click Setup and Maintenance.
2. On the Setup page, click Order Management.
3. Search for, and then open Manage Business Event Trigger Points.
4. Optional. Send an email notification when the sales order goes into hold:
   - On the Manage Business Event Trigger Points page, click the Hold row, and then make sure the Active option in this row contains a check mark.
   - In the Details area, add a check mark to the Send Email Notification option.
5. Optional. Send an email notification when the order header status updates:
   - On the Manage Business Event Trigger Points page, click the **Order Header Status Update** row, and then make sure the **Active** option in this row contains a check mark.
   - In the Details area, add a check mark to the **Raise Event** option and the **Send Email Notification** option for each Status Value, as necessary. For example, to configure Order Management to send an email notification when the order status changes to Closed, add a check mark to the options in the Closed row.

   The Send Email Notification option depends on the event, so you must make sure you add a check mark to each option.
   - Repeat this step for other status values, as necessary.

Administer Email Notification for Status Updates on Fulfillment Lines

Administering Order Management to send an email notification when the status updates on a fulfillment line requires that you modify the orchestration process definition.

Administer email notification for status updates on fulfillment lines:

1. On the Manage Business Event Trigger Points page, click the **Fulfillment Line Status Update** row, and then make sure the **Active** option in this row contains a check mark.
2. Click **Save and Close**.
3. On the Setup page, search for, and then open Manage Orchestration Process Definitions.
   - Each orchestration process controls the status value for each fulfillment line, so you must modify the orchestration process that controls the status value. In this example, you modify the orchestration process that controls the shipping status value.
5. In the Search Results, click the row that contains ShipOrderGenericProcess, click **Actions**, and then click **Edit**.
6. In the Process Details area, click **Status Conditions**, click **Fulfillment Line Status Values**, and then click **Edit Status Rule Set**.
7. On the Edit Status Rule Set page, add a check mark to the **Notify External System** option and the **Send Email Notification** option for each Status Value where you must send a notification.
   - For example, to send an email notification when the fulfillment line status changes to Shipped, add a check mark to the options in the Shipped row.
   - The Send Email Notification option depends on the Notify External Systems event, so you must make sure you add a check mark to each option.
8. Repeat step 7 for other status value, as necessary.

Order management will send an email notification when each fulfillment line that references this orchestration process definition reaches the status that you specify in steps 7 and 8. Order Management sends this email to the customer contact and to the ship-to contact that the sales order references.

9. Repeat steps 4 through 8 for each orchestration process definition in your deployment that updates status values.

**Administering Email Formats in Order Management: Explained**

The Order Entry Specialist can click Actions, Create Document, and then Send Email from the sales order header to send order details through an email message. Order Management Cloud sends order details in a PDF (Portable Document Format) file that it attaches to this email.
Order Management comes predefined to send an email message that resembles the following format. You can use Oracle BI Publisher to modify this format, such as adding your company name and logo, or changing the template message text.

You can use the order management parameter named From Address for Email Messages to specify the value that Order Management displays in the email From address. For details, see Managing Order Management Parameters: Procedure.

Related Topics

- Managing Order Management Parameters: Procedure
- Using Oracle BI Publisher Enterprise to Modify Templates for Use with Formats: Explained

Administering Collaboration Messaging

Using Collaboration Messaging: Overview

Use Oracle Fusion Collaboration Messaging Framework to enable Oracle Applications Cloud establish business-to-business (B2B) messaging capabilities with trading partners.

Using this framework, you can send and receive real-time transactional messages without building new SOA components. You can use the existing B2B functionality to exchange messages with collaborators such as suppliers either directly or using an intermediary agency such as a B2B Service Provider.

The framework supports transformation of a B2B document, such as a purchase order, between the Oracle Applications Cloud format and an external message format supported by the trading partner. When you send messages to partners or receive messages from them, the framework performs the required transformation.
The following figure illustrates how Collaboration Messaging Framework delivers a message to the intended recipient.

Using collaboration messaging involves performing the following high-level tasks:

- Setting up external (B2B) trading partners and their messaging capabilities.
- Cross-referencing the Oracle Applications Cloud definition of a trading partner (such as a supplier) with the external trading partner definition setup earlier. Also, selecting the messages that must be enabled with the partner.
- Configuring the message delivery method for the partner.

To open the Collaboration Messaging Framework Overview page, click the Navigator menu and select **Collaboration Messaging**.
Use the Collaboration Messaging Framework Overview page to:

- Manage undelivered collaboration messages
- Reprocess failed collaboration messages
- Manage collaboration messaging history
- Validate inbound and outbound collaboration messaging setup

**Reprocessing Undelivered Messages: Procedure**

Any inbound or outbound message that isn’t processed because of some error, remains undelivered. You can view the undelivered messages on the Collaboration Messaging Framework Overview page. For each undelivered message, you can diagnose the errors, take corrective action, and resubmit a request to deliver it again.

To reprocess an undelivered message:

1. On the Collaboration Messaging Framework Overview page, click the tasks icon to view the tasks, and select the **Manage Undelivered Collaboration Messages** task.
2. On the Manage Undelivered Collaboration Messages page, search for the undelivered message. The message is listed in a table.
3. Click the message row to view the reason for delivery failure. The details appear under Processing History.
4. **Tip:** Click the message ID link to view the setup details of the message.
5. Take the required corrective action and return to the **Collaboration Messaging Framework Overview** page.
6. Select the specific message and click **Reprocess**. If there are no further problems, the message is submitted for delivery.

**Process Inbound Collaboration Messages**

**Validating Inbound Collaboration Messaging: Procedure**

After you set up an application partner, such as a supplier site, you can send a test inbound message to verify if the setup is appropriate for messaging. Use the Validate Inbound Collaboration Messaging Setup task on the Collaboration Messaging Overview page to validate an inbound message.

All messages that go through the validation process queue up and appear on the Collaboration Messaging History page. There you can examine the details of each processed message to check if it was transformed and processed as intended.

1. On the Overview page, click the tasks icon to view the tasks, and select the **Validate Inbound Collaboration Messaging Setup** task.
2. Select the service provider and the partner ID from whom you expect to receive the collaboration message.
3. Enter the external message ID.
4. Select the details pertaining to the External Message Type. The related details, such as document type and messaging standard, appear.
5. **Click Create Message Payload.** The message payload is generated in XML format and appears in the text box.
6. Update the generated payload or replace it with the XML payload that you want to test.
7. Examine the elements of the message payload. The following table contains an example of the mapping between the elements and actual data.
### XML Element | Corresponding Information
--- | ---
<Sender> | The ID of the partner who sent the document.
<Intermediary> | Contains the ID of the service provider.
<Receiver> | Contains the ID of the recipient.
<BODID> | An ID that the sender assigns to the message.

8. Click **Process**. The Processing Confirmation message appears.
9. Click **View Collaboration Message** to view the processed message.
10. Click **Done**.

To view the processed message again, search for it on the Manage Collaboration Messaging History page. In the search results, click the generated message ID to view its details.

**Tip:** If the message processing fails, you can view the reason for it on the Manage Failed Collaboration Messages page.

## Process Outbound Collaboration Messages

### Validating Outbound Collaboration Messaging: Procedure

After you set up an application partner, such as a supplier site, you can send a test outbound message to verify if the setup is appropriate for messaging. Use the **Validate Outbound Collaboration Messaging Setup** task on the Collaboration Messaging Framework Overview page to validate an outbound collaboration message.

All messages that go through the validation process queue up and appear on the Collaboration Messaging History page. There you can examine the details of each processed message to check if it was transformed and processed as intended.

1. On the Collaboration Messaging Framework Overview page, click the tasks icon to view the tasks, and select the **Validate Outbound Collaboration Messaging Setup** task.
3. Select the supplier. The related details, such as the supplier site and service provider appear.
4. Click **Create Message Payload**. The message payload is generated in XML format and appears in the text box.
5. Update the generated payload or replace it with the XML payload that you want to test.
6. Click **Process**. The generated message ID appears on the page.
7. Click **View Collaboration Message** to view the processed message.
8. Click **Done**.

To view the processed message again, search for it on the **Manage Collaboration Messaging History** page. In the search results, click the message ID to view its details.

**Tip:** If the message processing fails, you can view the reason for it on the Manage Failed Collaboration Messages page.
Manage Collaboration Messages

Creating a Trading Partner with a Service Provider: Procedure

A service provider is an intermediary for exchanging messages between Oracle Applications Cloud and trading partner. Whenever you set up a trading partner, you can link it with a service provider.

- **Note:** You must be signed in as a supplier and must have access to the Supplier task.
- 1. On the Supplier Overview page, navigate to the Supplier Business Classifications section and click the supplier.
- 2. On the Edit Supplier page, switch to the Sites tab and click the required site.
- 4. Under Associated Collaboration Documents, click Manage Trading Partners.
- 5. On the Manage Trading Partners dialog box, click **Create > Create Trading Partner with Service Provider**.
- 6. On the Create Trading Partner dialog box, fill the required details.
- 7. Click **Actions > Add Row** and fill the details to associate collaboration documents with the trading partner. You may add multiple documents.

- **Note:** To enable exchange of messages, you must set the collaboration document status to **Active**.
- 8. Click Save and Close.
- 9. On the Manage Trading Partners dialog box, click **OK**.
- 10. On the Edit Site page, click **Save and Close**.
- 11. On the Edit Supplier page, click **Save and Close**.

To update the details for a trading partner, use the edit option on the Manage Trading Partners dialog box.

Creating a Trading Partner without a Service Provider: Procedure

You can exchange messages with a trading partner directly, without using a service provider.

- **Note:** You must be signed in as a supplier and must have access to the Supplier task.
- 1. On the Supplier Overview page, navigate to the Supplier Business Classifications section and click the supplier.
- 2. On the Edit Supplier page, switch to the Sites tab and click the required site.
- 4. Under Associated Collaboration Documents, click Manage Trading Partners.
- 5. On the Manage Trading Partners dialog box, click **Create > Create Trading Partner without Service Provider**.
- 6. On the Create Trading Partner dialog box, fill the required details.

- **Note:** By default, the Service Provider is set to None because this setup doesn’t involve a service provider.
- 7. Click **Actions > Add Row** and fill the delivery method, outbound collaboration message, and inbound collaboration message details on the respective tabs.

- **Note:** The following table contains some tips on filling the important information on each tab.
Oracle SCM Cloud
Implementing Order Management

Chapter 9
Performing Other Implementation Work

<table>
<thead>
<tr>
<th>Tab</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delivery Methods</td>
<td>- Provide the endpoint URL and the associated authentication credentials to initiate the collaboration messaging web service. It must be in the format http://&lt;server&gt;:&lt;port&gt;/&lt;context&gt;, where &lt;context&gt; contains the name of the web service as defined in the application.</td>
</tr>
<tr>
<td>Outbound Collaboration Messages</td>
<td>- Specify a unique name and select the collaboration message. The associated details automatically appear in the row.</td>
</tr>
<tr>
<td></td>
<td>- Select a delivery method.</td>
</tr>
<tr>
<td></td>
<td>- Set the outbound collaboration document status to <strong>Active</strong> or <strong>Loopback</strong>.</td>
</tr>
<tr>
<td>Inbound Collaboration Messages</td>
<td>- Specify a unique name and select the collaboration message. The associated details automatically appear in the row.</td>
</tr>
<tr>
<td></td>
<td>- Specify an XPath that identifies the application partner in the inbound collaboration document.</td>
</tr>
<tr>
<td></td>
<td>- Set the inbound collaboration document status to <strong>Active</strong> or <strong>Loopback</strong>.</td>
</tr>
</tbody>
</table>

8. Click **Save and Close**.
9. On the Manage Trading Partners dialog box, click **OK**.
10. On the Edit Site page, click **Save and Close**.
11. On the Edit Supplier page, click **Save and Close**.

To update the details for a trading partner, use the edit option on the Manage Trading Partners dialog box.

Managing Associated Collaboration Documents: Procedure
To set up collaboration messaging, you must associate the supplier site with a trading partner, and select the documents you want to exchange with that partner. The documents that you set up here are associated with trading partners or the service providers of those trading partners.

Note: You must be signed in as a supplier and must have access to the Supplier task.

1. On the Supplier Overview page, navigate to the Supplier Business Classifications section and click the supplier link.
2. On the Edit Supplier page, switch to the Sites tab and click the required site.
4. Under Associated Collaboration Documents, click **Edit**.
5. On the Edit Associated Collaboration Documents dialog box, click **Add Row** and fill the details required to set up the document. The read-only particulars appear based on the selected details.
6. Click **Save** and repeat the steps to add more documents or click **Save and Close** to return to the previous page.
7. On the Edit Site page, click **Save and Close**.
8. On the Edit Supplier page, click **Save and Close**.

Related Topics
- Importing Collaboration Messaging Setup Data for Procure to Pay: Procedure
FAQs for Define Collaboration Messaging

What are the different undelivered collaboration message error statuses?
The following table describes the main differences among the various error statuses.

<table>
<thead>
<tr>
<th>Error</th>
<th>B2B Error</th>
<th>Hold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indicates that the messages haven’t been delivered because of a validation, configuration, or processing error in Collaboration Messaging Framework.</td>
<td>Indicates that the B2B component of the SOA suite couldn’t deliver the message because of a configuration or processing error.</td>
<td>Indicates that messages haven’t been processed because an administrator has put them on hold.</td>
</tr>
</tbody>
</table>

What are the different undelivered collaboration message error types?
The following table describes the main differences among the message error types.

<table>
<thead>
<tr>
<th>Document Retrieval Error</th>
<th>Inbound Processing Error</th>
<th>Outbound Processing Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occurs when the collaboration messaging framework fails to retrieve the document associated with a collaboration event.</td>
<td>Occurs when the collaboration messaging framework can’t process inbound messages because of setup or business rule validation issues.</td>
<td>Occurs when the collaboration messaging framework can’t process outbound messages because of setup or business rule validation issues.</td>
</tr>
</tbody>
</table>

What happens if I don’t enable a document type for storage?
If you don’t enable a document type for storage, the message processing and delivery details of such documents aren’t stored in the log table.

Why did my message fail?
To know the cause of a message failure, search for the failed message on the Manage Failed Collaboration Messages page. When you click the message row, the cause of the failure appears under Processing History.

Configuring Collaboration Messaging for a Customer: Procedure

Using this task, you can associate a customer account with an existing trading partner and select the collaboration messaging documents to be exchanged with the customer.

To configure collaboration messaging for a customer:

1. On the Collaboration Messaging Overview page, click the Tasks icon, and select the Manage Customer Collaboration Configuration task.
2. Search for the customer account, select the row, and click Edit Collaboration Configuration.
3. Under Associated Service Providers, click Actions - Add Row and fill the details of the service provider and the trading partner.
4. Click Actions - Add Row and select at least one collaboration document.
5. Set the Association Status to Active to enable messaging with the selected service provider.
6. Click **Save and Close**.

Administering Social Conversations in Order Management: Procedure

You can enable Oracle Social Network in Order Management Cloud so that your user community can engage in social conversations about sales orders.

Administer social conversations for Order Management:

1. Navigate to the Manage Oracle Social Network Objects page.
2. In the Name column, expand **Order Management**, and then click **Sales Order**.
3. Click **Enable Object**.
4. In the Enable Object dialog, choose one of the following options, and then click **OK**.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>Create social conversations only for conversations that the Order Entry Specialist manually initiates for each sales order. It is strongly recommended that you choose Manual.</td>
</tr>
<tr>
<td>Automatic</td>
<td>Create a social conversation for every sales order. In most situations, it is not necessary to choose Automatic.</td>
</tr>
</tbody>
</table>

5. In the Attributes area, click **Actions**, and then click **New**.
6. In the Select Attributes dialog, add a check mark to the **Enabled** option for each attribute that Order Management must include in the conversation.

Order Management will stamp each attribute and the value for each attribute that you enable into the social conversation thread. It will also stamp the current attribute value every time the conversation updates.

**Related Topics**

- Social Networking: Overview
- Manage Social Networking: Overview
Glossary

**action**
The kind of access, such as view or edit, named in a security policy.

**activity**
An event that occurs outside of Order Management Cloud. 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. A user can perform an activity as part of completing order fulfillment. Order Management can assign an activity to a user.

**advance shipment notice**
A notification that a delivery is pending.

**application partner**
A B2B trading partner defined in Oracle Applications Cloud that engages in the collaboration messaging activity. For example, a supplier site, or a customer account.

**approval rule**
A statement that specifies an IF THEN condition. The condition determines how and when to route a sales order through sales order approval. For example, if the order total is greater than $10,000, then get approval from an order manager.

**assignment rule**
A rule that assigns an orchestration process to fulfillment lines.

**assignment set**
A group of sourcing rules, bills of distribution, or both, in which each rule or bill is assigned to an assignment level, and to attribute values for the attributes applicable to that assignment level. An assignment set defines a supply chain.

**attribute**
The property of a sales order, order line, fulfillment line, or orchestration process. Customer and Ship-To Address are each an example of an attribute of a sales order. Quantity is an example of an attribute of an order line. Actual Start Date is an example of an attribute of an orchestration process. Jeopardy Score is an example of an attribute of a fulfillment line.

**back-to-back fulfillment**
A business flow where supply is reserved against a particular sales order until shipping. You create supply for a back-to-back order using one or more of the following: make, buy, transfer, and reservation of on-hand supply.

**blanket purchase agreement**
A long term agreement for the purchase of goods and services from a supplier. It includes terms and conditions, details of the goods or services to be purchased from the supplier, and negotiated amounts. The blanket agreement does not contain
delivery dates, individual delivery quantities, or amounts. Complete details necessary to supplier fulfillment are specified subsequently in purchase orders issued against the agreement.

**bucket**
A container that defines a range of values.

**bucket set**
A container that holds the overall range of values that a group of buckets defines.

**business event**
An occurrence in an application or system that might be significant to objects in some other application or system. Applying a hold on a sales order is an example of a business event.

**business event trigger point**
A condition that causes a business event to occur.

**business object**
A resource in an enterprise database, such as an invoice or purchase order.

**business rule**
A statement that describes how to implement a business policy or make a business decision. For example, a business rule can enforce a spending policy, constrain a process so it meets a regulatory requirement, compute a discount or premium, provide an offer according to customer value, and so on.

**business unit**
A unit of an enterprise that performs one or many business functions that can be rolled up in a management hierarchy.

**catalog search line**
Area in the Order Lines tab on the Create Order page in Order Management that allows the Order Entry Specialist to search for an item before adding it as an order line.

**client business unit**
A business unit that can requisition and process invoices for a supplier site. One or more procurement business units can service a client business unit.

**compensation pattern**
A rule that you can set on an orchestration process step that specifies the adjustments to make when an order changes. Undo, Redo, Update, Cancel, and None are each an example of a compensation pattern. For example, assume the compensation pattern for a Create Shipment step is Redo, and that this step calls the Cancel service and the Create service. If Order Management receives a change order that includes a new warehouse for this step, then it runs the Cancel service and the Create service again.
configuration model
A hierarchical structure that specifies how to represent a configured item. A desktop computer that includes a monitor, keyboard, mouse, and external speakers is an example of a configuration model. In this example, the desktop computer is the parent item, and the peripherals are the child configure options. Oracle Fusion Configurator specifies the structure, which includes referenced models, configurator rules, user interface specifications, and so on.

Configurator
A feature that allows you to configure a configured item. The Configurator allows you to choose configure options for an item from a list of available options. It uses constraints to make sure the items that you choose fit together correctly. It reduces configuration errors, which in turn reduces change order processing and any rework that is required during order fulfillment.

configure option
A child component of a configured item. A desktop computer is an example of a configured item. The hard drive, monitor, and mouse are each an example of a configure option. You can choose the value for a configure option from a list of available options.

configured item
An item that includes one or more configure options that you can configure. A desktop computer where you choose the hard drive, monitor, and mouse is an example of a configured item. The hard drive, monitor, and mouse are each an example of a configure option.

connector
Configuration templates that include mapping and protocol settings of the target PLM system to be integrated with Oracle Fusion PLM.

constraint entity
The business object or orchestration process that a processing constraint constrains.

cost of change
A numeric value that measures how much a change impacts an orchestration process. For example, the monetary cost to a company, or the difficulty that is associated with incorporating the change.

coverage
An agreement that adds value to an item by providing support, service, repair, replacement, and so on. Extended warranty, service level agreement, and software maintenance are each an example of a coverage.

coverage item
An item that adds value by providing coverage for another item. For example, a six month warranty is a coverage item for a laptop computer, and the laptop computer is the covered item. Extended Warranty, Service Level Agreement, and Software Maintenance are each an example of a coverage.

covered item
An item that a coverage item covers. For example, a six month warranty is a coverage item for a laptop computer, and the laptop computer is the covered item.
credit check
Calculation that determines whether or not the credit that is currently available for the customer account is equal to or greater than the credit amount of the transaction.

cross-reference
A reference that relates business data between an order capture system, order fulfillment system, and Order Management Cloud. For example, an item cross-reference can create a relationship between an item that resides in a source system, such as Widget A, and a representation of this same item in Order Management. Cross-references help to manage the representation of data across systems.

customer
A customer is someone with whom you have a selling relationship. The selling relationship can result from the purchase of products and services, or from the negotiation of terms and conditions that provide the basis for future purchases.

decision table
A set of IF statements displayed as a table. You can use a decision table when you define a rule that includes more than one IF statement.

descriptive flexfield
Customizable expansion space, such as fields used to capture additional descriptive information or attributes about an entity, such as a customer case. You may configure information collection and storage based on the context.

drop shipment
A type of shipment where the shipper is not the seller. This shipper ships directly to the customer. The shipper for an internal drop shipment is another business unit. The shipper for an external drop shipment is an external supplier.

duty role
A group of function and data privileges representing one duty of a job. Duty roles are specific to applications, stored in the policy store, and shared within an application instance.

extensible flexfield
Expandable fields that you can use to capture multiple sets of information in a context or in multiple contexts. Some extensible flexfields let you group contexts into categories.

external interface routing rule
Routing rule that specifies how to route a fulfillment request to a fulfillment system. You can specify each rule so it selects the fulfillment system connector according to order, fulfillment line, or orchestration process attribute.

external system or external application
A system or application that is external to and not part of Order Management. An order capture system that resides upstream of Order Management is an example of an external system. A fulfillment application that resides downstream of Order Management is an example of an external application.
**flexfield**
A flexible data field that you can configure such that it contains one or more segments or stores additional information. Each segment has a value and a meaning.

**flexfield segment**
An extensible data field that represents an attribute and captures a value corresponding to a predefined, single extension column in the database. A segment appears globally or based on a context of other captured information.

**fulfillment line**
A request to fulfill an item. A fulfillment line contains information about this request, such as customer, sales order number, item, quantity, scheduled ship date, actual ship date, shipping method, and so on. You can schedule a fulfillment line, reserve product for it, substitute an item for it, change the warehouse, change the shipping method, change the demand class, and so on.

**fulfillment system**
A system that resides downstream of Order Management Cloud that is responsible for fulfilling a sales order. For example, a fulfillment system is responsible for shipping the physical goods to the customer, such as shipping a laptop computer from a warehouse to a customer location.

**fulfillment task**
A type of task that Order Management Cloud performs to fulfill a sales order. Schedule and Ship are each an example of a fulfillment task. Order Management might use multiple orchestration process steps to complete a single fulfillment task.

**hold**
A temporary stoppage of processing an order line or fulfillment line during order fulfillment.

**hybrid configuration model**
A type of configuration model that includes an assemble-to-order model as the child and a purchase-to-order model as the parent. Order management uses one blanket purchase agreement to source the assemble-to-order configure options, and a different blanket purchase agreement to source the purchase-to-order configure options.

**import file**
A flat file that you use as an interface between your source system and Order Management. You copy source orders into this file, and then use a scheduled process to import orders from the import file into Order Management.

**integration algorithm**
A set of rules that uses conditional logic, variables, functions, and code to manipulate data in a service data object (SDO). An integration algorithm manages an integration between Order Management Cloud and some other application.

**interface table**
A database table that stores data during data transfer between applications or from an external system or data file.
internal identifier
An identifier that Order Management uses to create a reference between a lookup that you specify for an item, customer, or reference data, to information from the Product Information Manager work area, Trading Community Architecture, or collection data. This data resides in the Order Orchestration and Planning Repository. For example, if your order uses Payment Terms, then you must set up this value in Oracle, and then set up collections processing.

internal material transfer
Goods or services moving between internal organizations. Internal material transfers are executed through transfer orders or interorganization transactions.

internal order
A sales order that you use transfer an item from one part of your organization to another part of the same organization. For example, to transfer an item from Computer Service and Rentals - Warehouse A, to Computer Service and Rentals - Warehouse B. For another example, to transfer an item from Computer Service and Rentals - Korea Manufacturing, to Computer Service and Rentals - USA Distribution Center.

inventory organization
A logical or physical entity in the enterprise that tracks inventory transactions and balances, stores definitions of items, and manufactures or distributes products.

item
A product that resides in the Product master database. Order Management displays items in the Item column when you add an order line to a sales order. For example, items for a company that sells hardware might include nuts, bolts, and screws. Items for a company that sells computers might include hard drives, computer monitors, and desktop computers. Items for a wireless service provider might include service plans, international calling services, and instant messaging services. Items for an insurance company might include an item that applies for a period of time, such as fire insurance coverage.

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.

item validation organization
An inventory organization whose primary or secondary unit of measure is used as the costing unit of measure for the item in the cost organization to which that inventory organization belongs. The item master organization can also be designated as the item validation organization.

jeopardy
Measures the level of risk that is associated with the delay of an orchestration process task as low, medium, or high.

jeopardy priority
A value that describes the level of risk that is associated with delaying a fulfillment task. Order Management displays a jeopardy priority value of Low, Medium, or High.
jeopardy score
A numeric ranking that indicates the severity of a delay in completing a fulfillment task.

jeopardy threshold
The range of time that a task is delayed.

job role
Log in that allows each user to access the application features that are required to perform a job in you organization, and to access the data that these features reference. Example job roles in Order Management include Order Entry Specialist, Order Manager, Order Administrator and Supply Chain Application Administrator.

job role
A role, such as an accounts payable manager or application implementation consultant, that usually identifies and aggregates the duties or responsibilities that make up the job.

kit
A configured item that includes one or more configure options. The Order Management work area allows the Order Entry Specialist to modify configure options for a configured item, but not for a kit.

lead-time
The amount of time that Order Management requires to complete an orchestration process step, including wait steps and pause steps.

lead-time expression
A business rule that determines the amount of time you expect an orchestration process step will require to finish. You can use it to calculate planning for an orchestration process. For example, an item that uses complex packing might require a longer lead-time for shipping.

line selection rule
A type of rule that determines whether or not an orchestration process step runs a fulfillment line. For example, a line selection rule can specify not to ship a nonshippable item, such as a contract that a customer completes online.

long-running task
A type of task that includes a wait step, such as a shipping step, receiving step, and so on.

lookup
A value that an Oracle application displays in a list of values. This list typically that allows the user to choose a single value, such as in a drop-down list. The application uses various logic to search for the lookup values that it displays.
Navigator
The menu in the global header that you can use to open the work areas and dashboards that you have access to.

offering
A comprehensive grouping of business functions, such as Sales or Product Management, that is delivered as a unit to support one or more business processes.

Oracle BI Publisher
An Oracle application that performs the following formatting tasks for Oracle Fusion Payments: 1) formats extracted data into a message, such as a settlement batch or payment file, that can be understood by the payment system, 2) supports remittance advice formatting and delivery.

orchestration planning
The process of orchestrating and planning the fulfillment of a sales order. For example, planning dates, planning shipments, and so on. Replanning is the process of updating an orchestration plan for a sales order to accommodate a change that occurs to this sales order or in the fulfillment environment. For example, when a customer changes the quantity of an existing sales order.

orchestration process
A process that includes a sequence of steps. These steps process one or more fulfillment lines during order fulfillment. An orchestration process coordinates the orchestration of physical goods and activities in a single order, and it automates order orchestration across fulfillment systems. It contains the instructions that describe how to process an order, such as the steps and services to use, step dependencies, conditional branching, lead-time information, how to handle change orders, and the status values to use. It describes how to schedule, reserve, ship, return, and invoice a sales order.

orchestration process branch
A path in an orchestration process that Order Management runs when a condition is met.

orchestration process class
A set of statuses that you can assign to an orchestration process.

orchestration process instance
An instance of an orchestration process definition. An orchestration process definition specifies the steps, behavior, processing, and so on that automates order orchestration across fulfillment systems. An orchestration process instance is the run time representation of this definition. It includes temporary data and transactional data that orchestration requires to process a specific sales order or fulfillment line.

orchestration process step
A step of an orchestration process that specifies the task layer service that Order Management calls or the subprocess to start. Each step references a task type, task, and service. A step might also specify branching. A task includes one or more orchestration process steps.

orchestration subprocess
A set of sequential steps. This set of steps is an orchestration process that another orchestration process references.
order compensation
The process of modifying an order so that it accurately reflects the changes that have occurred to the order. For example, assume Order Management receives a change order that specifies to ship an item from a different warehouse. In this example, it adjusts the order so that this order uses the different warehouse, updates delivery dates so that these dates accurately reflect the time required to ship the item from the different warehouse, and so on.

Order Entry Specialist
Job role that allows an end-user to create and manage sales orders in the Order Management work area.

order fulfillment
The part of the order management lifecycle that fulfills a sales order. Order fulfillment typically begins after the user submits a sales order in Order Management Cloud. Order Management schedules, ships, and bills a sales order during order fulfillment. Order Management communicates with a downstream fulfillment system during order fulfillment.

order header
The fields of a sales order that apply to the entire sales order, such as Customer, Contact, Purchase Order, Business Unit, and Sales Credits. For example, on the Create Order page, the fields that display above the Order Lines area constitutes the order header.

order import template
A template that you can use to structure your source data so that you can import it into Order Management.

order line
A line on a sales order that specifies details about an item, such as item number, quantity, shipping details, billing details, and so on. Order Management transforms an order line into a fulfillment line when the Order Entry Specialist clicks Submit.

order orchestration
An order management feature that uses an automated sequence of fulfillment steps to process a sales order through order management, across one or more source systems, and across one or more order fulfillment systems.

order transformation
The process of transforming a source order into the business objects that define a sales order. These business objects include the order, order lines, fulfillment lines, orchestration processes, and return fulfillment lines.

order validation
The process that Order Management performs to make sure a sales order meets a set of criteria. For example, to make sure the sales order includes details for each required field and other details according to predefined set ups and set ups that you define.

order-to-cash
The front-to-back flow of information that Order Management Cloud uses to process a sales order through various systems and applications, from order entry to order fulfillment.
**pause task**
A task that temporarily stops an orchestration process from running so that it can wait for a condition to be met.

**planning data repository**
The set of data collected from source systems and stored for use by order management, order promising, and supply chain planning processes.

**predefined**
An object, integration, or configuration that comes already defined with an Oracle application and is ready to use with little or no modification.

**privilege**
A grant of access to functions and data; a single, real world action on a single business object.

**processing constraint**
A rule that controls who can change a sales order, what can change in this sales order, and when the change can occur.

**requisition**
Document generated for a department or an individual to notify the purchasing department of goods or services it is ordering, their quantity, and the expected delivery time frame.

**requisitioning business unit (BU)**
The business unit that manages and owns the requisitioning transaction.

**role**
Controls access to application functions and data.

**root parent**
The item that resides at the top of the hierarchy of a configured item. For example, if a configured item includes a top level for computer, a mid level for chassis, and a bottom level for graphics card, then the computer is the root parent of the configuration, and the chassis is the parent of the graphics card.

**routing rule**
A rule that determines how to route data through a system or application. It typically uses an IF-THEN structure that specifies a condition and an action to take depending on the result of the condition. For example, an external interface routing rule can specify how to route a fulfillment request to a fulfillment system depending on the current state of the fulfillment environment, such as available inventory, transit time, and so on.

**sales order**
A contractual document between a sales organization and a customer. You create a sales order in the Order Management work area. Order Management also transforms a source order it receives from a source system into a sales order that it can submit to order fulfillment.
sales order approval
A process that includes one or more approvers who must approve a sales order before Order Management sends it to order fulfillment. An approval rule determines whether or not approval is required.

scheduled process
A program that you run to process data and, in some cases, generate output as a report.

selling business unit
A business unit that is a profit center that resides in the selling legal entity. If you specify All Business Units as the selling business unit, then the application applies the buy and sells terms to all the profit center business units that the selling legal entity contains.

service data object
A data structure that does not conform to a single programming language. It facilitates communication between applications. It allows a service to call another service without having to conform to the requirements of a single language. A service data object typically includes elements arranged in a tree structure that includes a root and branches, and it allows the calling service to access the data that these elements reference.

service mapping
A pricing entity that specifies a group of entities and attributes that a service can retrieve and update. You can use it to model entities and attributes in a declarative environment without having to write software code. The service mapping allows an application, such as Order Management Cloud, to price an item without having to get the pricing schema that Pricing requires.

service provider
An intermediary that facilitates exchange of messages between Oracle Applications Cloud and trading partners.

set
Classified and grouped reference data that organizational entities share.

shipment set
A set of order lines that Order Management ships together as one group. All of these order lines ship and arrive on the same date, although they might be spread across more than one package, depending on packing requirements.

shippable item
An item that Order Management can physically ship to the customer. Order Management typically fulfills a shippable item from a warehouse, and uses a carrier to physically move the item through the supply chain. A parcel service that uses a truck to transport the item from one location to another location, such as from a warehouse to a residence, is an example of a carrier. A laptop computer is an example of a shippable item. A warranty for a laptop computer is an example of a nonshippable item. Other examples of nonshippable items include a subscription to an online magazine, or a video that your customer can only stream from the cloud.
source data
Data that you import from an external system into Order Management.

source order
An order that you import into Order Management from a source system. Order Management transforms a source order to a sales order when you import it from a source system.

source system
System where the sales order was created. Order Management Cloud and an order capture system are each an example of a source system. A source system provides business application information to an Oracle application. Oracle can use this information to extract fulfillment data and planning data into data files.

sourcing rule
A specification of the means by which organizations can replenish items.

split fulfillment line
A supply chain technique that fulfills an order in a timely and efficient way. For example, assume a fulfillment line includes a quantity of 30 items, but the preferred warehouse, Warehouse A, only has 20 items in stock. Warehouse B has 15 items in stock. To avoid a delay, you can split the fulfillment line so that it gets the items it requires from both warehouses.

stamp
To impress a snapshot of data onto an object. For example, a sales order might contain a set of attribute values that exist on a specific date, such as quantity equals four and color equals blue on October 9, 2016. These values might change at some point, such as during order fulfillment. A stamp impresses the values of this sales order from a specific point in time onto an object, such as a social conversation, so that it provides an audit trail that you can use to communicate the values of attributes that existed in an historical context.

start-after condition
A condition on a Pause task in an orchestration process step that specifies when to pause processing before calling the next step, or when to resume processing.

status condition
A rule that determines when a sales order or object reaches a status. The following rule is an example of a status condition: If the Schedule task status equals Scheduled, then set the orchestration process status to Scheduled.

subscriber
A system or application that subscribes to a web service or business event. For example, a source system can subscribe to a business event that sends notifications about changes to sales orders. If a change occurs, then a web service sends information about this event to this source system in the web service payload. This configuration is similar to a subscription service that regularly sends information to subscribers, or that sends information when an event occurs.

subscription
An item that provides a product or service that recurs. For example, a one year subscription to a magazine, a 90 day subscription for cell phone service, and a six month subscription for software usage are each an example of a subscription.
A subscription might or might not bill the customer on a recurring basis. Fulfillment does recur periodically. For example, a weekly magazine fulfills one time each week.

**substitution**
Replacing one item with some other item in a sales order. For example, replacing an 80GB hard drive with a 100GB hard drive because there are no 80GB hard drives in inventory.

**task type**
A group of services that Order Management uses to perform a fulfillment task. It represents a common business function that Order Management uses so that it can process a sales order from the time that a user creates it to the time that Order Management sends this sales order to an order fulfillment system. Shipment is an example of a task type. It is a set of services that communicates with a shipping fulfillment system to ship the items that a fulfillment line references.

**trade compliance**
A structure of rules that makes sure trade between countries or regions occurs only according to the approved laws and guidelines that these countries or regions use.

**trade compliance policy**
A statement that controls trade. For example: Do not ship fire arms to a country that is on the sanction list.

**trade compliance screening**
The act of examining a sales order to determine whether it meets trade compliance requirements.

**trade compliance status**
The current status of the trade compliance review for a sales order. For example: Passed, Failed, or Under Review.

**trade control type**
The type of trade control that Order Management references when it verifies a sales order, such as antiterrorism or firearms convention.

**trading partner**
A B2B partner engaging in the collaboration messaging activity but defined outside Oracle Applications Cloud.

**transformation rule**
A rule that transforms a source order that resides on your source system to a sales order that resides in Order Management.

**user credential key**
A user and password combination that authenticates communication between Order Management and an external fulfillment system.