

Sterling Selling and Fulfillment Foundation



# Logistics Management Configuration Guide

*Release 9.2*



Sterling Selling and Fulfillment Foundation



# Logistics Management Configuration Guide

*Release 9.2*

**Note**

Before using this information and the product it supports, read the information in "Notices" on page 289.

**Copyright**

This edition applies to the 9.2 Version of IBM Sterling Selling and Fulfillment Foundation and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1999, 2012.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

## Chapter 1. Introduction . . . . . 1

Introducing the Logistics Management Business Application . . . . .	1
Business Models . . . . .	1
Multidivisional Corporation . . . . .	1
Third-Party Logistics . . . . .	2
Marketplace . . . . .	2
Logistics Management Configuration . . . . .	2
Icons Used in This Guide . . . . .	4

## Chapter 2. Navigating the Applications Manager . . . . . 7

Starting the Applications Manager . . . . .	7
The Applications Manager Layout . . . . .	7
Application Rules Side Panel . . . . .	8
Work Area . . . . .	14
Actions Available in the Applications Manager . . . . .	18
Using the Applications Manager's Lookup Functionality . . . . .	19
Viewing the Document Types Associated with an Application . . . . .	19
Viewing the User Logged into the Applications Manager . . . . .	20
Using Lists and List Filtering . . . . .	21
Date and Time Entry . . . . .	21
Troubleshooting Errors . . . . .	22
Using Special Characters . . . . .	22

## Chapter 3. Configuring Cross Application Carrier Services . . . . . 23

Configuring Cross Application Carrier Services . . . . .	23
Creating a Carrier Service Code . . . . .	23
Modifying a Carrier Service Code . . . . .	24
Deleting a Carrier Service Code . . . . .	24
Creating a Carrier Delivery Schedule . . . . .	24
Modifying a Carrier Delivery Schedule . . . . .	25

## Chapter 4. Configuring Cross Application Carrier Special Services . . . . . 27

Configuring Cross Application Carrier Special Services . . . . .	27
Creating a Carrier Special Service Code . . . . .	27
Modifying a Carrier Special Service Code . . . . .	28
Deleting a Carrier Special Service Code . . . . .	28

## Chapter 5. Configuring Cross Application Logistics Rules. . . . . 29

Configuring Cross Application Logistics Rules. . . . .	29
Creating a Logistics Rule . . . . .	29

## Chapter 6. Configuring Transportation 31

Configuring Transportation . . . . .	31
Creating an Activity Code . . . . .	31

Modifying an Activity Code . . . . .	31
Deleting an Activity Code . . . . .	32

## Chapter 7. Configuring a Document's Modification Reasons . . . . . 33

Configuring a Document's Modification Reasons . . . . .	33
Creating a Modification Reason. . . . .	33
Modifying a Modification Reason . . . . .	33
Deleting a Modification Reason. . . . .	34

## Chapter 8. Configuring a Load Document's Hold Types . . . . . 35

Configuring a Load Document's Hold Types . . . . .	35
Load Document: Creating a Hold Type . . . . .	35
Load Document: Modifying a Hold Type . . . . .	37
Load Document: Deleting a Hold Type . . . . .	38

## Chapter 9. Configuring a Load Document's Pipeline . . . . . 39

Configuring a Load Document's Pipeline . . . . .	39
Defining Process Type Details . . . . .	39
Process Type Pipeline Configuration . . . . .	39
Defining Pipeline Determination . . . . .	40
Load Document: Pipelines . . . . .	40
Load Document: Transactions . . . . .	41
Load Document: Statuses . . . . .	42
Load Document: Conditions . . . . .	43
Load Document: Actions . . . . .	44

## Chapter 10. Configuring a Document's Instruction Types . . . . . 45

Configuring a Document's Instruction Types . . . . .	45
Creating an Instruction Type. . . . .	45
Modifying an Instruction Type . . . . .	45
Deleting an Instruction Type. . . . .	46

## Chapter 11. Configuring a Load Document's Charge Categories . . . . . 47

Configuring a Load Document's Charge Categories . . . . .	47
Creating a Charge Category . . . . .	47
Adding a Charge Name Associated with a Charge Category . . . . .	47
Modifying a Charge Name Associated with a Charge Category . . . . .	48
Deleting a Charge Name Associated with a Charge Category . . . . .	48
Modifying a Charge Category . . . . .	48
Deleting a Charge Category . . . . .	49

## Chapter 12. Configuring a Load Document's Purge Criteria . . . . . 51

Configuring a Load Document's Purge Criteria . . . . .	51
--	----

Modifying a Load Document Type's Purge Criteria Rule . . . . .	51
--	----

**Chapter 13. Configuring a Load Document's Load Types . . . . . 53**

Configuring a Load Document's Load Types . . . . .	53
Creating a Load Type . . . . .	53
Modifying a Load Type . . . . .	53
Deleting a Load Type . . . . .	53

**Chapter 14. Configuring a Shipment Document's Hold Types . . . . . 55**

Configuring a Shipment Document's Hold Types . . . . .	55
Creating a Hold Type . . . . .	55
Modifying a Hold Type . . . . .	57
Deleting a Hold Type . . . . .	58

**Chapter 15. Configuring a Load Document's Stop Types . . . . . 59**

Configuring a Load Document's Stop Types . . . . .	59
Modifying a Stop Type . . . . .	59

**Chapter 16. Configuring a Load Document's Modification Components . . . . . 61**

Configuring a Document's Modification Components . . . . .	61
Defining Custom Modification Types . . . . .	61
Creating a Custom Modification Type . . . . .	62
Modifying a Custom Modification Type . . . . .	62
Deleting a Custom Modification Type . . . . .	63
Defining Status Modification Groups . . . . .	63
Configuring a Load Document's Modification Rules . . . . .	63
Load Document Modification Types . . . . .	63
Specifying a Condition for a Status Modification Group . . . . .	65
Overriding Status Modification Groups . . . . .	65
Inheriting Modification Rules from a Status Modification Group . . . . .	66
Resolving Conflicts Between Status Modification Groups . . . . .	66
Creating a Status Modification Group . . . . .	68
Modifying a Status Modification Group . . . . .	70
Deleting a Status Modification Group . . . . .	71
Changing a Load Document Type's Modification Requiring Auditing Rules. . . . .	71

**Chapter 17. Configuring an Order Document's Shipment Specific Components . . . . . 73**

Configuring an Order Document's Shipment Specific Components . . . . .	73
Defining Shipping Modification Groups . . . . .	73
Defining Shipping Hold Types . . . . .	73
Defining Shipping Process Type Details . . . . .	73
Defining Shipping Process Model . . . . .	74

**Chapter 18. Configuring Logistics Components . . . . . 75**

Defining Logistics . . . . .	75
Defining Freight Terms . . . . .	75
Defining Shipment Planning. . . . .	77
Creating a Routing Guide . . . . .	78
Modifying a Routing Guide Line . . . . .	79
Deleting a Routing Guide . . . . .	84

**Chapter 19. Time-Triggered Transaction Reference . . . . . 85**

Time-Triggered Transaction Reference. . . . .	85
Running Time-Triggered Transactions. . . . .	86
Steps to Complete Before Scheduling Time-Triggered Transactions . . . . .	86
Configuring Communication Between an Agent and a JMS Server . . . . .	86
Prerequisites . . . . .	87
Create an Initial Context Factory Code . . . . .	87
Define the Transaction Information . . . . .	88
Business Process Time-Triggered Transactions . . . . .	89
Asynchronous Request Processor . . . . .	89
Case Insensitive Data Loader . . . . .	90
Change Load Status . . . . .	92
Change Shipment Status . . . . .	93
Close Delivery Plan . . . . .	94
Close Load . . . . .	96
Close Manifest . . . . .	97
Close Order . . . . .	99
Close Receipts . . . . .	100
Close Shipment. . . . .	102
Collect Shipment Statistics . . . . .	103
Consolidate Additional Inventory . . . . .	105
Consolidate To Shipment . . . . .	106
Create Catalog Index . . . . .	108
Create Chained Order . . . . .	112
Create Derived Order. . . . .	113
Create Order Invoice . . . . .	115
Create Shipment Invoice. . . . .	116
ESP Evaluator . . . . .	117
Item Based Allocation . . . . .	119
Mark Load as Trailer Loaded . . . . .	123
Match Inventory . . . . .	124
Payment Collection . . . . .	125
Payment Execution . . . . .	127
Post Inventory Match. . . . .	129
Process Order Hold Type . . . . .	130
Process Work Order Hold Type . . . . .	132
Publish Negotiation Results . . . . .	133
Release . . . . .	135
Route Shipment . . . . .	137
Schedule . . . . .	138
Send Invoice . . . . .	141
Send Item Changes . . . . .	143
Send Customer Changes. . . . .	144
Send Order . . . . .	146
Send Release . . . . .	147
Start Order Negotiation . . . . .	148
Synchronize Colony Map . . . . .	149
Update Best Match Region . . . . .	151

PopulateOwnershipTransferSummary . . . . .	153
Time-Triggered Purge Transactions . . . . .	154
Purge Strategy . . . . .	154
Configuring Purge Transaction Log Files . . . . .	154
Available Purges . . . . .	155
Task Queue Syncher Time-Triggered Transactions	232
Load Execution Task Queue Syncher . . . . .	232
Order Delivery Task Queue Syncher . . . . .	233
Order Fulfillment Task Queue Syncher . . . . .	234
Order Negotiation Task Queue Syncher . . . . .	235
Quote Fulfillment Task Queue Syncher . . . . .	236
Monitors . . . . .	237
Availability Monitor . . . . .	238
Exception Monitor . . . . .	239
Inventory Monitor . . . . .	241
Negotiation Monitor . . . . .	242
Enhanced Order Monitor . . . . .	243
Enhanced Quote Monitor . . . . .	246
Enhanced Return Monitor . . . . .	248
Real-time Availability Monitor . . . . .	251
Shipment Monitor . . . . .	258
Work Order Monitor . . . . .	260

## Chapter 20. Condition Builder

<b>Attributes . . . . .</b>	<b>263</b>
Condition Builder Attributes . . . . .	263
Sales Order . . . . .	264
Order Fulfillment . . . . .	264
Order Negotiation . . . . .	266
Outbound Shipment . . . . .	267
Receipt . . . . .	268
Planned Order . . . . .	268
Planned Order Execution . . . . .	268
Planned Order Negotiation . . . . .	268
Return Order . . . . .	268
Reverse Logistics . . . . .	268

Return Shipment . . . . .	270
Return Receipt . . . . .	270
Template Order . . . . .	271
Purchase Order . . . . .	271
Purchase Order Execution . . . . .	271
Purchase Order Negotiation . . . . .	273
Inbound Shipment . . . . .	273
Purchase Order Receipt . . . . .	273
Transfer Order . . . . .	273
Transfer Order Execution . . . . .	273
Transfer Order Delivery . . . . .	273
Transfer Order Receipt . . . . .	274
Master Order Fulfillment . . . . .	274
Quote . . . . .	275
Quote Fulfillment . . . . .	275
Load Execution . . . . .	275
General . . . . .	276
WMS Putaway . . . . .	277
WMS Layout Definition . . . . .	277
WMS Inventory . . . . .	277
Trailer Loading . . . . .	277
Task Execution . . . . .	277
Move Request Execution . . . . .	278
Manifesting . . . . .	278
Over Pack Build . . . . .	278
Count Execution . . . . .	278
Pack Process . . . . .	279
Outbound Picking . . . . .	281
VAS Process . . . . .	281
Opportunity . . . . .	282
Opportunity Fulfillment . . . . .	282
Item Based Allocation . . . . .	283

<b>Index . . . . .</b>	<b>287</b>
------------------------	------------

<b>Notices . . . . .</b>	<b>289</b>
--------------------------	------------





---

## Chapter 1. Introduction

---

### Introducing the Logistics Management Business Application

The information in the Logistics Management configuration topics concentrates on the rules and setup configurations that make up the IBM® Sterling Logistics Management business application in the Applications Manager. The information is intended for both Hub and Enterprise administrators using the Applications Manager to set up the IBM Sterling Selling and Fulfillment Foundation environment. Business analysts should also use this information to plan appropriate business practices as they pertain to Sterling Selling and Fulfillment Foundation. Programmers should refer to the *Sterling Selling and Fulfillment Foundation: Customization Basics* for information about extending Sterling Selling and Fulfillment Foundation. System Integrators should refer to the *Sterling Selling and Fulfillment Foundation: Integration Guide* for information about integrating external applications with Sterling Selling and Fulfillment Foundation.

**Note:** You must have read and be familiar with the concepts and business functionality detailed in the *Sterling Selling and Fulfillment Foundation: Product Concepts Guide*.

The Applications Manager is a collection of all the rules and setup configurations necessary to implement Sterling Selling and Fulfillment Foundation organized so that configuration can be done for each business application separately. The following business applications can be configured within the Applications Manager:

- IBM Sterling Distributed Order Management
- IBM Sterling Global Inventory Visibility
- Catalog Management
- Sterling Logistics Management
- IBM Sterling Supply Collaboration
- IBM Sterling Reverse Logistics
- IBM Sterling Application Platform

---

### Business Models

There is no single business model that encompasses the environment in which all the Sterling Selling and Fulfillment Foundation applications can be used. Therefore, there is no single way to configure your Sterling Selling and Fulfillment Foundation environment.

For example, your company might be considered a multidivisional corporation, a third-party logistics company, or a marketplace business. Each of these business models require a different conceptual approach to the Sterling Selling and Fulfillment Foundation configuration.

#### Multidivisional Corporation

The multidivisional corporation model is a business corporation whose primary focus is managing purchase and sales activities. A typical multidivisional corporation can be a buyer, a seller, or both. It could also be a retailer, a manufacturer, or both. Whatever form the multidivisional corporation takes, it

normally has multiple channels with different types of customers, such as, consumers, retailers, dealers, and original equipment manufacturers.

In the multidivisional corporation model, each division might be set up as an Enterprise in Sterling Selling and Fulfillment Foundation. This setup allows both segregation of transactions by division and global visibility at the corporate level. Each Enterprise configures their own business rules, workflow, and transaction processing.

## **Third-Party Logistics**

Traditional third-party logistics companies provide a range of outsourced services such as warehousing, transportation, and contract manufacturing.

Large companies can gain the competitive advantage through the real-time management of their supply chains. These advantages include lower costs and improved customer service. Additionally, new sales channels such as web stores, hand-held devices, and in-store kiosks provide companies new methods of reaching their customers. All of these issues have increased the complexity of the fulfillment process.

In the third-party logistics model, each client might be set up as an Enterprise. This setup allows the third-party logistics Hub to have visibility of all transactions in the Hub environment, while the clients that are set up as Enterprises only have visibility to their own transactions. This allows the third-party logistics business to provide unique transaction processing to its clients.

## **Marketplace**

A marketplace is an online intermediary that connects Buyers and Sellers. Marketplaces eliminate inefficiencies by aggregating offerings from many Sellers or by matching Buyers and Sellers in an exchange or auction. For Buyers, they lower purchasing costs and help them reach new Sellers. For Sellers, they lower sales costs and give them access to new customers. It is a central location, or Hub, where a trusted intermediary integrates both procedures and technology to lower the costs and enhance the effectiveness of Buyer and Seller transactions.

In the marketplace model, each market might be set up as an Enterprise. This setup allows each market to be unique with their own product or service handling.

## **Logistics Management Configuration**

The Sterling Logistics Management application is a collection of common components used for creating delivery plans in the Application Console.

A delivery plan is a complete sequence of movements needed to deliver one or more orders from one or multiple origins to one or multiple destinations. A delivery plan is comprised of shipments, loads, origins, stops, and destinations.

### **Shipment**

A shipment is a delivery of one or more orders and order lines from a single shipper to a single consignee. A shipment can be carried through multiple loads and by multiple carriers.

## Load

A load carries one or more complete shipments (never a partial shipment) between two points. A load has one origin and one destination, but it can have multiple intermediate stops. Shipments can be added to a load at its origin or any intermediate stop and can be dropped off at the load destination or any intermediate stop.

## Origin

An origin is the node the load originally ships from.

## Stop

A stop is any location where a shipment is picked up or dropped off. A load has a stop sequence that determines its travel route.

## Destination

A destination is the last node or address in the load's travel route where all remaining shipments in a load are dropped off.

## Logistics Management Configuration: Carrier Services

Carrier Services is used for defining codes that identify the different carrier services a Carrier can use to ship orders. For more information about Carrier Services, see the "Configuring Cross Application Carrier Services" chapter.

## Logistics Management Configuration: Carrier Special Services

Carrier Special Services is used for defining codes that identify the different carrier special services a Carrier can use to ship and deliver orders. For more information about Carrier Special Services, see the "Configuring Cross Application Carrier Special Services" chapter.

## Logistics Management Configuration: Modification Reasons

Modification Reasons is used to define common codes for **modification reasons**. These codes define why a modification was made by a user. For more information about Modification Reasons, see the "Configuring a Document's Modification Reasons" chapter.

## Logistics Management Configuration: Load Execution Process Model

You can define a load document's business process workflow by creating process type pipelines. A **process type pipeline** is a series of transactions and statuses that guide a load document through a predefined process. A pipeline consists of the different statuses a document goes through during a load's planning. You can also set up transactions consisting of events, actions, and conditions, as they pertain to the pipeline you are configuring. For more information about Load Execution Process Model, see the "Configuring a Load Document's Pipeline" chapter.

## Logistics Management Configuration: Instruction Types

Instruction Types is used to define the common codes used when adding special instructions to a load document. For more information about Instruction Types, see the "Configuring a Document's Instruction Types" chapter.

## Logistics Management Configuration: Charge Categories

Charge Categories is used to define **charge definitions** that you can associate with orders and invoices through charge categories. These categories contain a group of related charge names that can be used when the particular category is used. For more information about Charge Categories, see the "Configuring a Load Document's Charge Categories" chapter.

## Logistics Management Configuration: Purge Criteria

Purge Criteria is used to define the parameters used when purging load document related records from the system. For more information about Purge Criteria, see the "Configuring a Load Document's Purge Criteria" chapter.

## Logistics Management Configuration: Load Type

Load Type is used to define codes for load types that appear on a load document. For more information about Load Type, see the "Configuring a Load Document's Load Types" chapter.

## Logistics Management Configuration: Stop Type

Stop Type is used to modify codes for stop types that appear on a load document. For more information about Stop Types, see the "Configuring a Load Document's Stop Types" chapter.

## Logistics Management Configuration: Modification Rules

Modification Rules is used to define rules that determine when modifications can be made during a load document's life cycle. For more information about Modification Rules, see the "Configuring a Load Document's Modification Rules" chapter.

## Logistics Management Configuration: Logistics Components

Logistics Components is used to define the different logistics related functionality throughout the business application module. For more information about Logistics Components, see the "Configuring Logistics Components" chapter.

---

## Icons Used in This Guide

The following table describes the icons used in this guide.

*Table 1. Icons Used in This Guide*












Icon	Name	Description
	Create New	Click this icon to create new information or details for components used to define delivery plans. As shown, you may see this icon with different backgrounds, depending on the context within which you are working.
	Save	Click this icon to save entries you have made. The Save icon may be displayed in blue or gold.

Table 1. Icons Used in This Guide (continued)

Icon	Name	Description
	<b>Details</b>	Click this icon to view or change details for components used to define delivery plans. As shown, you may see this icon with different backgrounds, depending on the context within which you are working.
	<b>Delete</b>	Click this icon to delete components used to define delivery plan. As shown, you may see this icon with different backgrounds, depending on the context within which you are working.
	<b>Transactions</b>	Click this icon to view a list of transactions in the Transactions tab window.
	<b>Statuses</b>	Click this icon to view a list of statuses in the Statuses tab window.
	<b>Conditions</b>	Click this icon to view a list of conditions in the Conditions tab window.
	<b>Actions</b>	Click this icon to view a list of actions in the Actions tab window.
	<b>Service Definitions</b>	Click this icon to view a list of service definitions in the Service Definitions tab window.
	<b>Right Arrow</b>	Click this icon to move data from one panel to another.
	<b>Left Arrow</b>	Click this icon to move data from one panel to another.



---

## Chapter 2. Navigating the Applications Manager

---

### Starting the Applications Manager

#### About this task

To access the Applications Manager:

#### Procedure

1. Point your browser to `http://<hostname>:<portname>/smcfs/console/start.jsp` where,
  - `hostname` is the computer name or IP address of the computer where Sterling Selling and Fulfillment Foundation is installed.
  - `portnumber` is the listening port of the computer where Sterling Selling and Fulfillment Foundation is installed.

The browser displays the Sign In window.

2. Enter your login ID and password and choose the Sign In button. The Console Home Page is displayed.
3. From the menu bar, choose Configuration > Launch Applications Manager. The Applications Manager opens in a new window. Additionally, enterprise users who maintain an enterprise can access the Applications Manager by means of `http://<Sterling Selling and Fulfillment Foundation installation server>/smcfs/console/login.jsp`. If both the Applications Manager and the monitor in the System Management Console are opened at the same time, and if a dialogue window is opened in either application, the other stops responding to user input until that dialogue window is closed. This is due to a bug in the Java platform.

---

### The Applications Manager Layout

The Applications Manager is a graphical user interface that can be used to configure different aspects of Sterling Selling and Fulfillment Foundation. The different configurations are defined by logical groupings called applications that can be accessed from the Applications Manager menu bar.

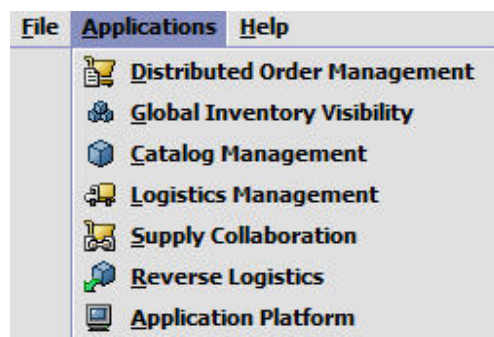


Figure 1. Applications Menu

Each application focuses on a particular aspect of Sterling Selling and Fulfillment Foundation and contains all of the rules, common codes, and settings necessary for Sterling Selling and Fulfillment Foundation to work in a real-world business setting.

The following applications can be configured in this version of Sterling Selling and Fulfillment Foundation:

- Distributed Order Management
- Global Inventory Visibility
- Catalog Management
- Logistics Management
- Supply Collaboration
- Reverse Logistics
- Application Platform

When you select the application that you want to configure, the Applications Manager displays a side panel containing all of the available configuration rules for the selected application and a work area in which these rules can be configured.

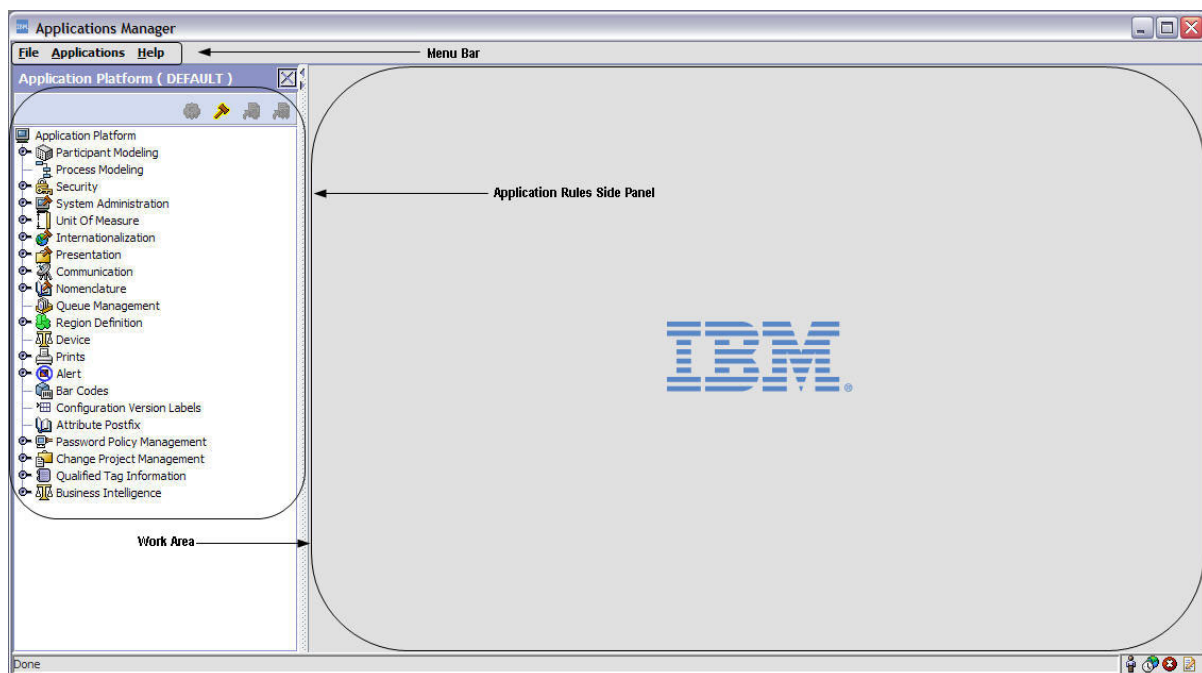


Figure 2. The Standard Applications Manager Interface

## Application Rules Side Panel

The application rules side panel displays a hierarchical tree of elements specific to processes used within the application.



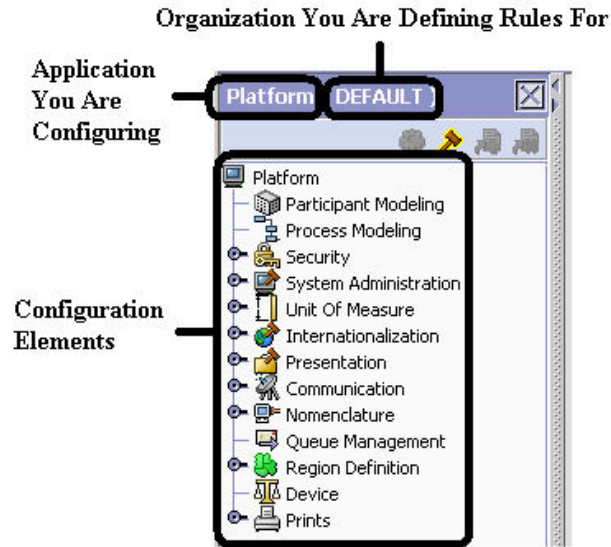


Figure 3. Example of Application Rules Side Panel

The application rules side panel also identifies the organization you are configuring rules for and what, if any, rules are inherited from another organization.

You can use the application rules side panel for accessing configuration screens, determining inheritance, and loading another organization's rules.

### Accessing Configuration Screens

The main purpose of the application rules side panel is to provide an interface to access the application's individual configuration screens. To access a configuration screen, browse through the application tree and double-click on the applicable configuration element, the element's configuration screen displays in the work area.

### Determining Inheritance

In Sterling Selling and Fulfillment Foundation, when an Enterprise is created it can inherit all or part of an existing Enterprise's configuration rules. This inheritance is done at the configuration group level. A configuration group is a classification of similar configuration elements. For example, all of the rules and configurations dealing with items are grouped together into one configuration group and all of the rules and configurations dealing with organizations are grouped into another.

An administrator organization is set for every organization defined within the system. Only the administrator organization can modify the rules defined for a particular organization. If a particular organization administers multiple organizations, then they can load the rules of organization that it administers within the application tree. For more information about loading another organization's rules, see "Loading Another Organization's Rules" on page 14.

Configuration groups are associated with organization levels. Organization levels determine how configuration groups are inherited and which organizations can maintain them. The organization levels defined in Sterling Selling and Fulfillment Foundation are:

- Hub Level - Configuration groups that are associated with the Hub organization

- Enterprise Level - Configuration groups that are associated with the individual Enterprise organizations within the Hub environment
- Catalog Organization - Configuration groups that are associated with the organization(s) that maintains the catalog(s) within the Hub environment
- Inventory Organization - Configuration groups that are associated with the organization(s) that maintains the inventory within the Hub environment
- Pricing Organization - Configuration groups that are associated with the organization(s) that maintains the pricing within the Hub environment
- Organization - Configuration groups that are associated with any organization within the Hub environment

The Applications Manager does not load configuration data and permissions based on Data Access Policies that are described in the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

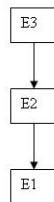
### Enhanced Inheritance for Process Models

An Enterprise can inherit the configurations of the following entities from other Enterprises:

- Pipelines
- User Exits
- Services
- Actions
- Conditions
- Statuses
- Transactions
- Events

When an Enterprise inherits these entities from some other Enterprise, the current Enterprise can view the configurations that are inherited from all other Enterprises (including the Hub) in the inheritance hierarchy. In addition, the current Enterprise can view the configurations that are defined for the Hub.

For example, consider the following inheritance hierarchy:



In this hierarchy, Enterprise E1 is inheriting from Enterprise E2, which in turn is inheriting from Enterprise E3. Enterprise E1 can view the configurations that are defined for Enterprise E2 and Enterprise E3. In addition, Enterprise E1 can view the configurations that are defined for the Hub.

## Organization Level Rules

The following table details the rules used to determine which organizations can maintain a configuration group as defined by the organization level. The table also describes the rules that determine how configuration groups are inherited when an organization is created.

Table 2. Organization Level Rules

Organization Level	Organizations That Can Modify at this Level...	Inheritance Details
Hub Level	Only the Hub organization can modify configuration groups at the Hub level. All other organizations have read-only access.	All organizations share this information.
Enterprise Level	Only Enterprise organizations can modify configuration groups at the Enterprise level.  Any business transaction requiring Enterprise configuration is picked up from the Enterprise established by the transactional context. For example, order documents have a specific Enterprise.	An Enterprise can inherit this configuration from another Enterprise. Additionally, this configuration can be overridden at a configuration group level.
Catalog Organization	Organizations that are designated as catalog organizations can modify configuration groups at the catalog organization level.	None.
Inventory Organization	Organizations that are designated as inventory organizations can modify configuration groups at the inventory organization level.	None.
Pricing Organizations	Organizations that are designated as pricing organizations can modify configuration groups at the pricing organization level.	None.
Organization	Any organization assigned a role (Seller, Buyer, etc.) can modify configuration groups at the organization level.	None.

You cannot inherit from an Enterprise that does not have the same inventory, capacity, and catalog organizations as the organization you are configuring.

## Applications Rule Side Panel

The application rules side panel displays rules that have been inherited as grayed out.

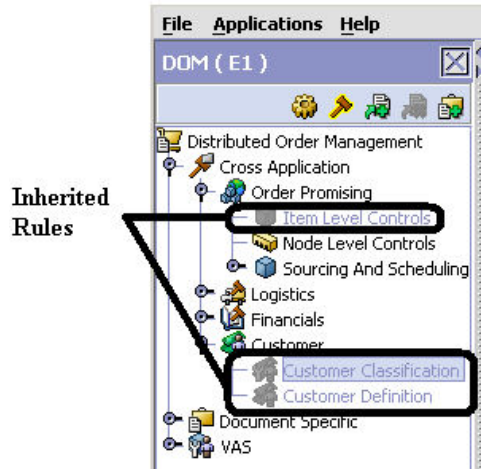


Figure 4. Inherited Rules in the Application Rules Side Panel

As stated in the table above, depending on the organization you are logged in as, you may be able to override some inherited rules. If a rule can be overridden, the Override Configuration icon becomes available in the application rule side panel when you highlight the rule.

#### Override Configuration Icon is Available

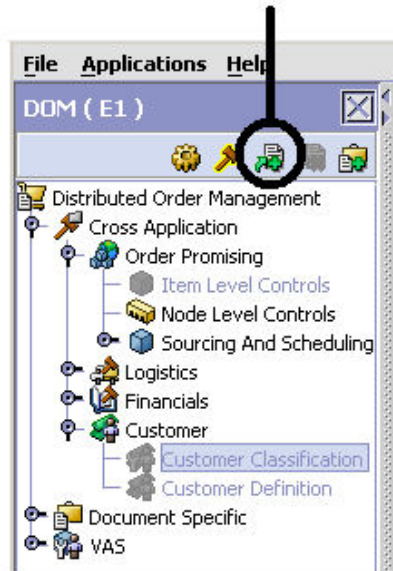


Figure 5. Override Configuration Icon

When you choose to override a rule you also override any other rules in the configuration group the rule you are overriding is associated with. When you choose the Override Configuration icon the Configuration Override Details pop-up window displays. This window provides the list of rules that are overridden.

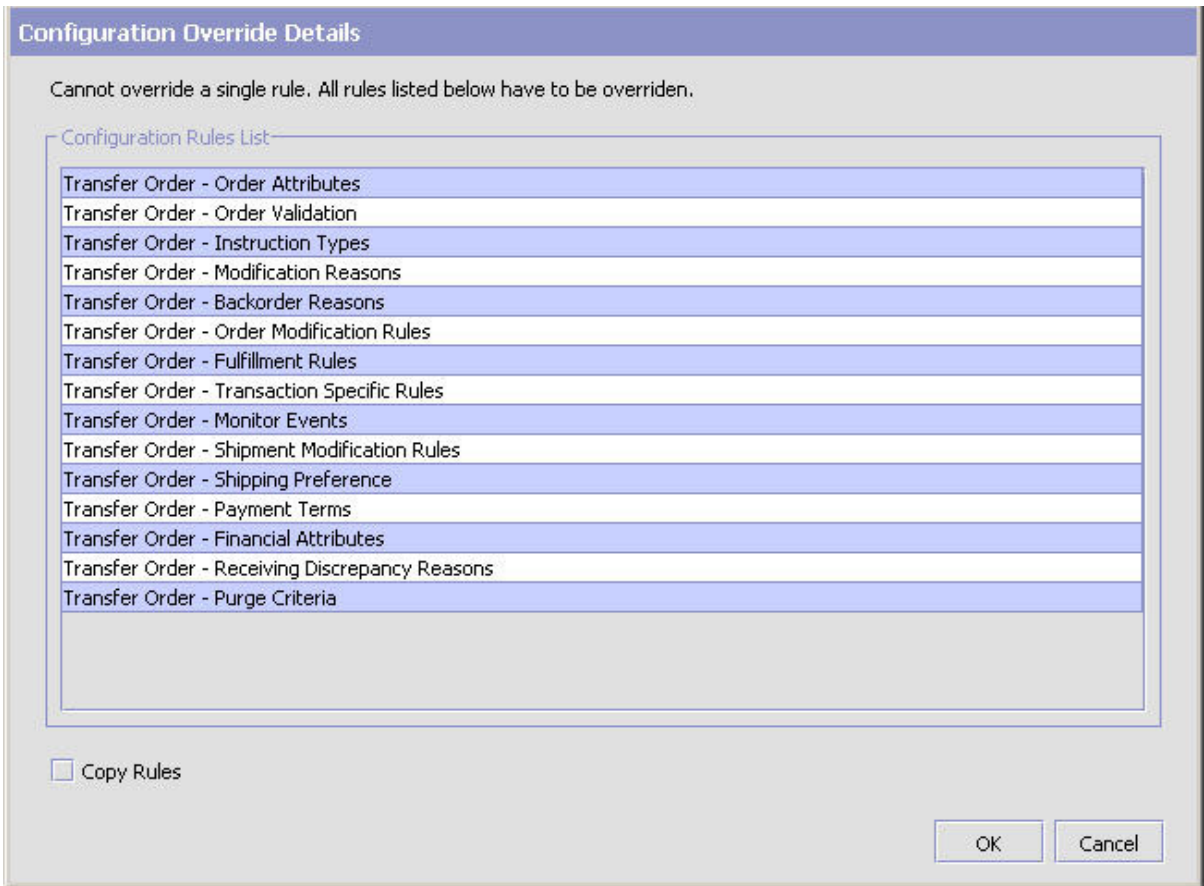


Figure 6. Example of Configuration Override Details Pop-Up Window

### Overriding a Configuration Group

If you override a configuration group and then decide to "re-inherit" the original rules, you can choose the Give Back Configuration Ownership icon. This icon becomes available in the application rules side panel for rules that have been overridden.

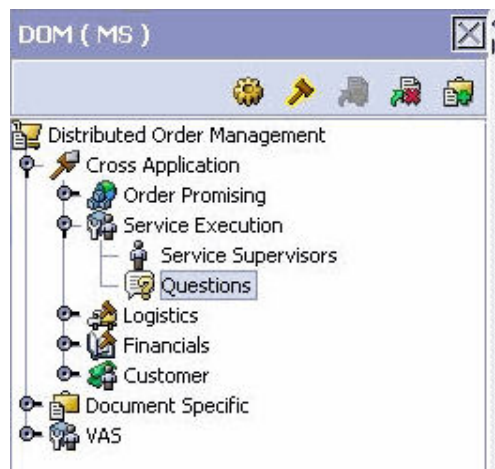


Figure 7. Give Back Configuration Ownership Icon

When you select the Give Back Configuration Ownership Icon, the Configuration Override Details pop-up window displays. This window provides the list of rules that are re-inherited.

**Note:** If you select the Delete Rules field on the Configuration Override Details pop-up window, you give back rule ownership to the organization you originally inherited from, but you do not retain any of the rules that you inherited from them. If you do not select this field, you give back rule ownership to the organization you originally inherited from, but you retain the rules that you inherited from them.

## **Loading Another Organization's Rules**

### **About this task**

An administrator organization is set for every organization defined within the system. Only the administrator organization can modify the rules defined for a particular organization. If a particular organization administers multiple organizations, then they can load the rules of organization that it administers within the application tree. See Table 2 on page 11 for the rules that determine which organizations you can administer.

The rules that are available from the tree in the application rules side panel may vary depending on the type of organization you select and the roles it has been assigned.

To load another organization's rules:

### **Procedure**

1. From the applicable application rules side panel, choose the **Load Rules for Organization** icon. The Load Organizations for Configuration pop-up window displays.
2. From Organization, select the organization that you want to work with.
3. Choose OK. The organization's rules display in the application rules side panel.

### **Results**

The application rules side panel displays the organization you are working with in parentheses.

## **Work Area**

The work area is the main area in which different configuration screens appear. The main types of screens that you can see in the work area are the Search, List, Details, and Drag and Drop windows.

### **Search Window**

A search window provides you with a means to perform a filtered search. The upper panel of a search window offers criteria applicable to the entity you are searching through which you can narrow your search. The lower panel lists the results of a search once it has been performed.

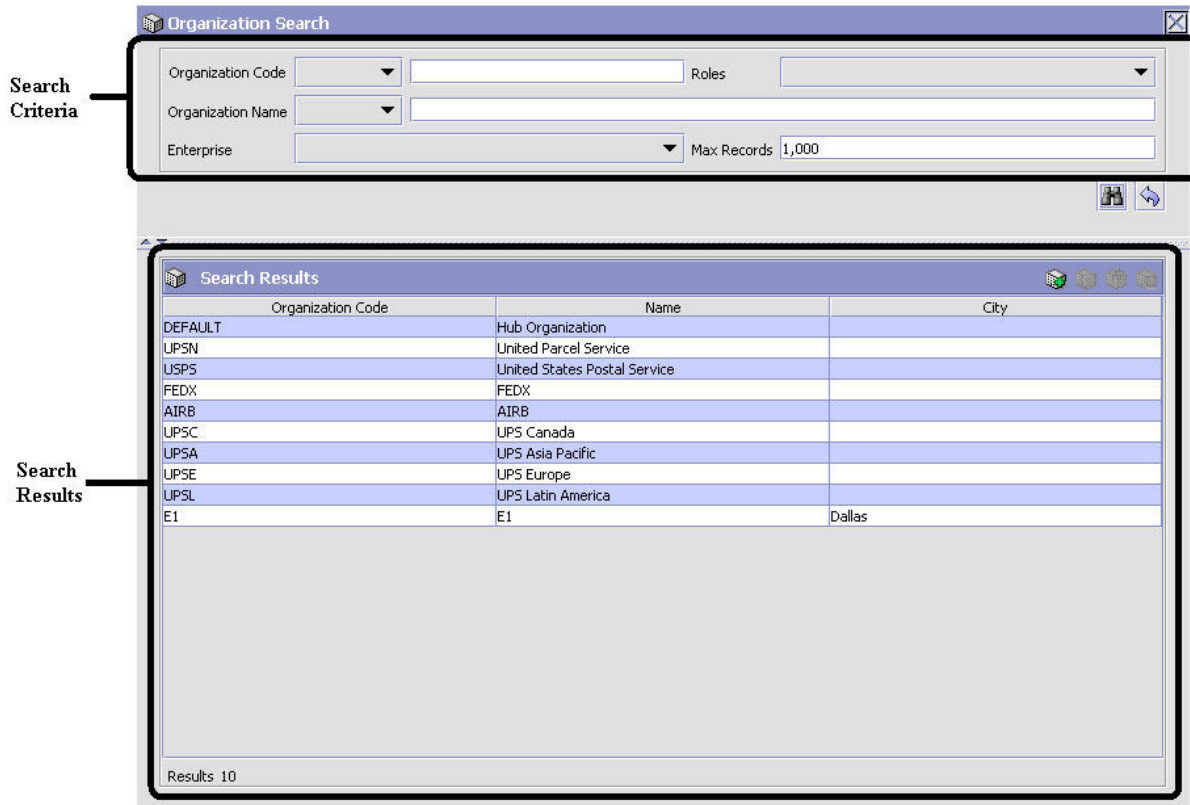


Figure 8. Search Window Example

## List Window

When you choose to configure a specific rule or code that does not require a search, the Applications Manager may display a basic list window of the rules and codes that have previously been configured.

Dimension UOMs ( DEFAULT )	
UOM Code	UOM Description
CM	Centimeter
FEET	Feet
IN	Inch
KM	Kilometer
METER	Meter
MILE	Mile

Results 6 Of 6

Figure 9. List Window Example

## Details Window

A details window is the main interface through which a bulk of the configuration is done. A details window can contain editable fields and tables, tabs to configure different aspects of an entity, and additional actions that can be performed on an entity.



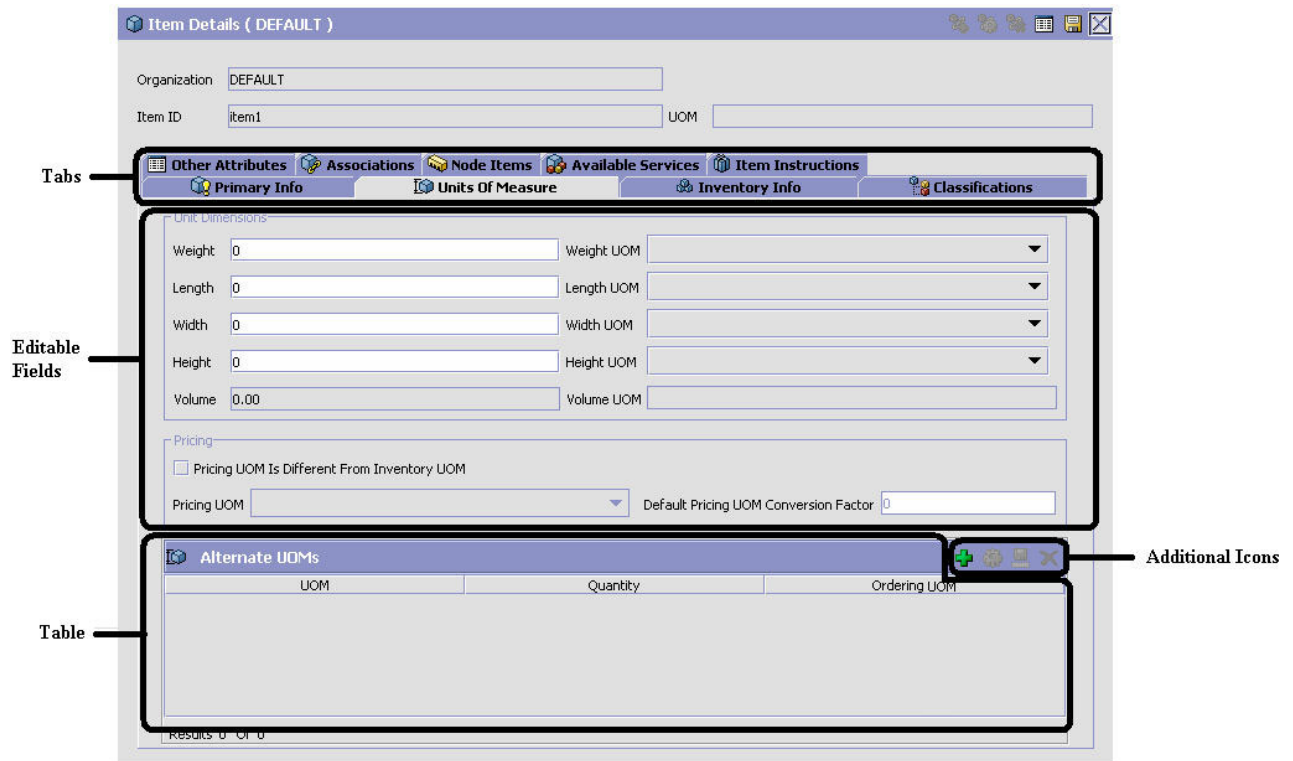


Figure 10. Details Window Example

## Drag and Drop Window

You can use a graphical drag and drop window to ease the construction of pipelines, pipeline determination, event handlers, status monitoring rules, and services. A drag and drop window consists of a pallet and a graphical work area.

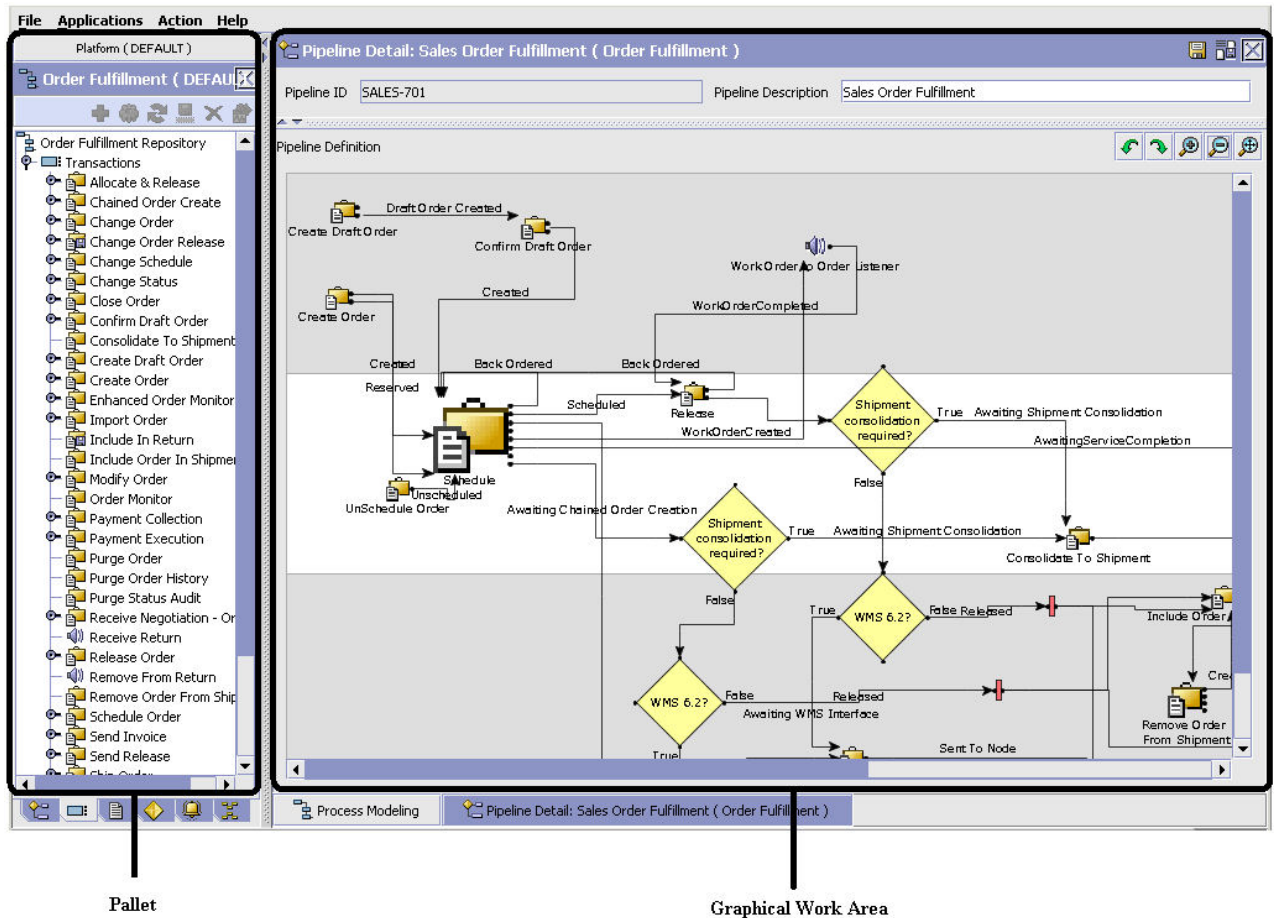


Figure 11. Drag and Drop Window Example

To begin building any of these entities, choose a component, such as a transaction, from the pallet. Drag the component into the graphical work area. The transaction is now displays as a graphical representation of itself.

To connect one component to another, you must drag the mouse from the outgoing port of a component until it forms a connecting line with the incoming port of another component. The links between components can be set up either horizontally or vertically.

To delete components or links, right-click on the component and choose Delete. Once components and links have been established you can move them around by dragging them, the links redraw themselves according to the new position. If you press and hold the CTRL key while dragging a component, the component is copied within the graphical work area.

## Actions Available in the Applications Manager

Through the Applications Manager, you can use the lookup functionality, view logged in users, use lists and list filtering, troubleshoot errors, and use special characters.

## Using the Applications Manager's Lookup Functionality

Throughout the Applications Manager there are many fields that have a lookup functionality to find or create additional records as they pertain to that field. For example, on the Primary Info tab of the Organization Details screen, the Locale field has a lookup functionality to create a new locale from that screen. When you choose the Create New lookup button the Locale Details information displays in a pop-up screen for you to modify.



*Figure 12. Lookup Icon Example*

The information that displays in a lookup field varies depending on how many records you have pertaining to that particular field. When there are 20 or less records, the lookup displays as a drop-down list with a Create New button. When there are between 21 and 75 records, the lookup displays as a drop-down list with a Search button.

When there are more than 75 records, the lookup displays as a text box with a Search button. You can type the value in the text box or search for the value using the Search button. If you enter a value, it is validated when it is saved. You should always type the value as it would appear if it was displayed as a drop-down list. For example, for a currency lookup, you should type the currency description in the text box even though the currency code is saved in the table. An error displays on save if the user has entered an invalid value.

When you use a lookup for a particular field in the Applications Manager, you should refer to the corresponding section in this guide to set up the particular information.

## Viewing the Document Types Associated with an Application

In the Distributed Order Management, Supply Collaboration, Reverse Logistics, and Logistic Management configuration applications, you can view all of the document types associated with the application. Sales Order, Transfer Order, Master Order, Quote, and Purchase Order are all examples of document types.

To view an application's associated document types, open the applicable application from the menu and choose the **Create New** icon from the application rules side panel. The Associated Document Types window displays displaying a list of all of the document types associated with the application you are working in.

Document Type	Description
0004	Template Order
0006	Transfer Order
0001	Sales Order
0007	Master Order
0015	Quote

Results 5 of 5

Figure 13. Associated Document Types Window

## Adding a Document Type to an Application

### About this task

You can add a document type that is associated with another application to the application you are currently working in.

An added document type's associated screens may be irrelevant to the application you are associating it with.

To add a document type to an application:

### Procedure

1. From the Associated Document Types window, choose the **Create New** icon. The Associated Document Type pop-up window displays.
2. From Document Type, select the document type that you want to associate with the application.
3. Select Enable Access To This Document Through This Application's Console.
4. Choose the **Save** icon.

## Viewing the User Logged into the Applications Manager

### About this task

You can view the user logged into the Applications Manager and their locale at any time. To view this information, move your mouse over the User icon and Locale icons in the bottom right-hand corner of the application to display the tool tips.

## Using Lists and List Filtering

### About this task

When viewing any list in the Applications Manager, it is possible to filter the contents of the list based in criteria that you define. Filtering is accomplished by right-clicking anywhere on the list's column headings and using the Table Filter Editor associated with the list.

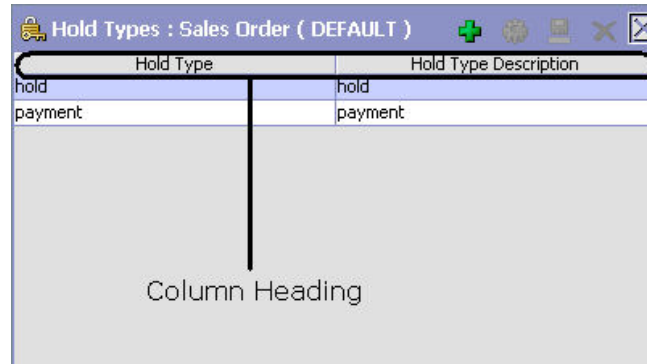


Figure 14. Column Headings in a List

Table 3. Table Filter Editor Window

Field	Description
<b>Apply To Existing Records</b>	Checking this box applies a new filter set of results that have been previously filtered instead of the whole set.
<b>Max Records</b>	Specify the maximum number of records that are to be returned from a filter. The default number is 100
<b>Dynamic Fields</b>	Fields such as "Hold Type" and "Hold Type Description" are dynamically populated based on the list you are currently viewing.  These fields can be searched using text strings combined with criteria such as <b>Is</b> , <b>Starts With</b> , or <b>Contains</b> .

Search strings are case sensitive. For example, "Item" does not return the same values as "item".

## Date and Time Entry

Date fields through the Applications Manager have a calendar icon that can be used to find dates as it pertains to that field. When you click on this icon, a small calendar displays. You can navigate through this calendar to determine the appropriate date. For example, on the Create Calendar window, the Default Effective To field has a calendar icon that you can use to verify the appropriate ship by date to populate the field.



Figure 15. Calendar Icon example

You can also enter time of day information throughout the Applications Manager. To do this, double click on the time field, and enter the time of day.

Shift Name	Start Time	End Time
	<input type="text"/>	

Figure 16. Time Field example

Time should be entered in a 24 hour time format everywhere throughout the Applications Manager .

## Troubleshooting Errors

### About this task

You can view the description and cause of any error raised in Sterling Selling and Fulfillment Foundation, as well as the actions to troubleshoot it.

To view the Sterling Selling and Fulfillment Foundation system error descriptions:

### Procedure

1. From the menu bar, choose Help > Troubleshooting. The Error Search window displays.
2. Enter the applicable search criteria and choose the **Search** icon. A list of error codes and their descriptions display.
3. Choose the **Details** icon to view the cause of the error and action to troubleshoot it.

## Using Special Characters

Throughout the Applications Manager there may be instances where you need to use special characters in data entry. For information about the use of special characters in Sterling Selling and Fulfillment Foundation, see the *Sterling Selling and Fulfillment Foundation: Customization Basics*.

---

## Chapter 3. Configuring Cross Application Carrier Services

---

### Configuring Cross Application Carrier Services

You can set up codes to identify the different carrier services a Carrier uses to ship orders.

The following are examples of different carrier service codes:

- Standard Mail
- 2nd Day Air
- Ground

---

### Creating a Carrier Service Code

#### About this task

To create a carrier service code:

#### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Carrier Services. The Carrier Services window displays in the work area.
2. Choose the **Create New** icon. The Carrier Service Details pop-up window displays.
3. Enter information in the applicable fields. Refer to Table 4 for field value descriptions.
4. Choose the **Save** icon.

Table 4. Carrier Service Details Pop-Up Window

Field	Description
Carrier Service Code	Enter the name of the carrier service.
Carrier Service Description	Enter a brief description of the carrier service.
Minimum Transit Days	Enter the minimum number of days that a carrier service could take to deliver a shipment. For example, Ground service could have a minimum of 3 transit days.
Distance Per Day	Enter the maximum distance that the service travels each transit day. <b>Note:</b> This number is used for order line scheduling. This value is only used if the Use Advanced Transit Time Calculations flag on the Other Rules tab under Distributed Order Management > Cross Application > Logistics > Logistics Attributes is selected. For more information about this field, see the <i>Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide</i> .
Maximum Transit Days	Enter the maximum number of days that a carrier service allows for delivery. For example, Ground service has a maximum of 5 transit days.
Carrier Type	Select a carrier type from the drop-down list.
Used For Ordering	Check this box to enable the carrier service to be selected during order creation.

Table 4. Carrier Service Details Pop-Up Window (continued)

Field	Description
Allow Up To <> Days Of Non-Transit Time For An Item	Enter the maximum number of days a carrier service holds deliveries that can't be delivered on a scheduled delivery day. If no value is entered, the carrier holds deliveries for an unlimited number of days. Use this option to calculate shipment dates. See the <i>Sterling Selling and Fulfillment Foundation: Product Concepts Guide</i> guide for more information.
Delivery Schedule List	Displays the delivery schedules for the selected carrier. You can add, modify, and delete schedules in this list.

## Modifying a Carrier Service Code

### About this task

To modify a carrier service code:

#### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Carrier Services. The Carrier Services window displays in the work area.
2. Select the applicable carrier service code and choose the **Details** icon. The Carrier Service Details pop-up window for the carrier displays.
3. Enter information in the applicable fields. Refer to Table 4 on page 23 for field value descriptions.
4. Choose the **Save** icon.

## Deleting a Carrier Service Code

### About this task

To delete a carrier service code:

#### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Carrier Services. The Carrier Services window displays in the work area.
2. Select the applicable carrier service code and choose the **Delete** icon.

---

## Creating a Carrier Delivery Schedule

### About this task

Sterling Selling and Fulfillment Foundation lets you define an unlimited number of delivery schedules for a carrier. You can set up schedules for specific time periods, such as months or seasons, and configure overrides in the schedule for days, such as holidays.

For example, you can configure a holiday schedule in which the carrier transports shipments Sunday through Saturday and makes deliveries Monday through Saturday. You can configure exceptions to the schedule, such as changing a delivery on Dec. 24 from 6 to 4 p.m.

To create a delivery schedule:



## Procedure

1. From the tree in the application rules side panel, choose Cross Application > Carrier Services. The Carrier Services window displays in the work area.
2. Select the applicable carrier service code and choose the **Details** icon. The Carrier Service Details popup window for the carrier displays.
3. Select the **Create New** icon on the Delivery Schedule List window. The Delivery Schedule Details window displays.
4. Enter information in the applicable fields. Refer to Table 5 for field value descriptions.
5. Choose the **Save** icon.

Table 5. Delivery Schedule Details Window

Field	Description
Effective From Date	Select the date that this carrier schedule is applicable from.
Effective To Date	Select the date that this carrier schedule is applicable to.  Ensure that you specify a date range that is not already in use by another delivery schedule. For example, if a delivery schedule has a date range of January through June, you cannot specify a date range of May through July for a different delivery schedule.
Delivers On	Check the days of the week when this carrier delivers shipments.
Transfers On	Check the days of the week when this carrier transports shipments.
Delivery Date Overrides	
Override Date	Select the date for which you want to override the carrier's delivery/transport schedule.
Can Transfer	Select Yes to allow transfer or No to disallow transfer on the specified override date.
Can Deliver	Select Yes to allow delivery or No to disallow delivery on the specified override date.

## Modifying a Carrier Delivery Schedule

### About this task

To modify a carrier delivery schedule :

### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Carrier Services. The Carrier Services window displays in the work area.
2. Select the applicable carrier service code and choose the **Details** icon. The Carrier Service Details popup window for the carrier displays.
3. In the Delivery Schedule List, select the applicable delivery schedule and choose the **Details** icon. The Delivery Schedule Details window displays.
4. Enter information in the applicable fields. Refer to Table 5 for field value descriptions.
5. Choose the **Save** icon.



---

## Chapter 4. Configuring Cross Application Carrier Special Services

---

### Configuring Cross Application Carrier Special Services

You can set up codes to identify the different carrier special services that a Carrier uses while shipping and delivering orders.

The following are examples for carrier special service codes:

- Adult Signature Required
- Cash on Delivery
- COD
- Contract Print Return Label
- Declared Value Insurance
- Delivery Confirmation
- Hazardous Material
- Hold For PickUp
- Hundred Weight
- Online Call Tag
- Saturday Delivery
- Saturday PickUp
- Ship Alert
- Ship Notification
- Signature Required
- Sunday Delivery
- Verbal Confirmation

---

### Creating a Carrier Special Service Code

#### About this task

To create a carrier special service code:

#### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Carrier Special Services. The Carrier Special Services window displays in the work area.
2. Enter information in the applicable fields. Refer to Table 6 for field value descriptions.
3. Choose the **Save** icon.

*Table 6. Carrier Special Services Window*

Field	Description
Carrier Special Services Code	Enter the name of the carrier special service.
Description	Enter a brief description of the carrier special service.

---

## Modifying a Carrier Special Service Code

### About this task

To modify a carrier special service code:

### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Carrier Special Services. The Carrier Special Services window displays in the work area.
2. Modify information in the applicable fields. Refer to Table 6 on page 27 for field value descriptions.
3. Choose the **Save** icon.

---

## Deleting a Carrier Special Service Code

### About this task

To delete a carrier special service code:

### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Carrier Special Services. The Carrier Special Services window displays in the work area.
2. Select the applicable carrier special service code and choose the **Delete** icon.

---

## Chapter 5. Configuring Cross Application Logistics Rules

---

### Configuring Cross Application Logistics Rules

Any transaction between the buyer and the seller is expected to follow a sequence like creating a shipment, load, and transit updates. Most often this sequence is not followed and the buyer receives the information in a more random sequence. Often, without having an existing shipment, a shipment cannot be added to a load. But, by enabling this logistics rule, some unique attributes of the shipment can be added to a load even without an existing shipment, such as:

- ShipmentNo, ShipNode, and SellerOrganizationCode
- BOLNo and ShipNode
- PRONo and ShipNode

The getLoadDetails API returns these shipment attributes from the load shipment.

---

### Creating a Logistics Rule

#### About this task

To create a logistics rule:

#### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Logistics Rules. The Logistics Rules window displays in the work area.
2. Enter information in the applicable fields. Refer to Table 7 for field value descriptions.

Table 7. Logistics Rule

Field	Description
Allow Addition of Shipments not Available on System to a Load	Check this box to allow the creation of a load with shipments even when shipments do not exist in the system. When a new shipment is created in the system, it checks if this shipment is previously associated with any load and updates the load shipment record.



---

## Chapter 6. Configuring Transportation

---

### Configuring Transportation

You can record transportation or in-transit updates for a container. These updates are recorded against activities defined in Sterling Selling and Fulfillment Foundation as transportation activities.

---

### Creating an Activity Code

#### About this task

To create an activity code:

#### Procedure

1. From the tree in the application rules side panel, choose Transportation > Activities. The Activity Search window displays.
2. Enter information in the applicable fields. Refer to Table 8 for field value descriptions.
3. Choose the **Save** icon.

*Table 8. Activity Search Window*

Field	Description
Activity Group	This field is automatically populated by the system as 'Transportation'.
Activity List	
Activity Code	Enter a name for the activity code.  This Activity Code is the unique identity of the activity.
Description	Enter a brief description for the activity code.

---

### Modifying an Activity Code

#### About this task

To modify an activity code:

#### Procedure

1. From the tree in the application rules side panel, choose Transportation > Activities. The Activity Search window displays, with the list of activities.
2. Enter information in the applicable fields. Refer to Table 8 for field value descriptions.
3. Choose the **Save** icon.

**Important:** Do not modify the list of activity codes provided with the product.

---

## Deleting an Activity Code

### About this task

To delete an activity code:

### Procedure

1. From the tree in the application rules side panel, choose Transportation > Activities. The Activity Search window displays, with the list of activities.
2. Choose the Activity Code to be deleted.
3. Choose the **Delete** icon.

**Important:** Do not modify the list of activity codes provided with the product.



---

## Chapter 7. Configuring a Document's Modification Reasons

---

### Configuring a Document's Modification Reasons

You can define common codes for **modification reasons**. These codes define why a modification was made by a user in the Application Consoles.

In addition to modification reasons, the codes that you define are used as hold reasons when you put an order on hold in the Application Consoles.

You can use the Modification Reasons branch for creating, modifying, and deleting a modification reason.

---

### Creating a Modification Reason

#### About this task

To create a modification reason:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Modification Reasons. The Modification Reasons window displays in the work area.
2. Choose the **Create New** icon. The Modification Reason Details pop-up window displays.
3. In Modification Reason, enter the modification reason.
4. In Short Description, enter a brief description of the modification reason.
5. In Long Description, enter a more detailed description of the modification reason.
6. If this modification reason requires that the order be repriced due to a reduced quantity, check the Re-Price Order With Reduced Quantity checkbox.

This flag is applicable only if this modification reason is used for cancellations, where repricing needs to occur against a reduced quantity: the quantity against which the order line is repriced (repricing quantity) is adjusted to the reduced quantity. For more information about repricing quantity, see the *Sterling Selling and Fulfillment Foundation: Javadocs*.

If this modification reason is used for a modification that does not reduce quantity, this flag is not applicable.

This field does not exist for Load Modification Reasons.

7. Choose the **Save** icon.

---

### Modifying a Modification Reason

#### About this task

To modify a modification reason:

## Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Modification Reasons. The Modification Reasons window displays in the work area.
2. Select the applicable modification reason and choose the **Details** icon. The Modification Reason Details pop-up window displays.
3. In Short Description, enter a brief description of the modification reason.
4. In Long Description, enter a more detailed description of the modification reason.
5. If this modification reason requires that the order be repriced due to a reduced quantity, check the Re-Price Order With Reduced Quantity checkbox.  
This flag is applicable only if this modification reason is used for cancellations, where repricing needs to occur against a reduced quantity: the quantity against which the order line is repriced (repricing quantity) is adjusted to the reduced quantity. For more information about repricing quantity, see the *Sterling Selling and Fulfillment Foundation: Javadocs*.  
If this modification reason is used for a modification that does not reduce quantity, this flag is not applicable.  
This field does not exist for Load Modification Reasons.
6. Choose the **Save** icon.

---

## Deleting a Modification Reason

### About this task

To delete a modification reason:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Modification Reasons. The Modification Reasons window displays in the work area.
2. Select the applicable modification reason and choose the **Delete** icon.

---

## Chapter 8. Configuring a Load Document's Hold Types

---

### Configuring a Load Document's Hold Types

You can configure hold types that are applied on a load document. A load can be put on hold either manually or automatically by applying a particular hold type to it.

---

### Load Document: Creating a Hold Type

#### About this task

To create a hold type:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific (Load) > Load > Hold Types. The Hold Types window displays in the work area.
2. Click the **Create New** icon. The Hold Type pop-up window displays.
3. In Hold Type, enter the hold type.
4. In Hold Type Description, enter the description of the hold.
5. Enter information in the applicable fields. See Table 9, Table 10 on page 36, and Table 11 on page 37 for field value descriptions.
6. Click the **Save** icon.

Table 9. Hold Type Pop-Up Window, Hold Creation Tab

Field	Description
<b>Hold Created Automatically</b>	
On Load Creation	Check this box to apply this hold type to all loads after creating the load.
On Resolution Of The Hold Type	Check this box to apply a hold type upon resolution of another hold type. From the drop-down list, select this hold type. <b>Note:</b> Sterling Selling and Fulfillment Foundation does not check whether or not you are defining a circular hold type definition. For example, if you define hold type B as being applied upon resolution of hold type A, and hold type A as being applied upon resolution of hold type B, you could create an infinite loop that Sterling Selling and Fulfillment Foundation does not warn you against.
When The Following Modifications Are Performed	Check this box to automatically apply a hold type to a load for certain modification types.  Click the <b>Change Modification Type List</b> icon to modify the list. In the Modification Type List pop-up window: <ul style="list-style-type: none"><li>• Use the right arrow to move the available modification types you want to associate with the hold type to the subscribed list.</li><li>• Use the left arrow to unsubscribe the modification types you want to disassociate with the hold type and move them back under the available list.</li></ul>

Table 9. Hold Type Pop-Up Window, Hold Creation Tab (continued)

Field	Description
For All Loads	Choose this option if you want to apply the above conditions to all loads. <b>Note:</b> You can choose this button only after saving the hold that you created.
Only For Loads Satisfying The Following Condition	Choose this option if you want to apply the above conditions to loads satisfying a certain condition.  Click the <b>Add/Modify Condition</b> icon to build or modify a condition that is evaluated. For more information about using the condition builder, see the <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i> .  You can extend the available attributes for a condition. For more information about extending the attributes, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i> . <b>Note:</b> You can choose this button after saving the hold type that you created.
<b>Hold Created Manually</b>	
By Any User	Choose this option if any user group can apply the hold to a load.
By Users Who Belong To The Following Groups	Choose this option if only users belonging to certain user groups can apply the hold to a load.  Click <b>Change User List</b> icon to modify the list. In the subsequent pop-up window: <ul style="list-style-type: none"> <li>• Use the right arrow to move the available user groups that you want to associate with a hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe user groups that you want to disassociate with a hold type and move them back under the available list.</li> </ul>

Table 10. Hold Type Pop-Up Window, Hold Resolution Tab

Field	Description
<b>Hold Resolved Automatically</b>	
The Following Time Triggered Transaction Will Process Created Holds	From the drop-down list, select the time-triggered transaction that processes created holds.
The Following Time Triggered Transaction Will Process Rejected Holds	From the drop-down list, select the time-triggered transaction that processes rejected holds.
<b>Hold Resolved Manually</b>	
Any User Can Process This Hold	Choose this option if any user group can process the hold.

Table 10. Hold Type Pop-Up Window, Hold Resolution Tab (continued)

Field	Description
Users Belonging To The Following User Groups Can Process This Hold	<p>Choose this option if only users belonging to certain user groups can process this hold.</p> <p>Click the <b>Change User List</b> icon to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available user groups that you want to associate with a hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the user groups that you want to disassociate with a hold type and move them back under the available list.</li> </ul>

Table 11. Hold Type Pop-Up Window, Hold Effects Tab

Field	Description
The Following Transactions Will Be Stopped From Processing Loads On This Hold	<p>Transactions that are disallowed when a hold type is applied to a load.</p> <p>Click the <b>Change Transaction List</b> icon to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available modification types that you want to associate with a hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the modification types that you want to disassociate with a hold type and move them back under the available list.</li> </ul>
The Following Modifications Are Not Allowed For Loads On This Hold	<p>Modification types that are disallowed when a hold type is applied to a load.</p> <p>Click the <b>Change Modification Type List</b> icon to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available transactions that you want to associate with a hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe transactions that you want to disassociate with a hold type and move them back under the available list.</li> </ul>

---

## Load Document: Modifying a Hold Type

### About this task

You can modify a hold type.

To modify a hold type:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific (Load) > Load > Hold Types. The Hold Types window displays in the work area.
2. Select the applicable hold type and click the **Details** icon. The Hold Type pop-up window displays.

3. Enter information in the applicable fields. See Table 9 on page 35, Table 10 on page 36 and Table 11 on page 37 for field value descriptions.
4. Click the **Save** icon.

---

## Load Document: Deleting a Hold Type

### About this task

You can delete a hold type.

To delete a hold type:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific (Load) > Load > Hold Types. The Hold Types window displays in the work area.
2. Select the applicable hold type and click the **Delete** icon.

---

## Chapter 9. Configuring a Load Document's Pipeline

---

### Configuring a Load Document's Pipeline

**Note:** Be aware that return fulfillment requires sourcing configuration. Sourcing configuration is accessible through the Distributed Order Management configuration grouping. For more information about configuring sourcing, see the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

To complete a load document's lifecycle, each document has a set of different processes that it can go through. These processes are called process types.

- Receipt

You can configure the rules and components specific to a load document's process type.

---

### Defining Process Type Details

You can define the parameters and templates that distinguish a process type.

For more information about defining process type details, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

---

### Process Type Pipeline Configuration

A **process type pipeline** is a series of transactions and statuses that guide document types, such as a Sales Order, through a predefined process. A pipeline consists of the different statuses a document goes through during fulfillment, negotiation, shipment, or receipt. You can also set up transactions consisting of events, actions, and conditions, as they pertain to the pipeline you are configuring.

#### Repositories

A repository is a logical collection of entities that define the business process workflow.

The following entities are included in a repository:

- Pipelines
- Transactions
- Statuses
- Conditions
- Actions
- Services

Sterling Selling and Fulfillment Foundation provides a base repository for each of the system-defined process types. Some of the entities within a repository are copied when creating a new document type. For more information about creating a new document type, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

The load process is modeled through a pipeline. This represents the process configuration that is unique to an organization. An organization may also specify unique processes for each participating Enterprise.

## Defining Pipeline Determination

**Pipeline determination** is used to set up conditions that affect which pipeline is used during the start of the business process workflow. For example, an organization deals with sales orders that sometimes contain hazardous materials. They have two separate pipelines, one in which orders with order lines without any hazardous materials go through and one in which orders with order lines containing hazardous materials must go through for inspection before continuing through the order process. The organization uses pipeline determination to set up a condition that determines whether or not order lines contain hazardous materials and sends the order line down the correct pipeline.

When you expand the Pipeline Determination branch, the components displayed depends on what role you are logged in as. If you are logged in as a Hub role, the Hub Rule displays. If you are logged in as an Enterprise role, both the Hub Rule and My Rule components display. Double-click on the applicable node to display the pipeline determination rules.

**Note:** If you are logged in as an Enterprise role, the Hub Rule screen is grayed out and cannot be modified.

Drag conditions and pipelines into the work area to construct pipeline determination rules. A single pipeline or condition must be the root. Conditions cannot link back to an earlier component in the chain and a pipeline cannot be linked to twice.

**Note:** When configuring pipeline determination for an order document type pipeline, please note that pipeline determination is only considered when adding a line or creating an order. When changes are made to draft orders pipeline determination does not occur.

### Condition Variables for Pipeline Determination

For a list of the condition variables that can be used for pipeline determination, refer to “Condition Builder Attributes” on page 263.

## Load Document: Pipelines

### About this task

For more information about configuring pipelines, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

To view the load execution pipeline details:

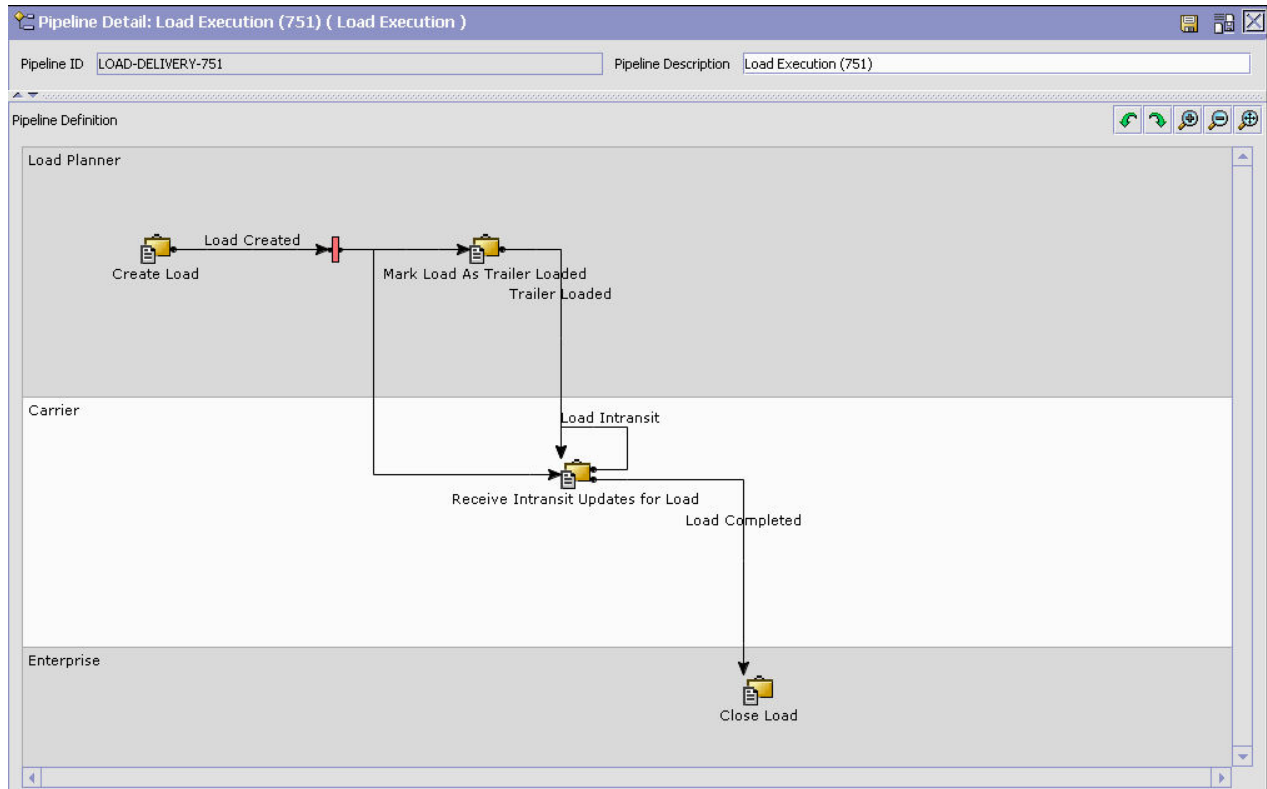
### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Load Execution Process Model. The Load Execution window displays.
2. In the Load Execution window, choose Load Execution Repository > Pipelines > Load Execution.
3. The Pipeline Detail: Load Execution (Load Execution) window displays.



## Results

For more information about creating and modifying a pipeline, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.



## Load Document: Transactions

### About this task

Every process type has a set of base transactions defined for it. A transaction is a logical unit of work that is necessary for performing an activity within Sterling Selling and Fulfillment Foundation. Base transactions are predefined transactions that contain information about how the transaction behaves, such as how many copies of a transaction can be kept in a process type and whether or not it can have configurable base pick and drop statuses. Base transactions can be used to create new transactions. These transactions can be changed within the limits defined in the base transaction.

For more information about transactions, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

To view the transaction details for a load execution pipeline:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Load Execution Process Model. The Load Execution window displays.
2. In the Load Execution window, choose the **Transactions** icon.

- The Transactions tab window displays. Refer to Table 12 for field value descriptions.

## Results

For more information about creating and modifying transactions, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

Table 12. Load Execution Pipeline - Transactions Tab Window

Field	Description
Change Load Status	This transaction represents any modifications that may be made involving a load's status.
Close Delivery Plan	This transaction represents a delivery plan being closed and ready to be purged from the system.
Close Load	This transaction represents a load being closed and ready to be purged from the system.
Create Delivery Plan	This transaction represents the creation of a delivery plan.
Create Load	This transaction represents a the creation of a load.
Delete Delivery Plan	This transaction represents the deletion of a delivery plan.
Mark Load As Trailer Loaded	This transaction indicates that all the shipments belonging to a load have been loaded onto the appropriate Carrier.
Mark Load Not Loaded To Trailer	This transaction indicates that a load has not been loaded onto the appropriate Carrier.
Modify Delivery Plan	This transaction represents modifications made to a delivery plan.
Modify Load	This transaction represents modifications made to a load.
Purge Delivery Plan	This transaction represents the process of removing delivery plans from the system.
Purge Load	This transaction represents the process of removing loads from the system.
Receive Intransit Updates for Load	This transaction represents the process of receiving any updated concerning the status of a load while it is in transit.
Synchronize Task Queue	This transaction represents the process of synching the load execution task queue.

## Load Document: Statuses

### About this task

**Statuses** are the actual states that a document moves through in the pipeline. A transaction can contain two types of statuses, a drop status and a pickup status. A document is moved into a **drop status** when the events and conditions of a transaction have been completed. A **pickup status** takes the document from the previous drop status and moves it through the next transaction. Created and Scheduled are examples of statuses.

For more information about statuses, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

To view the status details of a load execution pipeline:

## Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Load Execution Process Model. The Load Execution window displays.
2. In the Load Execution window, choose the **Statuses** icon.
3. The Statuses tab window displays. Refer to Table 13 for field value descriptions.

## Results

For more information about creating and modifying statuses, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

Table 13. Load Execution Pipeline - Statuses Tab Window

Field	Description
Load Created	This indicates that a load has been created.
Trailer Loaded	This indicates that a load has been loaded on to the appropriate Carrier.
Load Intransit	This indicates that a load is physically in between stops configured in the delivery plan.
Load Completed	This indicates that a load has been delivered to the applicable stop or destination as per the delivery plan.

## Load Document: Conditions

### About this task

A **condition** matches document type attributes against decision points and routes the documents to different paths based on the specified attribute and value combinations. The document type attributes against which conditions can be created are predefined in Sterling Selling and Fulfillment Foundation. You can use these attributes in any combination or you can create conditions that run the appropriate application logic for specific circumstances.

For more information about conditions, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

To view the condition details of a load execution pipeline:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Load Execution Process Model. The Load Execution window displays.
2. In the Load Execution window, choose the **Conditions** icon.
3. The Conditions tab window displays.

### Results

For more information about creating and modifying conditions, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

## Load Document: Actions

### About this task

An **action** is a process or program that is triggered by an event. These processes and programs send user alert notifications and automatically resolve issues.

For example, when an order is released (the event), you can set an action to send the customer an e-mail.

For more information about actions, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

To view the action details of an outbound shipment pipeline:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Load Execution Process Model. The Load Execution window displays.
2. In the Load Execution window, choose the **Actions** icon.
3. The Actions tab window displays.

### Results

For more information about creating and modifying actions, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

---

## Chapter 10. Configuring a Document's Instruction Types

---

### Configuring a Document's Instruction Types

You can define the common codes used when adding special instructions to an order document.

The default instruction types of Sterling Selling and Fulfillment Foundation are:

- PICK
- PACK
- SHIP
- GIFT
- ORDERING
- OTHER

You can use the Instruction Types branch to create, modify, or delete an instruction type.

---

### Creating an Instruction Type

#### About this task

To create an instruction type:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Instruction Types. The Instruction Types window displays in the work area.
2. Choose the **Create New** icon. The Instruction Type Details pop-up window displays.
3. In Instruction Type, enter the instruction type.
4. In Short Description, enter a brief description of the instruction type.
5. In Long Description, enter a more detailed description of the instruction type.
6. Check Automatically Copy Item Instruction with Matching Type To Order Line to force the system to automatically copy item instructions with matching instruction types to order lines when the items are added onto an order.
7. Choose the **Save** icon.

---

### Modifying an Instruction Type

#### About this task

To modify an instruction type:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Instruction Types. The Instruction Types window displays in the work area.

2. Select the applicable instruction type and choose the **Details** icon. The Instruction Type Details pop-up window displays.
3. In Short Description, enter a brief description of the instruction type.
4. In Long Description, enter a more detailed description of the instruction type.
5. Check Automatically Copy Item Instruction with Matching Type To Order Line to force the system to automatically copy item instructions with matching instruction types to order lines when the items are added onto an order.
6. Choose the **Save** icon.

---

## Deleting an Instruction Type

### About this task

To delete an instruction type:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > Instruction Types. The Instruction Types window displays in the work area.
2. Select the applicable instruction type and choose the **Delete** icon.

---

## Chapter 11. Configuring a Load Document's Charge Categories

---

### Configuring a Load Document's Charge Categories

You can define **charge definitions** that you can associate with orders and invoices through charge categories. These categories contain a group of related charge names that can be used when the particular category is used.

Following are the Sterling Selling and Fulfillment Foundation default charge definitions:

- Shipping
- Handling
- Personalization
- Discount

Do not use charge definitions to create tax structures.

---

### Creating a Charge Category

#### About this task

To create a charge category:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > Load > Charge Categories. The Charge Categories window displays in the work area.
2. Choose the **Create New** icon. The Charge Category Details window displays.
3. In Charge Category, enter the name of the charge category.
4. In Description, enter a brief description of the charge category.
5. Select Billable if the charge is billable. Non-billable charges are not considered in order totals, but do appear in invoices.
6. Select Discount if the charge you are creating is a discount charge type.
7. Select Consider For Profit Margin if the category should be used for profit margin calculation.
8. Choose the **Save** icon.

#### Results

Charge categories cannot be localized. For more information about localization, see the *Sterling Selling and Fulfillment Foundation: Localization Guide*.

### Adding a Charge Name Associated with a Charge Category

#### About this task

Charge names are names of the actual charges included in the charge definition.

To add a charge name to a charge category:

### Procedure

1. In the Charge Category Details window, choose the **Create New** icon. The Charge Name Details pop-up window displays.
2. In Charge Name, enter the charge name.
3. In Description, enter a brief description of the charge name.
4. Choose the **Save** icon.

### Results

Charge names cannot be localized. For more information about localization, see the *Sterling Selling and Fulfillment Foundation: Localization Guide*.

## Modifying a Charge Name Associated with a Charge Category

### About this task

To modify a charge category's charge name:

### Procedure

1. In the Charge Category Details window, select the applicable charge name and choose the **Details** icon. The Charge Name Details pop-up window displays.
2. In Description, enter a brief description of the charge name.
3. Choose the **Save** icon.

## Deleting a Charge Name Associated with a Charge Category

### About this task

To delete a charge category's charge name select the applicable charge name in the Charge Category Details window and choose the **Delete** icon.

---

## Modifying a Charge Category

### About this task

To modify a charge category:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific > Load > Charge Categories. The Charge Categories window displays in the work area.
2. Select the applicable charge category and choose the **Details** icon. The Charge Category Details window displays.
3. In Description, enter a brief description of the charge category.
4. Select Billable if the charge is billable. Non-billable charges are not considered in order totals, but do appear in invoices.
5. Select Discount if the charge you are creating is a discount charge type.
6. Select Consider For Profit Margin if the category should be used for profit margin calculation.
7. Choose the **Save** icon.



---

## Deleting a Charge Category

### About this task

To delete a charge definition:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific > Load > Charge Categories. The Charge Categories window displays in the work area.
2. Select the applicable charge category and choose the **Delete** icon.



---

## Chapter 12. Configuring a Load Document's Purge Criteria

---

### Configuring a Load Document's Purge Criteria

You can set qualifications around each type of purge. A **purge** is the process by which old data is removed from the system database. Purges minimize the number of unused database records to increase search efficiency and reduce the size of the required physical disk. In Purge Criteria, default purge rules are provided. These can be modified for your system operations.

Table 14 lists the purge rules provided for load document types in Sterling Selling and Fulfillment Foundation.

Table 14. Load Document Type Purge Rules

Rule	Description	Retention Days
DELIVERYPLANPRG	Purges delivery plan information.	30
LOADPRG	Purges load information.	30

---

### Modifying a Load Document Type's Purge Criteria Rule

#### About this task

To modify a load document type's purge criteria rule:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > Load > Purge Criteria. The Purge Criteria window displays in the work area.
2. Select the applicable purge criteria rule and choose the **Details** icon. The Purge Criteria Details pop-up window displays.
3. Enter information in the applicable fields. Refer to Table 15 for field value descriptions.
4. Choose the **Save** icon.

Table 15. Purge Criteria Details Pop-up Window

Field	Description
Purge Code	Identifies a purge program. This is a system defined code.
Description	Describes the type of purge.
Rollback Segment	Defines the rollback segment that should be explicitly used for the purge transaction qualified by the purge code. This is useful when there are huge logical data sets that have to be purged. This is optional and used for order related purges.
Retention Days	Enter the number of days of data to be retained in the database (going backwards from the time the program runs). Make sure that your table size takes into account the number of retention days entered here.
Write to Log File	Select this field if you want purged data written to a log. The log can be backed up and used as a journal at a later date.

Table 15. Purge Criteria Details Pop-up Window (continued)

Field	Description
Log File Name	<p>Enter a log file name. The log file is created in the directory specified in the <code>yfs.purge.path</code> of <code>yfs.properties</code>. If a variable is introduced, then <code>yfs.purge.path</code> is ignored. For more information about using variables for the log file directory, refer to the <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i>.</p> <p>For information about file name limitations relating to internationalization, see the <i>Sterling Selling and Fulfillment Foundation: Localization Guide</i>.</p>

---

## Chapter 13. Configuring a Load Document's Load Types

---

### Configuring a Load Document's Load Types

You can define codes for load types that appear on a load document. A load carries one or more shipments from one point to another point.

This code has no application logic associated with it and can be set up as per your business practices.

---

### Creating a Load Type

#### About this task

To create a load type:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > Load > Load Type. The Load Type window displays in the work area.
  2. Choose the **Create New** icon. The Load Type Details pop-up window displays.
  3. In Load Type, enter the name of the load type.
  4. In Short Description, enter a brief description of the load type.
  5. In Long Description, enter a more detailed description of the load type.
  6. Choose the **Save** icon.
- 

### Modifying a Load Type

#### About this task

To modify a load type:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > Load > Load Type. The Load Type window displays in the work area.
  2. Select the applicable load type and choose the **Details** icon. The Load Type Details pop-up window displays.
  3. In Short Description, enter a brief description of the load type.
  4. In Long Description, enter a more detailed description of the load type.
  5. Choose the **Save** icon.
- 

### Deleting a Load Type

#### About this task

To delete a load type:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > Load > Load Type. The Load Type window displays in the work area.
2. Select the applicable load type and choose the **Delete** icon.



---

## Chapter 14. Configuring a Shipment Document's Hold Types

---

### Configuring a Shipment Document's Hold Types

You can configure the hold types that are applied to a shipment document. A shipment can be put on hold either manually or automatically by applying a specific hold type on it.

---

### Creating a Hold Type

#### About this task

To create a hold type:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific (Shipping) > Sales Order > Hold Types. The Hold Types window displays in the work area.
2. Click the **Create New** icon. The Hold Type pop-up window displays.
3. In Hold Type, enter the type of the hold.
4. In Hold Type Description, enter the description of the hold.
5. Enter information in the applicable fields. See Table 16, Table 17 on page 56, and Table 18 on page 57 for field value descriptions.
6. Click the **Save** icon.

Table 16. Hold Type Pop-Up Window, Hold Creation Tab

Field	Description
<b>Hold Created Automatically</b>	
On Shipment Creation	Check this box to apply this hold type to all shipments upon shipment creation.
On Resolution Of The Hold Type	Check this box to apply this hold type upon resolution of another hold type. From the drop-down list, select the hold type that, upon resolution, triggers this hold type. <b>Note:</b> Sterling Selling and Fulfillment Foundation does not check whether or not you are defining a circular hold type definition. For example, if you define hold type B as being applied upon resolution of hold type A, and hold type A as being applied upon resolution of hold type B, you could create an infinite loop that Sterling Selling and Fulfillment Foundation does not warn you against.
When The Following Modifications Are Performed	Check this box for modification types that automatically apply this hold type to a shipment.  Click the <b>Details</b> icon to modify the list. In the Modification Type List pop-up window: <ul style="list-style-type: none"><li>• Use the right arrow to move the available modification types that you wish to associate with this hold type to the subscribed list.</li><li>• Use the left arrow to unsubscribe the modification types that you wish to disassociate with this hold type and move them back into the available list.</li></ul>

Table 16. Hold Type Pop-Up Window, Hold Creation Tab (continued)

Field	Description
For All Shipments	Choose this option if the above conditions need to be applied to all shipments. <b>Note:</b> You can select this option only after the created hold has been saved.
Only For Shipments Satisfying The Following Condition	Choose this option if the above conditions need to be applied to shipments satisfying a certain condition.  Click the <b>Details</b> icon to build or modify the condition that is evaluated. For more information about using the condition builder, refer to the <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i> .  The available attributes for this condition can be extended. For more information, refer to the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i> . <b>Note:</b> You can select this option only after the created hold has been saved.
<b>Hold Created Manually</b>	
By Any User	Choose this option if any user group can apply this hold to a shipment.
By Users Who Belong To The Following Groups	Choose this option if only users belonging to certain user groups may apply this hold to a shipment.  Click the <b>Details</b> icon to modify the list. In the subsequent pop-up window: <ul style="list-style-type: none"> <li>• Use the right arrow to move the available user groups that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the user groups that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>

Table 17. Hold Type Pop-Up Window, Hold Resolution Tab

Field	Description
<b>Hold Resolved Automatically</b>	
The Following Time Triggered Transaction Will Process Created Holds	From the drop-down list, select the time-triggered transaction that processes the created holds.
The Following Time Triggered Transaction Will Process Rejected Holds	From the drop-down list, select the time-triggered transaction that processes the rejected holds.
<b>Hold Resolved Manually</b>	
Any User Can Process This Hold	Choose this option if any user group can process this hold.



Table 17. Hold Type Pop-Up Window, Hold Resolution Tab (continued)

Field	Description
Users Belonging To The Following User Groups Can Process This Hold	<p>Choose this option if only users belonging to certain user groups can process this hold.</p> <p>Click the <b>Details</b> icon to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available user groups that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the user groups that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>

Table 18. Hold Type Pop-Up Window, Hold Effects Tab

Field	Description
The Following Transactions Will Be Stopped From Processing Shipments On This Hold	<p>Transactions that are disallowed when this hold type is applied to a shipment.</p> <p>Click the <b>Details</b> icon to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available modification types that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe the modification types that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>
The Following Modifications Are Not Allowed For Shipments On This Hold	<p>Modification types that are disallowed when this hold type is applied to a shipment.</p> <p>Click the <b>Details</b> icon to modify the list. In the subsequent pop-up window:</p> <ul style="list-style-type: none"> <li>• Use the right arrow to move the available transactions that you wish to associate with this hold type to the subscribed list.</li> <li>• Use the left arrow to unsubscribe transactions that you wish to disassociate with this hold type and move them back into the available list.</li> </ul>

## Modifying a Hold Type

### About this task

To modify a hold type:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific (Shipping) > Sales Order > Hold Types. The Hold Types window displays in the work area.
2. Select the applicable hold type and click the **Details** icon. The Hold Type pop-up window displays.
3. Enter information in the applicable fields. See Table 16 on page 55, Table 17 on page 56 and Table 18 for field value descriptions.

4. Click the **Save** icon.
- 

## Deleting a Hold Type

### About this task

To delete a hold type:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific (Shipping) > Sales Order > Hold Types. The Hold Types window displays in the work area.
2. Select the applicable hold type and click the **Delete** icon.

---

## Chapter 15. Configuring a Load Document's Stop Types

---

### Configuring a Load Document's Stop Types

You can modify codes for stop types that appear on a load document. A stop is a location where a shipment is picked up or dropped off.

This code has no application logic associated with it and can be set up as per your business practices.

Following are the Sterling Selling and Fulfillment Foundation default stop types:

- D - Destination
- I - Intermediate
- O - Origin

---

### Modifying a Stop Type

#### About this task

To modify a stop type:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > Load > Stop Type. The Stop Type window displays in the work area.
2. Select the applicable stop type and choose the **Details** icon. The Stop Type Details pop-up window displays.
3. In Short Description, enter a brief description of the stop type.
4. In Long Description, enter a more detailed description of the stop type.
5. Choose the **Save** icon.



---

## Chapter 16. Configuring a Load Document's Modification Components

---

### Configuring a Document's Modification Components

You can configure the modification rules and types of a document when it is in a specific status. These rules determine which parts of a document can be modified as well as in which status the modifications can be performed. The set of rules and types can be classified into modification groups. Each modification group can be associated with a user-defined condition.

If you are using the Distributed Order Management module, you can configure modification components at the following process type levels:

- Fulfillment
- Outbound Logistics

If you are using the Logistics Management module, you can configure modification components at the load process type level.

If you are using the Supply Collaboration module, you can configure modification components at the following process type levels:

- Fulfillment
- Inbound Logistics
- Receipt
- Negotiation

If you are using the Reverse Logistics module, you can configure modification components at the following process type levels:

- Fulfillment
- Logistics
- Receipt

You can use the Order Modification branch to define custom modification types, modification groups, modifications impacting pricing, and modifications requiring auditing.

---

### Defining Custom Modification Types

You can define custom modification types for a process type. Creating a modification type allows you to classify certain attributes (including extended attributes) into one group. Later, when you define status modification groups, you can configure modification rules for the custom modification type. Modification rules determine modification permissions for the attributes. You can decide whether to allow, disallow, or ignore the custom modification type for a given status.

You can use the Order Modification Types branch for creating, modifying, and deleting a custom modification type. After you define a custom modification type, use the Status Modification Group Details window to configure modification rules for the custom modification type.

## Creating a Custom Modification Type

### About this task

To create a custom modification type:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > (*Process Type*) > (*Process Type*) Modification > (*Process Type*) Modification Types. The Custom Modification List window displays in the work area.
2. From the Custom Modification List, click the **Create New** icon. The Custom Modification window displays. Enter information in the applicable fields. Refer to Table 19 for field value descriptions.
3. Click the **Save** icon. A pop-up warning you to sign out of the application for changes to take place displays.

Table 19. Custom Modification Window

Field	Description
<b>Modification Level</b>	Select the level of the modification type. For example, Header, Line, or Release.
<b>Modification Type</b>	Enter the name of the modification type.
<b>Description</b>	Enter a brief description of the modification type.
<b>Min Allowed Status</b>	Select the minimum status the modification type can be performed at.
<b>Max Allowed Status</b>	Select the maximum status the modification type can be performed at.
<b>Available</b>	A list of XML attributes that can be associated with the modification type. To add an available attribute to the modification type, select the attribute you want to add and click the <b>Right Arrow</b> icon.
<b>Subscribed</b>	A list of XML attributes that have been associated with the modification type. To remove a subscribed attribute, select the attribute you want to remove and click the <b>Left Arrow</b> icon.

## Modifying a Custom Modification Type

### About this task

To modify a custom modification type:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > (*Process Type*) > (*Process Type*) Modification > (*Process Type*) Modification Types. The Custom Modification List window displays in the work area.
2. From the Custom Modification List, locate the applicable Custom Modification and click the **Details** icon. The Custom Modification window displays.
3. Enter information in the applicable fields. Refer to Table 19 for field value descriptions.
4. Click the **Save** icon.

## Deleting a Custom Modification Type

### About this task

To delete a custom modification type:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > (*Process Type*) > (*Process Type*) Modification > (*Process Type*) Modification Types. The Custom Modification List window displays in the work area.
2. From the Custom Modification List, locate the applicable Custom Modification and click the **Delete** icon.

---

## Defining Status Modification Groups

Status modification groups enable you to configure status modification rules that are based on enterprise, process type, and order conditions. You define a status modification group by configuring the following functionalities:

- Modification Rules - Define modification rules for subscribed modification types.
- Condition - Specify a condition for validating the status modification group.
- Is Override - Specify the status modification group has priority over other status modification groups.
- Inherited From Status Modification Group - Specify the status modification group inherits status modification rules from another status modification group.

## Configuring a Load Document's Modification Rules

Most load document types flow through a pipeline without requiring any intervention. However, there are times when modifications are required. Sterling Selling and Fulfillment Foundation supports the modifications through the Application Console. It is critical for you to decide which modifications are allowed for each modification type, modification level, and status combination.

**Note:** Contemplate the business and system integration implications before turning on a setting that is disallowed as part of the system defaults.

Modification type indicates the type of modification carried out on the load document type. Sterling Selling and Fulfillment Foundation provides ability to perform modifications on specific attributes. An example of a modification type is adding a stop to a load's delivery plan.

Modification level indicates the level at which a particular modification type is carried out. The load document type has a modification level of Load. Modifications are applied to a particular level and a particular processing status.

## Load Document Modification Types

The following are the default order modification types and their associated modification level:

Table 20. Load Document Modification Types

Modification Type	Description	Modification Level
Add Instruction	An instruction can be added to a load document.	Load
Add Shipment	A shipment can be added to a load document.	Load
Add Stop	A stop can be added to a load document.	Load
Add/Remove Additional Date	Additional dates that can be associated with a load document can be added and removed.	Load
Add/Remove Charge	A charge can either be added to or removed from an load document.	Load
Change Instruction	A modification can be made to an instruction associated with a load document.  The following instruction fields can be modified when this modification type is allowed: <ul style="list-style-type: none"> <li>• Instruction Type</li> <li>• Text</li> <li>• URL</li> </ul>	Load
Change Other Attributes	A modification can be made to fields that do not have system or user-defined modification types associated with them.	Load
Change Other Relationships	Not used in this version.	Load
Delete Load	A load document can be deleted.	Load
Delete Stop	A stop can be deleted from a load document.	Load
Modify Stop	A modification can be made to a stop associated with a load document.	Load
Remove Shipment	A shipment can be removed from a load document.	Load
Resequence Stop	A load document's stops can be reordered.	Load

You can group modifications in the window by modification type, modification level, or status, by selecting the corresponding grouping from Group By. The Modification Rules window then displays the grouping you have chosen in a hierarchical structure.

Table 21. Load Document Type Rule Modifications

Field	Description
Status	Indicates each status that is applicable to a modification level and type.



Table 21. Load Document Type Rule Modifications (continued)

Field	Description
Allow	Indicates whether or not modifications may be made at this modification level and type for the specified status.
Disallow	Indicates that no modifications may be made at this modification level and type for the specified status.
Ignore	Indicates that modifications are ignored at this modification level and type for the specified status.

Any permissions set up for user groups override configured modification rules. For more information about configuring user group permissions, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

## Specifying a Condition for a Status Modification Group

You can define a condition for a status modification group that determines how the group is validated against status modifications. For example, when a user of an enterprise performs an operation on a supported document type or calls an order-related API, such as `getOrderDetails`, the status modification is validated using this process:

- Obtain the conditional XML file from the order. The conditional XML file is created when you specify a condition for the status modification group.
- Obtain all applicable status modification groups that meet the conditions specified in the conditional XML file.
- Obtain all modification types from applicable status modification groups, including inherited configurations. Then, any conflicting status modification rules are resolved.

Conditions for status modification groups are specified by order attribute. For example, the `CurrentUserRole` attribute lets you configure status modification rules for a primary user group, such as Field Sales. Consider a scenario in which you want to give field sales representatives the ability to create quotes. However, you do not want to allow them to change the price on existing quotes. To configure this scenario, you would create a status modification group where the `CurrentUserRole` attribute is set to the primary user group id for field sales representatives. As part of creating the status modification group, you would define status modification rules that allow changes to quotes but prevent changes to prices on existing quotes. Additionally, you could create a different status modification group that allows supervisors to change the price on existing quotes. In this example, assume that the enterprise and process type are the same for both status modification groups.

If you do not specify a condition for a status modification group, the status modification group validation is enforced. Use the Condition field in the Status Modification Group Details window to specify a condition for a status modification group.

## Overriding Status Modification Groups

You can use the **Is Override** option to specify that a status modification group has priority over other status modification groups. For example, if three status modification groups contain the same status modification rule, where one group has the **Is Override** option selected, the modification rule is set to the value specified in the override group.

When multiple status modification groups have the **Is Override** option selected, conflicts are resolved using the logic described in Table 22.

Table 22. Multiple Status Modification Groups and the Is Override Option

	Status	Status	Status	Status	Status	Status
<b>Is Override</b> - checked for Group 1	Allow	Allow	Ignore	Disallow	Allow	Disallow
<b>Is Override</b> - checked for Group 2	Allow	Disallow	Ignore	Disallow	Ignore	Ignore
<b>Result</b>	Allow	Disallow	Disallow	Disallow	Allow	Disallow

## Inheriting Modification Rules from a Status Modification Group

When you create a status modification group, it inherits modification rules from the default status modification group. You can then define the group by configuring its modification rules. Additionally, modification groups can inherit status modification rules from other status modification groups. For example, you can create a status modification group that allows field sales representatives to create orders. If you create a group for managers that inherits status modifications rules from the Field Sales Representative Group, the managers can also create orders. Later, if you want to prevent field sales representatives and managers from creating orders, you only need to disallow the create order status for the Field Sales Representative Group. The Manager Group inherits the updated status modification.

When one group inherits rules from another group, the modification rules defined in the child modification group override the parent's modification rules. For example, the Field Sales Representative Group can create orders but cannot change the price on a quote. If you want the Manager Group to have the ability to create orders and change the price on quotes, configure the Manager Group to inherit status modification rules from the Field Sales Representative group, and then configure the Manager Group to have the ability to change the price on a quote.

Additionally, the **Is Override** option can be enabled for child groups. For example, a manager belongs to the following status modification groups:

- Manager Group 1 - This group is not allowed to create orders.
- Manager Group 2 - This group inherits the ability to create orders from the Field Sales Group. Manager Group 2 has the **Is Override** option enabled.

Because the **Is Override** option is enabled for Manager Group 2, the manager has the ability to create orders.

## Resolving Conflicts Between Status Modification Groups

In some cases, multiple status modification groups contain conflicting status modification rules. Conflicts between status modification groups are resolved in the following way:

- If two status modification groups contain the same rule, where one group has the rule set to Allow and the other group has the rule set to Disallow, the rule is set to Allow.

- If no status modification group has the rule defined, the rule is set to the value specified in the parent group. If there is no parent group, the rule is set to the value in Default Group.
- If multiple status modification groups have the **Is Override** option selected, but the groups do not have the rule defined, the rule is set to the value specified in the parent group. If there is no parent group, the rule is set to the value in Default Group.

The following example provides a scenario for resolving conflicts between status modification groups.

### Example: Resolving Conflicts Between Status Modification Groups

In this example, three status modification groups contain status modification rules that affect the same order.

*Table 23. Default Status Modification Group*

1500 OrderLine	Cancel	Disallow
2300 Order ShipTo	Change	Disallow
2400 Order ShipNode	Change	Disallow

Use Status Modification Group 1 when condition 1 is true:

*Table 24. Status Modification Group 1*

<b>Inherits from Default Group</b>		
1500 OrderLine	Cancel	Disallow
2400 Order ShipNode	Change	Disallow

Use Status Modification Group 2 when condition 2 is true:

*Table 25. Status Modification Group 2*

<b>Inherits from Default Group</b>		
1500 OrderLine	Cancel	Disallow

Use Status Modification Group 3 when condition 3 is true:

*Table 26. Status Modification Group 3*

<b>Inherits from Default Group</b>		
Is Override is Yes		
2400 Order ShipNode	Change	Disallow

If the order satisfies all conditions, status modification groups 1, 2, and 3 are applicable. As a result, the following access is granted:

*Table 27. Resulting Access for Conflicting Status Modification Groups*

1500 OrderLine	Cancel	Allowed	Condition 1 applied
2300 Order ShipTo	Change	Disallow	Default applied
2400 Order ShipNode	Change	Disallow	Condition 3 applied

## Creating a Status Modification Group

### About this task

To create a status modification group:

#### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (Document Type) > (Process Type) > (Process Type) Modification > (Process Type) Modification Groups. The Status Modification Group List window displays in the work area.
2. From the Status Modification Group List, click the **Create New** icon. The Status Modification Group Details window displays.
3. Enter information in the applicable fields. For field value descriptions, see Table 28.
4. Click the **Save** icon to save the modification group.

Table 28. Status Modification Group Details Window

Field	Description
<b>Status Modification Group ID</b>	Enter a name for the status modification group.
<b>Description</b>	Enter a description for the status modification group.
<b>Inherited From Status Modification Group</b>	(Optional) Select a parent status modification group if you want to inherit modification rules from another group.
<b>Is Override</b>	(Optional) Check this option if you want this status modification group to take precedence.
<b>Condition</b>	(Optional) Select a condition if you want to specify a condition for this status modification group. Or, you can create a condition by clicking the <b>Create New</b> icon, which opens the Condition Detail window. Refer to "Adding Conditions to a Status Modification Group" on page 69 for information about configuring conditions for status modification groups. <b>Note:</b> If you do not specify a condition, this group's validation is enforced.
<b>Subscribed Modification Types</b>	
<b>Available</b>	Displays a list of available modification types. This list includes all system-supplied modification types as well as the custom modification types that you created in the Custom Modification window. To subscribe a modification type, select the modification type in the Available list and click the <b>Right Arrow</b> icon.

Table 28. Status Modification Group Details Window (continued)

Field	Description
<b>Subscribed</b>	Displays a list of the modification types for which you want to define modification rules. To remove a modification type from the subscribed list, select the modification type in the Subscribed list and click the <b>Left Arrow</b> icon. Click the <b>Save</b> icon to display the subscribed modification types in the Modification Rules tab.
<b>Modification Rules</b>	
<b>Group By</b>	Select a sort option for displaying modification types in the <b>Primary Info</b> tab. You can sort modification types by type, level, and status.
<b>Primary Info</b>	Displays the modification types that you subscribed in the Subscribed Modification Types tab. Expand the applicable modification types and levels for which you want to set up modification rules. Right click on the applicable rule and choose Allow, Disallow, or Ignore as per your business practices.

## Adding Conditions to a Status Modification Group

### About this task

The Condition Detail window allows you to create a new condition for the status modification group. The Condition Detail window opens when you click the **Create New** icon, next to the Condition field, in the Status Modification Group Details window.

To create a condition for the status modification group:

### Procedure

In the Condition Detail window, enter information in the following fields:

Option	Description
<b>Condition ID</b>	Enter the condition ID.
<b>Condition Name</b>	Enter the name of the condition for the status modification group.
<b>Condition Group</b>	Enter the name of the condition's group, if applicable. Condition Group allows you to group related conditions within the condition tree.
<b>Static</b>	If this is checked, you must enter a condition value for the static condition.
<b>Dynamic</b>	If this is checked, you must enter a Java class name that evaluates the condition at runtime.

Option	Description
Advanced XML	<p>If you are creating a new condition, this option is disabled as a new condition of the advanced XML type must be created using the IBM Greex Editor IDE tool. For more information about creating an advanced XML condition using the IBM Greex Editor, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This option is automatically selected whenever you modify a condition of the advanced XML type.</p>
Condition Value (if Static is checked)	<p>Click the <b>Condition Builder</b> icon to use the condition builder, which is where you set up the conditional value for the status modification group. You can set it up in a formulaic readout using the available symbols. You can enter your own attribute or an extended attribute if Static condition is checked. For example, if you are configuring a status modification group for the Field Sales Representative Group, use the {Enter Your Own Attribute} option in the Condition Builder to set the CurrentUserGroup attribute equal to the primary user group id for field sales. For more information about creating these attributes, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p>
Class Name (if Dynamic is checked)	<p>Enter the class name that implements the following Java interface: com.yantra.ycp.japi.YCPDynamicCondition</p> <p><b>Note:</b> To use extended attributes for a condition, implement the YCPDynamicConditionEx interface. For more information about implementing this interface, refer to the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p>
Condition Properties (if Dynamic is checked)	<p>Specify the custom name or value properties which are set into the condition evaluating Java class file before evaluating the condition. For more information about creating custom attributes, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p>

## Modifying a Status Modification Group

### About this task

To modify a status modification group:

## Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > (*Process Type*) > (*Process Type*) Modification > (*Process Type*) Modification Groups. The Status Modification Group List window displays in the work area.
2. From the Status Modification Group List, select the applicable modification group and choose the **Details** icon. The Status Modification Group Details window displays.
3. Enter information in the applicable fields. Refer to Table 28 on page 68 for field value descriptions.
4. Click the **Save** icon.

## Deleting a Status Modification Group

### About this task

To delete a status modification group:

### Procedure

1. From the tree in the application rules side panel, choose Document Specific > (*Document Type*) > (*Process Type*) > (*Process Type*) Modification > (*Process Type*) Modification Groups. The Status Modification Group List window displays in the work area.
2. From the Status Modification Group List, locate the applicable status modification group and click the **Delete** icon.

---

## Changing a Load Document Type's Modification Requiring Auditing Rules

### About this task

To change a load document type's modification requiring auditing rules:

**Note:** The Modification Requiring Auditing configurations you make will be effective only when the auditing is enabled for the YFS\_LOAD entity.

### Procedure

1. From the tree in the application rules side panel, choose Document Specific > Load > Load Modification Requiring Auditing. The Modifications Requiring Auditing List window displays in the work area.
2. Choose the **Modify** icon. The Modification Type List window displays. Refer to Table 29 for field value descriptions.

Table 29. Modification Type List Window

Field	Description
Modification Types	A list of modification types for which auditing can be enabled. To add a modification type to the modification requiring auditing list, select the modification type you want to add and choose the <b>Right Arrow</b> icon.
Modifications Requiring Auditing	A list of modification types that require auditing. To remove a modification type from the modification requiring auditing list, select the modification type you want to remove and choose the <b>Left Arrow</b> icon.

3. Choose the **Save** icon.



---

## Chapter 17. Configuring an Order Document's Shipment Specific Components

---

### Configuring an Order Document's Shipment Specific Components

To complete an order document's life cycle, each document has a set of different processes that it goes through. These processes are called process types. Each order document has a defined set of process types in Sterling Selling and Fulfillment Foundation.

You can configure rules and components specific to an order document's shipment process type.

---

### Defining Shipping Modification Groups

The shipping modification groups apply to the following document types:

- Sales Order
- Transfer Order
- Return Order
- Purchase Order

For more information about modification group, see the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

For more information about defining and changing modification groups, see *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

---

### Defining Shipping Hold Types

The shipping hold types apply to the following document types:

- Sales Order
- Transfer Order
- Return Order
- Purchase Order

For more information about hold types, see the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

For more information about configuring and modifying hold types, see the "Configuring a Shipment Document's Hold Types" chapter.

---

### Defining Shipping Process Type Details

The shipping process type details apply to the following document types:

- Sales Order
- Transfer Order
- Return Order
- Purchase Order

For more information about the shipping process type details, see the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

For more information about defining and modifying the process type details, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

---

## Defining Shipping Process Model

The shipping process model applies to the following document types:

- Sales Order
- Transfer Order
- Return Order
- Purchase Order

For more information about the shipping process model, see the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

For more information about pipeline determination, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

---

## Chapter 18. Configuring Logistics Components

---

### Defining Logistics

You can define rules and common codes associated logistics of shipping an order.

#### Defining Freight Terms

You can define common codes used when associating a freight term to a Carrier. A **freight term** identifies how transportation costs are calculated.

The default freight terms of Sterling Selling and Fulfillment Foundation are:

- Cost Insurance and Freight (CIF) - The freight cost is completely paid by either the Seller, the Enterprise, or the Hub.
- Cost and Freight (CFR) - The freight cost is paid by the Buyer and either the Seller, the Enterprise, or the Hub.
- Free On Board (FOB) - The freight cost is paid by the Buyer.

You can use the Freight Terms tab for creating, modifying, and deleting freight terms.

#### Creating a Freight Term

##### About this task

To create a freight term:

##### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Logistics Attributes. The Logistics window displays in the work area.
2. Choose the Freight Terms tab.
3. Choose the **Create New** icon. The Freight Terms Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 30 for field value descriptions.
5. Enter Choose the **Save** icon.

Table 30. Freight Terms Details

Field	Description
<b>Freight Terms</b>	Enter the name of the freight term.
<b>Short Description</b>	Enter a brief description of the freight term.
<b>Long Description</b>	Enter a more detailed description of the freight term.

Table 30. Freight Terms Details (continued)

Field	Description
Consider Buyer's Routing Guide	Both the Buyer and the Enterprise can establish routing guides (rules for shipping), and Economic Shipping parameters (ESP), which control how items are shipped. In some cases only the Buyer organization has established values for these rules. In other cases, only the enterprise has established values for these rules. If neither is set, then Hub rules are used.  In cases where both the Buyer and the Enterprise have set values for these rules, this setting determines whether to apply the Buyer's routing rules before applying the routing rules of the Enterprise. See the <i>Sterling Selling and Fulfillment Foundation: Product Concepts Guide</i> for more information about these shipping concepts.
First Buyer then Enterprise	Select to use any shipping rules established by the buyer first. Enterprise rules are applied if no applicable Buyer rule exists.
First Enterprise then Buyer	Select to use any shipping rules established by the enterprise first. Buyer rules are applied if no applicable Enterprise rule exists.
Charges paid by	
Buyer	Select this option if the Buyer pays shipping charges.
Shipper	Select this option if the Shipper pays shipping charges.

## Modifying a Freight Term

### About this task

To modify a freight term:

#### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Logistics Attributes. The Logistics window displays in the work area.
2. Choose the Freight Terms tab.
3. Select the applicable freight term and choose the **Details** icon. The Freight Terms Details pop-up window displays.
4. Enter the new information in the applicable fields. Refer to Table 30 on page 75 for field value descriptions.
5. Choose the **Save** icon.

## Deleting a Freight Term

### About this task

To delete a freight term:

#### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Logistics Attributes. The Logistics window displays in the work area.
2. Choose the Freight Terms tab.
3. Select the applicable freight term and choose the **Delete** icon.

# Defining Shipment Planning

## About this task

Shipment Planning is used to describe conditions that control how shipping is done. These include whether certain items can be shipped together, such as regular and rush orders, whether to use Economic Shipping Parameters, and how routing is performed.

To define shipping planning:

### Procedure

1. From the tree in the application rules side panel, choose Logistics > Shipment Planning. The Shipment Planning window displays in the work area.
2. Enter information in the applicable fields. Refer to Table 31 for field value descriptions.
3. Choose the **Save** icon.

Table 31. Shipment Planning Window

Field	Description
Do not mix in a shipment	If any of the following are selected, separate shipments must be created for items that have different values for these attributes.  For example, if Department Code is selected, items that are for different departments can not be included in the same shipment.
Buyer Mark For Node Id	The buyer mark for node id.
Customer PO #	Customer's Purchase Order number.
Department Code	The department for which the item is intended.
Gift Flag	The gift flag.
Level of Service	The level of service on the order.
Mark For	Person for whom this shipment is marked for.
Order #	The order number.
Order Type	The order type.
Transportation optimization	
Economic shipping parameters maintained	Economic Shipping Parameters (ESP) are used in shipping consolidation. Select this field to enable the following Economic Shipping Parameters fields.  ESP support consolidation of shipments until a weight or volume threshold is met, or until a certain time elapses. By consolidating shipments, shipping costs can be reduced  For example, you can set that shipments should be consolidated until the shipment weight is 300 pounds, or 50 cubic feet in volume. To ensure that eventually the shipment is set, you can establish a maximum number of days to wait until the conditions are met.  When either the weight, volume or delay shipment threshold is met, the shipment is moved to the next stage in shipping.

Table 31. Shipment Planning Window (continued)

Field	Description
Delay shipment by not more than __ Days	Enter the number of days this shipment can be delayed before it should be shipped.  For example, if a value is set for weight threshold of 300 pounds, and this field has been set to 3 days, the shipment is shipped after 3 days, regardless of whether the weight threshold has been met.
Consolidate up to weight threshold of	Enter a weight.
Consolidate up to volume threshold of	Enter a volume
Region Schema For Routing	Select the applicable region schema for routing.  Select the <b>Details</b> icon. The Region Schema Details window displays. For more information about modifying the region schema details, refer to the <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i> .
Routing Guide	
Not Maintained	Select this to use manual routing. Shipments are managed in the shipment console, and any routing guides are not consulted.
Maintained in Sterling Selling and Fulfillment Foundation	Select this to use the Routing Guides maintained in Sterling Selling and Fulfillment Foundation to determine how shipments should be routed. See "Creating a Routing Guide."  In addition to the routing guide maintained here by the enterprise, there may be a routing guide for the buyer organization. For more information about configuring routing guides for buyer organizations, see the <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i> .  For more information about using both buyer and enterprise routing guides, see "Creating a Routing Guide."
Maintained Externally	Select this to indicate that an external routing system is used. The routing guides maintained in Sterling Selling and Fulfillment Foundation are not consulted.  Examples of external routing systems include using an integrated Transportation Management System (TMS), or implementing a User Exit which consults with the buyer organization.

## Creating a Routing Guide

### About this task

*Routing Guides* are a list of conditions which determine how a shipment should be routed. A routing guide has a time period for which is effective, and conditions for when it should be applied. These conditions are based on Freight Terms and Department.

Each routing guide contains a list of *routing guide lines*, each of which describe detailed conditions for selecting a carrier. The routing guide information is based on data used by VICS (Voluntary InterIndustry Commerce Standards) routing.

To create a routing guide:

### Procedure

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Outbound Constraints. The Outbound Constraints window displays in the work area.
2. Select the **Create New** icon on the Routing Guides list window. The Routing Guide Details window displays in the work area.
3. Enter information in the applicable fields. Refer to Table 32 for field value descriptions.
4. Choose the **Save** icon.

Table 32. Routing Guide Details Window

Field	Description
<b>Name</b>	Enter a name for the routing guide.
<b>Routing Guide #</b>	A number for the routing guide.
<b>Publish Date</b>	When this routing guide is available within the system.
<b>Supersedes Routing Guide #</b>	Tracking information. For example, if a minor revision is made to routing guide "1234", you might create a routing guide "1234-A", and enter that it supersedes routing guide "1234". This field is for informational purposes and is not used to determine the effective routing guide.
<b>Effective Date</b>	The start date for applying the routing information in this routing guide. You can use the effective date and expiration date to apply routing guidelines for particular periods of time.
<b>Expiration Date</b>	The end date for applying the routing information in this routing guide.
<b>Apply this Routing Guide when</b>	
<b>Freight Terms</b>	Apply this routing guide when this condition is met. Select <i>is</i> , <i>is in</i> , or <i>is not</i> . Use: <ul style="list-style-type: none"> <li>• <i>is</i> to specify a single Freight Term.</li> <li>• <i>is in</i> to specify a group of Freight Terms, one of which must be matched.</li> <li>• <i>is not in</i> to specify a group of Freight Terms. The routing guide is used if the Freight Term does not match one of these values.</li> </ul>
<b>Item Classification</b>	Items can be classified.  This field displays when valid item classifications have been set up for Routing Guide.

## Modifying a Routing Guide Line

### About this task

To modify a routing guide line:

### Procedure

1. From the Routing Guidelines Details window, select the Routing Details Tab. A Routing Guide Line search window displays.

2. Select a routing guide line in the Routing Guide Line list window, and select the **Modify** icon. The Routing Guide Line Details window displays.
3. Enter the new information in the applicable fields. Refer to the "Routing Guide Line Details" table for field value descriptions.
4. Choose the **Save** icon.

## Creating a Routing Guideline

### About this task

*Routing guide lines* contain the specific conditions to use when routing a shipment. A routing guide can contain multiple routing guide lines.

When routing occurs, the shipment is matched against the routing guide lines. Based on the criteria specified, a carrier and carrier service is selected. The shipment mode is determined for shipments/loads based on the combination of the carrier and carrier service.

When routing results in a change to the shipment destination, the system reroutes the shipment, with the revised destination as the factor for routing. This type of configuration is used for consolidator nodes. While routing the second time, system looks for the routing guide entry that contains destination node, but without any other destination parameters filled out (such as address, etc.). However, the consolidator destination node should be defined.

To create a routing guideline:

### Procedure

1. From the Routing Guidelines Details window, select the Routing Details Tab. To have access to the Routing Details Tab, save the information you have entered on the Primary Info Tab.
2. A Routing Guide Line search window displays.
3. Select the **Create New** icon. A Routing Guide Line Details screen displays in the work area.
4. Enter information in the applicable fields. Refer to Table 33 for field value descriptions.
5. Choose the **Save** icon.

Table 33. Routing Guide Line Details

Setting conditions	
In many of the following fields, you can select <i>is</i> , <i>is in</i> , or <i>is not in</i> , and then specify a value. Use:	
<ul style="list-style-type: none"> <li>• <i>is</i> to specify that a single value must be matched</li> <li>• <i>is in</i> to specify a group of values, one of which must be matched.</li> <li>• <i>is not in</i> to specify a group of values. The routing guide line is used if none of these values match.</li> </ul>	
For example, to match any one of a group of states, specify	
State <i>is in</i> California, Washington, Oregon, Nevada.	
When assessing the condition, California would match, Florida would not.	
Field	Description
When shipping from:	



Table 33. Routing Guide Line Details (continued)

Node	Select the node.
When ship from is not node, select the following attribute(s)	Select one or more of the following conditions.
Country/Region	Select the country or region name(s).
State	Enter the state name(s).
City	Enter the city name(s).
Zip Code	Enter the zip code or zip code range.
And shipping to:	
Node	Select the node.
Region	Select the region.
When ship to is not node and region, select the following attribute(s)	Select one or more of the following conditions.
Country/Region	Select the country or region name(s).
State	Enter the state name(s).
City	Enter the city name(s).
Zip Code	Enter the zip code or zip code range.
Consolidator	Select the consolidator name(s)
Store#	Select the store number(s).
And weight is in the range:	You can match weight. For example, if you want packages that weigh between 100 and 500 pounds to be shipped using a specific carrier, you would specify From as '100' and To as '500'.
From	Enter the minimum value.
To	Enter the maximum value.
And volume is in the range:	You can match volume. For example, if you want packages that are between 3 and 10 cubic feet to be shipped using a specific carrier, you would specify From as '3' and To as '10'.
From	Enter the minimum value.
To	Enter the maximum value.
And handling units are in the range:	Number of cases.
From	Enter the minimum value.
To	Enter the maximum value.
And if requested carrier service code is	
Carrier Service Code	Select a carrier service code.
For more information about defining carrier services, see "Defining Carrier Services" on page 82.	
Then ship via:	

Table 33. Routing Guide Line Details (continued)

Priority	Indicates the number to give this rule a relative importance.  When a shipment is compared to the routing guide lines, there may be two carrier services that could be used. The priority serves as a tie breaker. The carrier service with the lowest number is used.						
Carrier / Service	Indicates the carrier and service code that is desired.						
Break Bulk Node	The break bulk location that is close to the buyer.						
Contact Specified	Indicates whether the contact details for the carrier is specified.						
With overrides:							
Override Freight Terms	Select to override the shipment's Freight Term.						
Override Ship To	To override the Ship To value, check this box and select one of the following:  <table border="1" data-bbox="786 751 1393 890"> <tr> <td>Node</td> <td>Select the node name.</td> </tr> <tr> <td>Consolidator</td> <td>Select the consolidator name.</td> </tr> <tr> <td>Store#</td> <td>Select the store number.</td> </tr> </table> This is only used when performing routing again due to a revised ship to address.	Node	Select the node name.	Consolidator	Select the consolidator name.	Store#	Select the store number.
Node	Select the node name.						
Consolidator	Select the consolidator name.						
Store#	Select the store number.						

## Results

When the conditions set are assessed, the routing guide line that matches most conditions is used. For example, imagine there are three routing guide lines:

*Routing guide line A* - What to do when shipping from Massachusetts

*Routing guide line B* - What to do when shipping from Massachusetts, and when shipping from the zip code 01810.

*Routing guide line C* - What to do when shipping from Massachusetts or NY.

If the shipment originates from the zip code 01810, it matches all of these routing guide lines. The actions specified in *Routing guide line B* is used, as more conditions are met (both the state and the zip code).

If the shipment originates from Massachusetts, but not from zip code 01810, then both *Routing guide line A* and *Routing guide line C* match.

**Defining Carrier Services:** When routing occurs, the shipment is matched against the routing guidelines. Based on the criteria specified, you select a carrier service to use.

You can use the Carrier Services panel for creating, modifying, or deleting a carrier service.

### Creating a Carrier Service as a Routing Option:

### About this task

To create a carrier service:

#### Procedure

1. From the Routing Guidelines Details window, in the Carrier Services panel, select the **Create New** icon. The Carrier Services window displays.
2. Enter information in the applicable fields. Refer to Table 34 for field value descriptions.
3. Choose the **Save** icon.

Table 34. Carrier Services

Fields	Description
Priority	Enter a number to give this rule a relative importance.  When a shipment is compared to the routing guide lines, there may be two carrier services that could be used. The priority serves as a tie breaker. The carrier service with the lowest number is used.
Carrier/Service	Select the carrier or service code that is desired.
Break Bulk Node	Select the break bulk location.
Consolidation Requirement for Break Bulk Node	Used to specify the minimum weight and volume criteria for consolidation of shipments for the Break Bulk Node.
Minimum Weight	Enter the minimum weight required to consolidate shipments.
Minimum Volume	Enter the minimum volume required to consolidate shipments.
Contact Address	Used to specify the contact address information. Click the <b>Details</b> icon to change the contact Address.

### Modifying a Carrier Service as a Routing Option:

#### About this task

To modify a carrier service:

#### Procedure

1. From the Routing Guidelines Details window, in the Carrier Services panel, select a carrier service in the Carrier Services list window, and select the **Details** icon. The Carrier Services window displays.
2. Enter information in the applicable fields. Refer to Table 34 for field value descriptions.
3. Choose the **Save** icon.

### Deleting a Carrier Service as a Routing Option:

#### About this task

To delete a carrier service:

#### Procedure

1. From the Routing Guidelines Details window, in the Carrier Services panel, select a carrier service in the Carrier Services list window and select the **Delete** icon. The Carrier Services window displays.
2. Choose the **Save** icon.

## **Modifying a Routing Guide Line**

### **About this task**

To modify a routing guide line:

#### **Procedure**

1. From the Routing Guidelines Details window, select the Routing Details Tab. A Routing Guide Line search window displays.
2. Select a routing guide line in the Routing Guide Line list window, and select the **Modify** icon. The Routing Guide Line Details window displays.
3. Enter the new information in the applicable fields. Refer to the "Routing Guide Line Details" table for field value descriptions.
4. Choose the **Save** icon.

## **Deleting a Routing Guide Line**

### **About this task**

To delete a Routing Guide Line:

#### **Procedure**

1. From the Routing Guide Lines Details window, select the Routing Details Tab. A Routing Guide Line search window displays.
2. Select a routing guide line in the Routing Guide Line list window, and choose the **Delete** icon.

## **Deleting a Routing Guide**

### **About this task**

To delete a routing guide:

#### **Procedure**

1. From the tree in the application rules side panel, choose Cross Application > Logistics > Outbound Constraints. The Outbound Constraints window displays in the work area.
2. Select the applicable Routing Guide and choose the **Delete** icon.

---

## Chapter 19. Time-Triggered Transaction Reference

---

### Time-Triggered Transaction Reference

Sterling Selling and Fulfillment Foundation provides a collection of time-triggered transactions, which are utilities that perform a variety of individual functions, automatically and at specific time intervals.

Time-triggered transactions perform repetitive actions on a scheduled basis, typically performing database updates, raising events, or calling APIs. One type of transaction, monitors, are designed to watch for processes or circumstances that are out of bounds and then raise alerts. Often, but not always, they retrieve tasks from the task queue or work from the pipeline.

Some transactions enable you to collect statistical data regarding the application's health. This data is collected periodically, using the value specified for the `yantra.statistics.persist.interval` attribute in the `yfs.properties` file. By default, statistics collection set to on. To override this property, add an entry in the `<INSTALL_DIR>/properties/customer_overrides.properties` file. For additional information about overriding properties using the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

For more information about statistics persistence, see the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*. For more information about the specific statistics parameters used, see the applicable time-triggered transactions.

The time-triggered transactions described in this chapter are unique transactions, that may or may not be document type specific. For document specific transactions, the nomenclature helps define which unique transaction it is based on: a transaction ID is in the format `Unique_Transaction_ID.Document_Type_Code`. For example, the transaction ID for Purge Return is `PURGE.0003`, indicating that it is based on the unique transaction `PURGE`, for document type `0003`, which is Return Order. Therefore, in order to be able to configure Purge Return, you should look for the `PURGE` transaction ID in this chapter, which is Order Purge.

Sterling Selling and Fulfillment Foundation provides the following types of time-triggered transactions:

- Business Process Time-Triggered Transactions - responsible for processing
- Time-Triggered Purge Transactions - clear out data that may be discarded after having been processed
- Task Queue Syncher Time-Triggered Transactions - update the task queue repository with the latest list of open tasks to be performed by each transaction, based on the latest pipeline configuration
- Monitors - watch and send alerts for processing delays and exceptions

Sterling Selling and Fulfillment Foundation tracks the following statistics for each time-triggered transaction:

- `ExecuteMessageCreated` - The number of jobs added to the JMS queue in a given time interval.
- `ExecuteMessageSuccess` - The number of jobs that were run successfully in a given time interval.

- `ExecuteMessageError` - The number of jobs that failed to run in a given time interval.
- `GetJobsProcessed` - The number of `GetJob` messages that were processed in a given time interval.

**Note:** Some of the statistics collected and tracked in Release 9.1 for time-triggered transactions, monitors, and integration and application servers may change with the next release of Sterling Selling and Fulfillment Foundation.

---

## Running Time-Triggered Transactions

All time-triggered transactions are threadable. This means that you can run multiple instances of a transaction within a single process. For more information about running time-triggered transactions, see the *Sterling Selling and Fulfillment Foundation: Installation Guide*. For more information about fine-tuning system performance while running them concurrently, see the *Sterling Selling and Fulfillment Foundation: Performance Management Guide*.

### Steps to Complete Before Scheduling Time-Triggered Transactions

#### About this task

Before running and scheduling a time-triggered transaction, ensure that you have completed the following:

#### Procedure

1. Configure a JMS Connection Factory to correlate with the QCF name configured for the time-triggered transaction. The Sterling Selling and Fulfillment Foundation factory defaults include the `AGENT_QCF` as the JMS Connection Factory. For more information about configuring JMS, see the documentation for your specific application server.
2. Configure JMS Server Destinations to correlate with the group or individual name of the time-triggered transaction. The Sterling Selling and Fulfillment Foundation factory defaults include the `DefaultAgentQueue` as the server destination.  
Do not put a dot (.) in the name of a JMS Server Destination, for example, 'A.0001'. If you do, Sterling Selling and Fulfillment Foundation is unable to communicate with it.
3. Using the Applications Manager, configure each time-triggered transaction required for your business process as described in the section entitled "Defining Transactions" in the *Sterling Selling and Fulfillment Foundation: Configuration Guide*. Each set of time-triggered transaction criteria parameters must ensure the appropriate association of a JMS Agent Server.

---

## Configuring Communication Between an Agent and a JMS Server

#### About this task

Setting up communication between an agent (time-triggered transaction) and a remote JMS server requires that you do some prerequisite setup on your JMS system, then do some configuration within the application, which consists of the following procedures:

- If an initial context factory code for your JMS system is not provided with the application, you must create one. See “Create an Initial Context Factory Code” for the list of codes that are provided.
- Defining the transaction details – the time-triggered transaction, or agent, must be edited to include connection information for your JMS system and the initial context factory you create. See “Define the Transaction Information” on page 88.

For more information about time-triggered transactions and how they fit into the larger picture of application business process modeling, see the *Configuring Process Models* chapter. Also see the *Configuring Alert Queues* chapter for additional information about queues and agents.

## Prerequisites

### About this task

Before starting, complete these tasks for your JMS Server. See your JMS Server documentation for more information about performing these tasks.

### Procedure

1. Configure the JMS Queue Connection Factory (QCF) and queues on your JMS server.
2. Configure the JNDI representation of the queues on your JMS server.  
Ensure that you have the following information available from these tasks:
  - JNDI name for each queue
  - JNDI QCF lookup
  - JMS location - the provider URL for the JMS server

### Results

Once you have completed the preceding tasks, complete the next two procedures in the order shown. These are both done in the application.

## Create an Initial Context Factory Code

### About this task

Using an Initial Context Factory (ICF) class enables remote Java clients to connect to your application. This class is provided by the application vendor. The application uses ICF codes to identify these when setting up agents. Initial context factory codes are predefined in the application for the following JMS vendors:

- IBM WebSphere® MQ (for MQSeries® accessed through a IBM WebSphere Internet Inter-ORB Protocol URL)
- File (for MQSeries accessed through a file URL, as with Oracle WebLogic)
- Oracle WebLogic (for WebLogic JMS)
- JBoss (for JBoss JMS)

If you are using a JMS server that is **not** in the preceding list (for example, ActiveMQ), you must create an initial context factory code for it in the application:

### Procedure

1. Open the Applications Manager. From the tree in the application rules side panel, choose System Administration > Initial Context Factory Codes. The Initial Context Factory Codes window displays in the work area.

2. Select the + icon to create a new initial context factory code. The Initial Context Factory window is displayed.
3. In the Initial Context Factory field, enter the name of the class provided by your JMS vendor. For example, for ActiveMQ, the class name is `org.apache.activemq.jndi.ActiveMQInitialContextFactory`.
4. In the Short Description field, enter a descriptive name, up to 40 characters. Make note of this name, because you will use it in the next procedure (see "Define the Transaction Information"). For ActiveMQ, enter **ActiveMQ**.
5. In the Long Description field, enter a more detailed description for the initial context factory, up to 100 characters.
6. Save the new initial context factory code and close the window.

## Results

For more information about ICFs, see *Creating an Initial Context Factory Code*.

## Define the Transaction Information

### About this task

For the JMS server to communicate with the application, there must be a time-triggered transaction configured with the JMS server and ICF information.

### Procedure

1. Open the Applications Manager. From the tree in the application rules side panel, double-click Process Modeling. The Process Modeling window displays in the work area.
2. Select the desired tab, then Base Document Type, then double-click Process Type.
3. Double-click the transaction that corresponds to the agent to be run.
4. Select the Time Triggered tab.
5. Create or select an existing Agent Criteria Definition to edit.
6. The Agent Criteria Details screen is displayed. Select the Runtime Properties tab.
7. Select an existing Agent Server from the list or create your own (recommended).
8. Select an existing Alert Queue from the list or create your own.
9. In the JMS Queue Name field, enter the JNDI name for the queue that you created. See "Prerequisites" on page 87.
10. Enter the desired number of threads the agent should run (recommended not to exceed 5 threads - if more than 5 are needed, start another agent in its own JVM).
11. Select the Initial Context Factory code you created. See "Create an Initial Context Factory Code" on page 87.
12. In the QCF Lookup field, enter the JNDI QCF lookup for the queue that you created (this is the Queue Connection Factory created for the applicable JMS Server). See "Prerequisites" on page 87.
13. Enter the Provider URL. This is the location where the JMS system resides, and is JMS vendor specific.



14. Select whether the agent should trigger itself (recommended) and at what interval (in minutes) or use an external trigger (triggeragent.sh in the `<install_dir>/install/bin` directory).
15. See Setting up the JMS Security Properties for information about setting the JMS Security option.
16. Leave the Criteria Parameters tab values at the default values.
17. Save the Agent Criteria Details and close the window.
18. Launch the agent in its own JVM by executing the `startagentserver.sh/cmd` script in the `<install_dir>/install/bin` directory.

## Results

For additional information about defining transactions and about this procedure, see the sections *Defining Transactions* and *Specifying a Transaction as Time-Triggered* in the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

---

## Business Process Time-Triggered Transactions

All Business Process Time-Triggered Transactions have a `CollectPendingJobs` criteria parameter. If this parameter is set to `N`, the agent does not collect information about the pending jobs pertaining to this monitor. This pending job information is used for monitoring the monitor in the System Management Console.

By default, `CollectPendingJobs` is set to `Y`. It can be helpful to set it to `N` if one particular time-triggered transaction is performing a significant amount of `getPendingJobs` queries, and the overhead cost is too high.

## Asynchronous Request Processor

This transaction completes any API request or service request in offline mode. It picks up the API messages or service messages from the `YFS_ASYNC_REQ` table and invokes the corresponding API or service. The messages can be inserted into the `YFS_ASYNC_REQ` table using the `createAsyncRequest` API.

### Attributes

Following are the attributes for this time-triggered transaction:

*Table 35. Asynchronous Request Processor Attributes*

Attribute	Value
Base Transaction ID	ASYNC_REQ_PROCESSOR
Base Process Type	General
Abstract Transaction	No

## Criteria Parameters

Following are the criteria parameters for this transaction:

*Table 36. Asynchronous Request Processor Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Lead Days	Number of days before the present date the agent will purge the records. If left blank or specified as 0 (zero), it defaults to 30.
Maximum Error Count	Maximum number of times the record is processed if an exception is thrown. Once the number of unsuccessful attempts equals this number, that record is not processed further by the agent. If left blank or specified as 0 (zero), it defaults to 20.
Reprocess Interval In Minutes	Time in minutes after which the transaction will be reprocessed - after it has been processed and has thrown an exception.
ColonyID	Required in a multischema deployment where the YFS_ASYNC_REQ table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

None

## Pending Job Count

None

## Events Raised

The following events are raised by this time-triggered transaction:

*Table 37. Events Raised by the Asynchronous Request Processor*

Transaction/Event	Key Data	Data Published*	Template Support?
HAS_EXCEPTIONS	None	YCP_ASYNC_REQ_PROCESSOR.HAS_EXCEPTIONS.html	Yes
<p>*These files are located in the following directory: &lt;INSTALL_DIR&gt;/xapidocs/api_javadocs/XSD/HTML</p>			

## Case Insensitive Data Loader

The Case Insensitive Data Loader agent migrates data from columns marked CaseInsensitiveSearch to shadow columns. The agent uses the transaction criteria to identify the records that need to be updated and then converts the original

column values to lowercase values in the shadow columns. For more information about enabling case insensitive searches, refer to the *Sterling Selling and Fulfillment Foundation: Extending the Database*.

The Case Insensitive Data Loader agent is required for updating the existing data. Once the shadow columns have been created, the Case Insensitive Data Loader agent only needs to be run once for each table or table type. The shadow columns are then populated in real-time by the application.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 38. Case Insensitive Data Loader Attributes*

Attribute	Value
Base Transaction ID	DATA_LOADER
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 39. Case Insensitive Data Loader Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. <ul style="list-style-type: none"> <li>If left blank or the number specified is less than 10000, it defaults to 5000.</li> <li>If the number specified is greater than 10000, then that value is used.</li> </ul>
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
TableType	Required in a multischema deployment when a table may exist in multiple schemas.  Valid Values: CONFIGURATION, TRANSACTION, MASTER.  If set to CONFIGURATION, the agent runs for the records associated with tables that have TableType as CONFIGURATION.  If set to TRANSACTION, the agent runs for the records associated with tables that have TableType as TRANSACTION.
Table Name	Required. The table name for the records to be migrated to shadow columns.

Table 39. Case Insensitive Data Loader Criteria Parameters (continued)

Parameter	Description
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

None.

### Pending Job Count

None.

### Events Raised

None.

## Change Load Status

This transaction is equivalent to the `changeLoadStatus()` API. For detailed information about this transaction, see the *Sterling Selling and Fulfillment Foundation: Javadocs*.

To be configured as part of your load processing pipeline, this transaction can be used whenever an automatic change in the status of a load is required. This automatic change could represent exporting load information to load planning software or transmission to the load's carrier.

This transaction should be configured to work from the task queue.

### Attributes

The following are the attributes for this time-triggered transaction:

Table 40. Change Load Status Attributes

Attribute	Value
Base Transaction ID	CHANGE_LOAD_STATUS
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	Yes
APIs Called	<code>changeLoadStatus()</code>

### Criteria Parameters

The following are the criteria parameters for this transaction:

Table 41. Change Load Status Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.

Table 41. Change Load Status Parameters (continued)

Parameter	Description
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 42. Change Load Status Statistics

Statistic Name	Description
NumLoadsChanged	Number of loads whose status was changed.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the CurrentDate value in the YFS\_Task\_Q table.

## Events Raised

This transaction raises events as specified under the changeLoadStatus() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Change Shipment Status

This transaction is equivalent to the changeShipmentStatus() API. For detailed information about this transaction, see the *Sterling Selling and Fulfillment Foundation: Javadocs*.

To be configured as part of your shipment processing pipeline, this transaction can be used whenever an automatic change in the status of a shipment is required. For example, this automatic change could represent exporting shipment information to a warehouse management system or to transmit an Advance Shipping Notice to the buyer.

This transaction should be configured to work from the task queue.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 43. Change Shipment Status Attributes

Attribute	Value
Base Transaction ID	CHANGE_SHIPMENT_STATUS
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	Yes
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 44. Change Shipment Status Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 45. Create Chained Order Statistics

Statistic Name	Description
NumShipmentsChanged	Number of shipments whose status was changed.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

## Events Raised

This transaction raises events as specified under the changeShipmentStatus() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Close Delivery Plan

To boost system performance, this transaction serves as a temporary purge until the Delivery Plan Purge deletes delivery plan-related data (see “Delivery Plan Purge” on page 164).

This transaction picks all delivery plans that do not have any of their loads or shipments still open and marks the deliveryplan\_closed\_flag='Y'. This flag indicates no further operations are possible on the plan.

This transaction corresponds to the base transaction close delivery plan (CLOSE\_DELIVERY\_PLAN) in the load pipeline.

Any enterprise using the Console must schedule purge jobs.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 46. Close Delivery Plan Attributes

Attribute	Value
Base Transaction ID	CLOSE_DELIVERY_PLAN
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 47. Close Delivery Plan Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 48. Close Delivery Plan Statistics

Statistic Name	Description
NumDeliveryPlansClosed	Number of delivery plans closed.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 49. Events Raised by Close Delivery Plan Transaction

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	delivery_plan_dbd.txt	YDM_CLOSE_DELIVERY_PLAN.ON_SUCCESS.xml	Yes

However, note that the template name would read <TransactionId>.ON\_SUCCESS.xml.

## Close Load

To boost system performance, this transaction serves as a temporary purge until the Load Purge deletes load-related data (see “Load Purge” on page 178).

This transaction corresponds to the base transaction Close Load (CLOSE\_LOAD) in the load pipeline.

If you use the Load processing pipeline, you must schedule this transaction. Only closed loads are picked up by the purge transaction. Therefore, it is required that this transaction be made part of the pipeline and scheduled to run at the end of the day.

This transaction should be made part of the pipeline. In addition, it should be configured to work from the task queue.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table 50. Close Load Attributes*

Attribute	Value
Base Transaction ID	CLOSE_LOAD
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 51. Close Load Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.



## Statistics Tracked

The following statistics are tracked for this transaction:

Table 52. Close Load Statistics

Statistic Name	Description
NumLoadsClosed	Number of loads closed.

## Pending Job Count

For this transaction the pending job count is the number of open delivery plans, which are not associated to any open loads and open shipments.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 53. Events Raised by the Close Load Transaction

Transaction/Event	Data Published	Template Support?
ON_SUCCESS	YDM_CLOSE_LOAD_PLAN.ON_SUCCESS.xml	Yes

However, note that the template name would read `<TransactionId>.ON_SUCCESS.xml`.

## Close Manifest

This time-triggered transaction sets the manifest's `MANIFEST_CLOSED_FLAG` flag to 'Y' and updates the manifest status to `CLOSED`. This time-triggered transaction confirms all the shipments that are pending confirmation, and closes the manifest.

**Note:** If the Close Manifest Agent is triggered without any criteria, it closes all the candidate manifests across all `ShipNodes`.

The `yfs.closemanifest.online` property in the `yfs.properties_ysc_ext.in` file is used to set this time-triggered transaction to work in online or offline mode.

- **Online mode:** In the online mode, the close manifest transaction runs as usual, confirming all shipments in the manifest and then closing the manifest.
- **Offline mode:** In the offline mode, the close manifest transaction triggers an agent and changes the manifest status to 'Closure Requested'. When the agent runs, it confirms either each shipment of the manifest, or closes the manifest, in an execution call.

The mode of operation (online or offline) is decided on the basis of the value specified for the `yfs.closemanifest.online` property in the `yfs.properties_ysc_ext.in` file. To override this property, add an entry for it in the `<INSTALL_DIR>/properties/customer_overrides.properties` file. For additional information about overriding properties using the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.

The default out-of-the-box shipped property causes the Close Manifest transaction to run in online mode.

In instances where the Close Manifest transaction is run in offline mode, ensure that all Agent Criteria defined for the transaction are configured properly.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 54. Close Manifest Attributes*

Attribute	Value
Base Transaction ID	CLOSE_MANIFEST
Base Document Type	General
Base Process Type	Manifesting
Abstract Transaction	No
APIs Called	confirmShipment()

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 55. Close Manifest Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
AgentCriteriaGroup	Optional. Used to classify nodes. This value can be accepted by Sterling Selling and Fulfillment Foundation time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ShipNode	Optional. Ship node for which the Close Manifest needs to be run. If not passed, then all ship nodes are monitored.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following are statistics are tracked for this transaction:

*Table 56. Close Manifest Statistics*

Statistic Name	Description
NumShipmentsConfirmed	Number of shipments confirmed.
NumManifestsClosed	Number of manifests closed.
NumManifestsErrored	Number of manifests errored.
NumShipmentsErrored	Number of shipments errored.

## Pending Job Count

For this transaction the pending job count is the sum of open manifests and shipments belonging to manifests (with MANIFEST\_STATUS='1200').

## Events Raised

The following events are raised by this time-triggered transaction:

Table 57. Events Raised by the Close Manifest Transaction

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	manifest_dbd.txt	YDM_CLOSE_MANIFEST .ON_SUCCESS.xml	Yes

## Close Order

This time-triggered transaction sets the order's ORDER\_CLOSED flag to 'Y' and raises the ON\_SUCCESS event. These actions are only performed when the entire ORDER\_QTY for all the order lines reaches the configured pickup status. If an order has ORDER\_CLOSED set to 'Y', it is not picked up for monitoring.

The Close Order agent must be configured along with the Purge transaction in the pipeline.

Many of this transaction's elements and attributes are template-driven. Refer to the XML for element level details.

The Close Order agent must be run before running the Monitor agent in order to avoid alerts getting raised for cancelled orders.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 58. Close Order Attributes

Attribute	Value
Base Transaction ID	CLOSE_ORDER
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 59. Close Order Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.

Table 59. Close Order Criteria Parameters (continued)

Parameter	Description
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 60. Close Order Statistics

Statistic Name	Description
NumOrdersProcessed	Number of orders processed.
NumOrdersClosed	Number of orders closed.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table, if tasks on hold are not ready to be processed.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 61. Events Raised by the Close Order Transaction

Transaction/Event	Data Published	Template Support?
ON_SUCCESS	YFS_CLOSE_ORDER.ON_SUCCESS.xml	Yes

## Close Receipts

This time-triggered transaction closes receipts using the receiving rule specified.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 62. Close Receipts Attributes

Attribute	Value
Base Transaction ID	RECEIPT_COMPLETE
Base Document Type	Order
Base Process Type	Receipt (Purchase Order Receipt, Return Receipt, Transfer Order Receipt, Receipt)
Abstract Transaction	No

Table 62. Close Receipts Attributes (continued)

Attribute	Value
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 63. Close Receipts Criteria Parameters

Parameter	Description
Action	Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Enterprise for which the Close Receipts needs to be run. If not passed, then all enterprises are monitored.
Node	Node for which the Close Receipts needs to be run. <b>Note:</b> Both <b>Node</b> and <b>AgentCriteriaGroup</b> cannot be blank. You must specify a value for the <b>Node</b> parameter, the <b>AgentCriteriaGroup</b> parameter, or both parameters.
AgentCriteriaGroup	Used to classify nodes. This value can be accepted by Sterling Selling and Fulfillment Foundation time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 64. Close Receipts Statistics

Statistic Name	Description
NumReceiptsClosed	Number of receipts closed.

## Pending Job Count

For this transaction the pending job count is the number of Receipts that can be closed (with OPEN\_RECEIPT\_FLAG='Y').

## Events Raised

The following events are raised by this time-triggered transaction:

*Table 65. Events Raised by the Close Receipts Transaction*

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	receipt_dbd.txt	YFS_RECEIPT_COMPLETE .ON_SUCCESS.xml	Yes

## Close Shipment

To boost system performance, this transaction serves as a temporary purge until the Shipment Purge deletes all shipment-related data (see “Shipment Purge” on page 215).

This transaction picks all shipments eligible to be closed, based on the pipeline configuration for pickup for transaction CLOSE\_SHIPMENT, and marks the shipment\_closed\_flag='Y'. This flag indicates no further operations are possible on the shipment. There is no status change involved. This transaction can be configured in the pipeline so that it picks up either Shipped or Delivered status.

This transaction corresponds to the base transaction close shipment (CLOSE\_SHIPMENT) in the shipment pipeline.

This transaction should be made part of the pipeline. In addition, it should be configured to work from the task queue.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 66. Close Shipment Attributes*

Attribute	Value
Base Transaction ID	CLOSE_SHIPMENT
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 67. Close Shipment Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

Table 67. Close Shipment Criteria Parameters (continued)

Parameter	Description
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following are statistics are tracked for this transaction:

Table 68. Close Shipment Statistics

Statistic Name	Description
NumShipmentsClosed	Number of shipments closed.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 69. Events Raised by the Close Shipment Transaction

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	shipment_dbd.txt	YDM_CLOSE_SHIPMENT. ON_SUCCESS.xml	Yes

## Collect Shipment Statistics

Collect Shipment Statistics is a time-triggered transaction which can be invoked to process the shipments, and generate information required for the Daily Shipment Report.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 70. Collect Shipment Statistics Attributes

Attribute	Value
Transaction Name	Collect Shipment Statistics
Transaction ID	COLLECT_STATISTICS
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None

Table 70. Collect Shipment Statistics Attributes (continued)

Attribute	Value
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 71. Collect Shipment Statistics Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Node	Required. The ship node for which records are being processed.
AgentCriteriaGroup	Optional. Used to classify nodes. This value can be accepted by Sterling Selling and Fulfillment Foundation time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 72. Statistics for Collect Shipment Statistics

Statistic Name	Description
NumDaysStatisticsCollected	Number of days for which shipment statistics have been collected.

## Pending Job Count

For this transaction the pending job count is the number of days for which shipment statistics needs to be collected. The number of days is calculated as the difference (in days) between the current date and the last date when shipment statistics was collected.



## Events Raised

The following events are raised by this time-triggered transaction:

Table 73. Events Raised by the Collect Shipment Statistics Transaction

Transaction/Event	Data Published	Template Support?
ON_SUCCESS	YDM_COLLECT_STATISTICS.ON_SUCCESS.xml	No

## Consolidate Additional Inventory

The Consolidate Additional Inventory time-triggered transaction consolidates supply and demand from the YFS\_INVENTORY\_SUPPLY\_ADDNL and YFS\_INVENTORY\_DEMAND\_ADDNL tables. Consolidation is performed by summing up the quantities of additional supply and demand in the YFS\_INVENTORY\_SUPPLY and YFS\_INVENTORY\_DEMAND tables.

If no matching supply or demand is found, a new supply or demand is created with the sum quantity of the changes in the YFS\_INVENTORY\_SUPPLY\_ADDNL and YFS\_INVENTORY\_DEMAND\_ADDNL tables. After the changes are applied, the records in the YFS\_INVENTORY\_SUPPLY\_ADDNL and YFS\_INVENTORY\_DEMAND\_ADDNL tables that were used in the consolidation process, are deleted.

### Attributes

The following are the attributes for this time-triggered transaction:

Table 74. Consolidate Additional Inventory Attributes

Attribute	Value
Base Transaction ID	CONSOLIDATE_ADDNL_INV
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the parameters for this transaction:

Table 75. Consolidate Additional Inventory Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of inventory item records (whose additional supplies and demands are consolidated) to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

Table 75. Consolidate Additional Inventory Criteria Parameters (continued)

Parameter	Description
ColonyID	Required in a multischema deployment where the YFS_INVENTORY_SUPPLY_ADDNL and YFS_INVENTORY_DEMAND_ADDNL tables may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 76. Consolidate Additional Inventory Statistics

Statistic Name	Description
NumInventorySupplyAddnlsProcessed	Number of additional inventory supply records processed in the consolidation.
NumInventoryDemandAddnlsProcessed	Number of additional inventory demand records processed in the consolidation.
NumInventoryDemandDtlsProcessed	Number of inventory demand details records processed in the consolidation.

## Pending Job Count

For this transaction the pending job count is the number of distinct inventory items in the YFS\_INVENTORY\_SUPPLY\_ADDNL and YFS\_INVENTORY\_DEMAND\_ADDNL tables, multiplied by two.

## Events Raised

None.

## Consolidate To Shipment

This is a task queue based transaction in the order pipeline that corresponds to base transaction CONSOLIDATE\_TO\_SHIPMENT. This transaction finds a shipment into which a given order release can be included. If it finds an existing shipment, it calls changeShipment() API. Otherwise, it calls the createShipment() API.

To find the existing shipments it matches ShipNode, ShipTo Address, SellerOrganizationCode, Carrier, DocumentType and so forth, of the Order Release with that of existing shipments.

This transaction is applicable only to the shipments in one of the following Statuses:

- Shipment Created
- ESP Check Required
- On ESP Hold
- Released from ESP Hold
- Released For Routing
- Awaiting Routing

- Shipment Routing
- Sent To Node

To successfully consolidate an Order Release to an existing shipment, the Add Line and related modification types on shipment in its current status should be allowed.

This transaction is a part of the Order Fulfillment pipeline. In addition, it should be configured to work from the task queue.

Order releases with GIFT\_FLAG set to Y are never consolidated with any other release.

For more information, see the details provided under the `createShipment()`, `changeShipment()`, `releaseOrder()`, and `consolidateToShipment()` APIs in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 77. Consolidate to Shipment Attributes

Attribute	Value
Base Transaction ID	CONSOLIDATE_TO_SHIPMENT
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	<code>createShipment()</code> and <code>changeShipment()</code>
User Exits	<ul style="list-style-type: none"> <li>• It calls <code>beforeConsolidateToShipment</code> in <code>com.yantra.ydm.japi.ue</code>.</li> <li>• <code>YDMBeforeConsolidateToShipment</code> for each release before it begins processing.</li> <li>• After it finds the shipments, it calls <code>determineShipmentToConsolidateWith</code> in <code>com.yantra.ydm.japi.ue</code>.</li> </ul> <p style="text-align: center;"><code>YDMDetermineShipmentToConsolidateWith</code>.</p> <p>For more information, see the <i>Sterling Selling and Fulfillment Foundation: Javadocs</i>.</p>

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 78. Consolidate to Shipment Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

Table 78. Consolidate to Shipment Criteria Parameters (continued)

Parameter	Description
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 79. Consolidate to Shipment Statistics

Statistic Name	Description
NumOrderReleasesConsolidated	Number of order releases consolidated.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 80. Events Raised by the Consolidate to Shipment Transaction

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	shipment_dbd.txt	YDM_CONSOLIDATE_TO_SHIPMENT.ON_SUCCESS.xml	Yes

This transaction also raises events as specified under the createShipment() and changeShipment() APIs in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

However, note that the template name would read <TransactionId>.ON\_SUCCESS.xml.

## Create Catalog Index

The Create Catalog Index transaction builds the Apache Lucene index file that is used by catalog search. This index file enhances search performance by storing denormalized item data that has been extracted from the Sterling Selling and Fulfillment Foundation database or from an external source.

The Create Catalog Index transaction can be configured to perform the following tasks:

- Run either a scheduled index build or user-initiated index build
- Build either a full or incremental index file
- Activate the index file

## The Index Building Process

The Create Catalog Index transaction provides an agent for index building. Index building is a multithread process in which the index building agent extracts item and item-related information from the active selling catalog in the Sterling Selling and Fulfillment Foundation database. If the corresponding XML configuration file has been extended, the agent may extract this information from an external source.

The agent writes this information to multiple files, which identify the item data that should be included in the final index. After the agent finishes writing the files, it merges them into the final index file.

The multithread process provides the advantage of parallel processing. Large amounts of database data are segmented and processed simultaneously, which is faster and more scalable than sequentially processing one long file.

When writing information to multiple files, the index building agent performs the following tasks for each item before looping to the next item:

- Queries the Sterling Selling and Fulfillment Foundation database or an external source for data about the item.
- Uses information from the XML configuration file and extension file to determine the data that be retrieved from the query.
- Retrieves relevant data from the Sterling Selling and Fulfillment Foundation database.
- Creates a Lucene document for the item.

After the transaction creates a Lucene document for each item, the transaction writes the documents to the index file based on the organization and the organization's locales.

## Configuration Options for Accessing Catalog Index Files

You can configure catalog index builds in one of the following two ways, depending on your business requirements:

- Build the index on a shared, central disk that is accessible from all servers.
  - Advantages:
    - Centralized control of shared index
    - No file transfer issues because the index is not copied across multiple servers
  - Limitation:
    - Shared disk could become a single point of failure (if no redundancy is involved)
    - Volume of reads and writes from shared disk might slow performance, depending on the setup
- Build and push a copy of the index to multiple servers via file transfer. Automate this file transfer process to occur on completion of an index build, but do not automatically activate the index. When all servers have acknowledged the completion of the file transfer, call the `manageSearchIndexTrigger` API to activate the index.
  - Advantage:
    - No central point of failure
  - Limitation:

- Possible overhead to building and pushing index files across servers  
If you choose this method of building the index in one location and reading it from another, refer to the *Sterling Selling and Fulfillment Foundation: Properties Guide* for information about enabling different properties for individual processes.

For more information about building and searching catalog indexes, see the *Sterling Selling and Fulfillment Foundation: Catalog Management Concepts Guide*.

## Attributes

The following table displays the attributes for the Create Catalog Index transaction.

Table 81. Create Catalog Index Attributes

Attribute	Value
Base Transaction ID	Create_Catalog_Index
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YCMParseAssetUE YCMGetAdditionalCatalogIndexInformationUE

## Criteria Parameters

The following table displays the criteria parameters for the Create Catalog Index transaction.

Table 82. Create Catalog Index Criteria Parameters

Parameter	Description
Organization Code	Required. The organization code of the catalog organization or subcatalog organization that maintains the search index.
Number of Messages	Required. Number of messages to use when building the index file.  Sterling Selling and Fulfillment Foundation processes only one message per thread. For example, if Number of Messages is set to 10 and Threads is set to 3, Sterling Selling and Fulfillment Foundation processes only 3 messages at a time. For more information about fine-tuning system performance, see the <i>Sterling Selling and Fulfillment Foundation: Performance Management Guide</i> .

Table 82. Create Catalog Index Criteria Parameters (continued)

Parameter	Description
<b>Incremental Build</b>	<p>Y or N.</p> <p>Y to rebuild the existing index file. If you specify Y, Sterling Selling and Fulfillment Foundation rebuilds the index based on the last successful index build. The MaxModifyTS column in the YFS_ITEM table determines whether or not an item's attributes have changed. If any external attributes of an item have changed, update the MaxModifyTS column by calling the manageItem API on the item.</p> <p>N to build a full index file.</p> <p>This parameter is ignored for user-initiated index builds. However, if scheduled builds are configured, ensure that you specify whether you want a full or incremental index build.</p>
<b>Category Domain</b>	<p>Optional. The catalog from which the index is built. The active selling catalog of the catalog organization or subcatalog organization is the default. If scheduled builds are configured, ensure that you specify a catalog.</p>
<b>Auto Activate</b>	<p>Y or N. Optional.</p> <p>Y to activate the index after building the index file.</p> <p>The default is N.</p>
<b>Auto Insert Search Index Trigger</b>	<p>Y or N. Optional.</p> <p>Y to enable scheduled builds of the catalog index file. The agent refers to information stored in the YFS_SEARCH_INDEX_TRIGGER table to determine when to run the scheduled index build. Specify the type of index build, whether full or incremental, in the agent criteria.</p> <p>N to enable user-initiated builds of the catalog index file. The agent continuously queries the YFS_SEARCH_INDEX_TRIGGER table to determine whether an index build is indicated. If a user starts an index build from the IBM Sterling Business Center, the status setting in the table changes to Scheduled, triggering the agent to build the index. The user specifies the type of index build, whether full or incremental, from the Sterling Business Center.</p> <p>After a scheduled or user-initiated build runs, the user can activate the index from the Sterling Business Center. Alternatively, the agent can be configured to automatically activate the index.</p> <p>To allow both scheduled and user-initiated index builds, configure the transaction to include two instances of the agent. Configure one instance to trigger user-initiated builds and the second instance to trigger scheduled index builds.</p>
<b>ColonyID</b>	<p>Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.</p>

## Statistics Tracked

The following table shows the statistics for the Create Catalog Index transaction.

Table 83. Create Catalog Index Statistics

Statistic Name	Description
SearchIndicesBuilt	Number of search indices that have been built.

## Pending Job Count

None.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 84. Events Raised by the Create Catalog Index Transaction

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	Not Published	CATALOG_INDEX_BUILD.ON_SUCCESS.xml	Yes

## Create Chained Order

This transaction creates one or more chained orders from an order whose OrderHeaderKey is stored in the task queue object. Chainable lines of the order can also be added to existing chained orders, instead of creating new chained orders with these lines. The existing chained orders must be identified by the determineChainedOrderForConsolidation user exit. If the user exit is not implemented, or if the user exit returns a blank document, one or more new chained orders are created.

For more information about the creation of chained orders, see the information provided under the createChainedOrder() API and the YFSdetermineChainedOrderForConsolidation user exit in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

This transaction should be invoked after order scheduling.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 85. Create Chained Order Attributes

Attribute	Value
Base Transaction ID	CHAINED_ORDER_CREATE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	Yes
APIs Called	createChainedOrder()



## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 86. Create Chained Order Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 87. Create Chained Order Statistics

Statistic Name	Description
NumOrdersProcessed	Number of orders processed for creating chained order.
NumOrdersCreated	Number of chained orders created.

If there are 2 orders being processed and the first order creates a chained order, the DetermineChainedOrderForConsolidation user exit causes the lines of the 2nd order to be added to the first order. The number of chained orders created is counted as 2.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

## Events Raised

This transaction raises events as specified under the createChainedOrder() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Create Derived Order

This transaction creates one or more derived orders from an order whose OrderHeaderKey is stored in the task queue object. For existing derived orders, you can add derivable lines or create new derived orders with these lines. The existing derived orders must be identified by the determineDerivedOrderForConsolidation user exit. If the user exit is not implemented or if the user exit returns a null document, new derived orders are created. For more information about the creation of derived orders, see the details

provided under the `createDerivedOrder()` API and `YFSDetermineDerivedOrderForConsolidation` user exit in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 88. Create Derived Order Attributes*

Attribute	Value
Base Transaction ID	DERIVED_ORDER_CREATE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	Yes
APIs Called	<code>createDerivedOrder()</code>

The `TransactionKey` posted in the task queue object must be an instance of the Abstract Transaction `DERIVED_ORDER_CREATE` for the `ProcessType` associated with the Order. Otherwise, an exception is thrown.

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 89. Create Derived Order Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to <code>Get</code> , the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 90. Create Derived Order Statistics*

Statistic Name	Description
<code>NumOrdersProcessed</code>	Number of orders processed.
<code>NumOrdersCreated</code>	Number of derived orders created.

If there are 2 orders being processed and the first order creates a derived order, the `DetermineChainedOrderForConsolidation` user exit causes the lines of the 2nd order to be added to the first order. The number of derived orders created is counted as 2.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

## Events Raised

This transaction raises events as specified under the createDerivedOrder() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Create Order Invoice

This transaction creates one or more invoices from an order whose OrderHeaderKey is stored in a task queue object. The createOrderInvoice() API is called for the OrderHeaderKey.

Configure this transaction in the pipeline only after all processing that can impact quantity or price has been completed. Post invoice creation, the line quantity cannot be reduced below the invoiced quantity.

Both the Create Order Invoice and Create Shipment Invoice transactions can create invoices for an Order. When configuring your pipeline, ensure that only *one* of these two transactions is configured to create invoices for a particular order line. For more information, see “Create Shipment Invoice” on page 116.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 91. Create Order Invoice Attributes

Attribute	Value
Base Transaction ID	CREATE_ORDER_INVOICE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	Yes
APIs Called	createOrderInvoice()

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 92. Create Order Invoice Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 93. Create Order Invoice Statistics

Statistic Name	Description
NumOrderInvoicesCreated	Number of order invoices created.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

## Events Raised

This transaction raises events as specified under the createOrderInvoice() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Create Shipment Invoice

Invoicing is mandatory if an order requires payment processing. Invoicing occurs if the following conditions are met:

- Invoicing is enabled at the document parameter level.
- The Seller requires payment processing.

This transaction creates one or more invoices for the shipment whose ShipmentKey is stored in the task queue object. The createShipmentInvoice() API is called for the ShipmentHeaderKey.

This transaction should be configured in the shipment pipeline only after the shipment has reached a shipped status.

Both the Create Order Invoice and Create Shipment Invoice can create invoices for an order. When configuring your pipeline, ensure that only *one* of these two transactions is configured to create invoices for a particular order line. See “Create Order Invoice” on page 115.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 94. Create Shipment Invoice Attributes

Attribute	Value
Base Transaction ID	CREATE_SHIPMENT_INVOICE
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	Yes
APIs Called	createShipmentInvoice()

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 95. Create Shipment Invoice Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 96. Create Shipment Invoice Statistics

Statistic Name	Description
NumShipmentInvoicesCreated	Number of shipment invoices created.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

## Events Raised

This transaction raises events as specified under the createShipmentInvoice() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## ESP Evaluator

The ESP Evaluator time-triggered transaction verifies whether a shipment meets certain economic shipping parameters (ESP). ESP can be configured either for buyer or enterprise, with the freight terms on the shipment determining which one is used.

If the configuration is defined to hold shipment for ESP, the shipment when created is held for ESP (with status *On ESP Hold*). This task queue based time-triggered transaction evaluates the shipment for ESP, and passes it on to the next step in the shipment pipeline if the criteria (weight and volume limits, plus maximum days of hold up) are met. The shipment status is now set to *Released from ESP hold*, and routing processing begins.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 97. ESP Evaluator Attributes

Attribute	Value
Base Transaction ID	ESP_EVALUATOR.0001
Base Document Type	Order
Base Process Type	Outbound Shipment
Abstract Transaction	No
APIs Called	None
User Exits Called	getNodeMinimumNotificationTime

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 98. ESP Evaluator Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
EnterpriseCode	Optional. Enterprise for which the ESP Evaluator needs to be run. If not passed, then all enterprises are monitored.
Number of Records to Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
Node	Required. The ship node for which records are being processed.
AgentCriteriaGroup	Optional. Used to classify nodes. This value can be accepted by Sterling Selling and Fulfillment Foundation time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

None.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 99. Events Raised by ESP Evaluator Transaction

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	shipment_dbd.txt	ESP_EVALUATOR.ON_SUCCESS.xml	Yes

## Item Based Allocation

The Item Based Allocation transaction allocates unpromised and promised demands of existing orders to more suitable supplies based upon inventory items and nodes which have been triggered for the Item Based Allocation process in the YFS\_IBA\_TRIGGER table.

The Item Based Allocation agent obtains and processes all Item Based Allocation triggers from the YFS\_IBA\_TRIGGER table that meet the following conditions:

- IBA\_RUN\_REQUIRED = "Y"
- LAST\_IBA\_PROCESSED\_TS was 'x' hours before current time, where 'x' is from the 'Item Based Allocation Agent Execution Interval (in hours)' rule in the Installation rules. For more information about installation rules, refer to the topic "System Administration Components: Defining Installation Rules" in the *Sterling Selling and Fulfillment Foundation: Configuration Guide*. This rule is used to indicate the interval that the Item Based Allocation agent should not reprocess the triggers in the YFS\_IBA\_TRIGGER table, which were processed earlier. This prevents the IBA agent from over-processing the item and node combination in the given time interval to avoid any high loads on the system.
- PROCESSING\_BY\_AGENT="N" or PROCESS\_OVER\_BY\_TS is before the current timestamp. The PROCESSING\_BY\_AGENT field is used to prevent the picking up of the IBA trigger which is being processed by another instance of the agent.

If InventoryOrganizationCode is specified in the agent criteria, only the IBA trigger with inventory items of that inventory organization is retrieved.

For each triggered item and node combination, the agent finds all of the applicable order lines or order line reservations that contain the item and node and tries to move their unpromised and promised demands to more suitable available supplies based on user-configured IBA selection rules or FIFO (First-In-First-Out) IBA selection rules.

Sterling Selling and Fulfillment Foundation creates new positive order line reservations with the matched supply's first ship date and negative order line reservations for the existing demand ship date. Once all orders are processed, they are placed on hold to be rescheduled if changes are detected in the order line reservations.

The following configuration is required for the Item Based Allocation process:

- The Use Item Based Allocation rule needs to be enabled.
- Item and node need to have Item Based Allocation Allowed enabled.

- A hold type is required to be set up for the change order line reservations modification type so that the order can be placed on hold for rescheduling. For more information, refer to the *Sterling Selling and Fulfillment Foundation: Javadocs*.

The 'When a line is backordered, backorder against the highest priority ship node' rule should be checked in order to reallocate backordered demand. For more information, see the Fulfillment Rules section in the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

Before processing the Item Based Allocation logic, the Item Based Allocation agent updates the following fields on the Item Based Allocation trigger:

- PROCESSING\_BY\_AGENT = "Y". This indicates that an instance of the agent is currently processing this trigger.
- PROCESS\_OVER\_BY\_TS = current time + 1 hr. This indicates the expected time that the agent should finish with processing this IBA trigger. One hour is the fixed window and cannot be changed. Sterling Selling and Fulfillment Foundation treats the PROCESSING\_BY\_AGENT flag as "N" regardless of the actual value when current timestamp is after this timestamp.
- IBA\_RUN\_REQUIRED = "N". This resets the IBA\_RUN\_REQUIRED flag back to "N".

### **Obtaining a List of Demands Based on Applicable Order Release Statuses and Order Line Reservations to be Allocated**

A list of demands is derived from applicable order release statuses and order line reservations, which have the item and node in the IBA trigger. The following types of demands are retrieved:

- Demands of chained orders
- Demands of orders with chained order already created
- Demands of orders with procurement node but chained order creation is not yet created
- Demands of orders without procurement node
- Demands from order line reservations

The demand quantity is derived based on the order release status quantity with the status from the Status Inventory Type configuration that has a demand type, which considers the supply type with 'Use Consider Demand Type for Item Based Allocation' enabled. For more information, refer to the *Sterling Selling and Fulfillment Foundation: Global Inventory Visibility Configuration Guide*.

### **Obtaining a List of Available Supplies for Allocation**

Sterling Selling and Fulfillment Foundation obtains the available supply based on the availability of the item at the node by ignoring unpromised and promised demands. If the inventory organization maintains its inventory externally, the external availability can be read by the YFSGetExternalInventoryUE user exit. Only the availability of supplies that consider the 'Demand Type Look for Availability during Item Based Allocation' are used in the allocation logic. For more information, refer to the *Sterling Selling and Fulfillment Foundation: Global Inventory Visibility Configuration Guide*.

Allocated demands should be matched with the same supplies as "Demand to look for during release".



## Matching Demands Against Supplies in FIFO (First-In-First-Out) Order

Sterling Selling and Fulfillment Foundation sorts the list of available supplies in the order of the first shippable date (ETA), and matches the obtained list of demands using the top-down logic (unlike the normal matching logic for obtaining availability, where matches are based on the closest ETA). Demands are allocated in the following orders:

- Demands of chained orders - first based on user-configured sequencing rules, and then in ascending order of order creation date. (These types of demands are matched based on the closest ETA to avoid any changes in the chained orders).
- Demands of orders with a chained order already created - first based on user-configured sequencing rules, then in ascending order of product availability date. (These types of demands are matched based on the closest ETA to avoid any changes in the orders).
- Demands of orders for which procurement node and chained order creation is imminent (within the advanced notification time window) - first based on user-configured sequencing rules, then in order of order creation date.
- Demands of orders without a procurement node and within the release window (advanced notification time window) - first based on user-configured sequencing rules, then in order of order creation date.
- Demands from order line reservations on the order lines in the order of requested reservation date, and leftover demands (outside of the advanced notification time window) of orders with or without a procurement node, first based on user-configured sequencing rules and then in the order of order creation date.
- Demands from inventory reservations in the order of ship date.

Notice that different types of demands are given different priorities based on their significance. The demands of chained orders or orders related to chained orders are treated with a higher priority than the demands of normal orders. Furthermore, the demands with a ship date within the advanced notification time window also have a higher priority than the demands with a date outside of the advanced notification time window.

## Updating Order Reservations for the Matched Demands

After matching the available supply and demand in user-configured sequencing and then in FIFO order, the system builds up a list of order line reservation changes and inventory demand changes (corresponding to the order line reservation changes) and summarize them to optimize the number of order reservation updates and inventory updates. Negative order line reservations are added for the matched demands. Positive order reservations are added for the matched demands with the product availability date set to the matched supplies' first ship date.

After the Item Based Allocation agent completes its tasks for an Item Based Allocation trigger, it updates the fields of the trigger with the following values:

- IBA\_REQUIRED = "N"
- LAST\_IBA\_PROCESSED\_TS = current timestamp.
- PROCESS\_OVER\_BY\_TS = current timestamp.
- PROCESSING\_BY\_AGENT = "N"

The Item Based Allocation agent should be used in conjunction with the rescheduling process as the rescheduling process reschedules the affected orders by utilizing the order line reservations created by the Item Based Allocation process.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 100. Item Based Allocation Attributes*

Attribute	Value
Base Transaction ID	ITEM_BASED_ALLOCATION
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	changeOrder – for updating the order line reservations created as part of the Item Based Allocation process.
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 101. Item Based Allocation Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
InventoryOrganization Code	The inventory organization code of the inventory items which are processed by the Item Based Allocation agent. If provided, only the IBA triggers with the inventory item that belongs to this inventory organization are processed.
ColonyID	Required in a multischema deployment where the YFS_IBA_TRIGGER table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 102. Item Based Allocation Statistics*

Statistic Name	Description
NumOrdersProcessed	Number of orders processed by the Item Based Allocation agent.
NumOrdersRequiredReschedule	Number of orders required rescheduling as the result of Item Based Allocation process.

## Pending Job Count

None.

## Events Raised

This transaction raises events as specified under the changeOrder API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Mark Load as Trailer Loaded

This is a time-triggered transaction which works on “Load pipeline”.

This time-triggered transaction gets records from the Task Q. This transaction is used to mark the load as trailer loaded when all containers for the load are on the trailer.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 103. Mark Load As Trailer Loaded Attributes*

Attribute	Value
Base Transaction ID	MARK_AS_TRAILER_LOADED
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 104. Mark Load As Trailer Loaded Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ReprocessInterval	Optional. Reprocess Interval is the time taken to reprocess the load.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 105. Mark Load As Trailer Loaded Statistics*

Statistic Name	Description
NumLoadsChanged	Number of trailer loads changed.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

## Events Raised

None.

## Match Inventory

Match Inventory processes all pending records in the YFS\_INVENTORY\_SHIPMENT table. Pending records have a smaller number in POSTED\_QUANTITY than in QUANTITY.

Each pending record is matched against the receipt records in YFS\_INVENTORY\_RECEIPT table by applying the inventory cost determination logic. The unit cost at which the sales and receipt data are matched is also posted in YFS\_INVENTORY\_MATCH table.

Use this transaction if any of the configured ship nodes maintain inventory cost.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 106. Match Inventory Attributes

Attribute	Value
Base Transaction ID	INVENTORY_MATCH
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 107. Match Inventory Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
InventoryOrganizationCode	Optional. Valid inventory owner organization. Organization to process in this run. If not passed, all inventory organizations are processed.
CutOffDate	Optional. If passed, records are matched up to this date. Defaults to all unmatched records in Database.

Table 107. Match Inventory Criteria Parameters (continued)

Parameter	Description
ColonyID	Required in a multischema deployment where the YFS_INVENTORY_SHIPMENT, YFS_INVENTORY_RECEIPT, and the YFS_INVENTORY_MATCH tables may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 108. Match Inventory Statistics

Statistic Name	Description
NumInventoryShipmentsProcessed	Number of inventory shipments processed.
NumInventoryMatchesInserted	Number of inventory matches inserted.

## Pending Job Count

For this transaction the pending job count is the number of distinct inventory items that exist in the YFS\_INVENTORY\_SHIPMENT table where the QUANTITY value is not equal to the POSTED\_QUANTITY value.

## Events Raised

None.

## Payment Collection

This transaction requests credit validation for orders that are pending authorization or charging.

Use this transaction for creating authorization and charge requests.

This transaction works in combination with the Payment Execution transaction. Although this transaction can run independent of that transaction, authorization and collection occurs *only* after the Payment Execution dependencies are met. For more details, see “Payment Execution” on page 127.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 109. Payment Collection Attributes for Sales Orders

Attribute	Value
Base Transaction ID	PAYMENT_COLLECTION
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	requestCollection()

Table 110. Payment Collection Attributes for Return Orders

Attribute	Value
Base Transaction ID	PAYMENT_COLLECTION.0003
Base Document Type	Order
Base Process Type	Reverse Logistics
Abstract Transaction	No
APIs Called	requestCollection()

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 111. Payment Collection Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. The enterprise for which the transaction needs to be run. If left blank, orders for all enterprises are processed. If specified, only orders for that enterprise are processed.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.
HoldTypeOnRollback	<p>If the HoldTypeOnRollback criteria is populated and the requestCollection agent throws an exception, for example, from the getFundsAvailable user exit, HoldTypeOnRollback will be used to put the order on hold. If using the old order hold functionality, this will be used as the hold reason. If the hold type does not exist, an exception is thrown.</p> <p>If the HoldTypeOnRollback criteria is not populated, the order will not be put on hold if an exception is thrown.</p>

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 112. Payment Collection Statistics

Statistic Name	Description
NumOrdersProcessed	Number of orders processed.
NumChargeReqsCreated	Number of charge requests created.
NumAuthorizationReqsCreated	Number of authorization requests created.

## Pending Job Count

For this transaction the pending job count is the number of orders in the appropriate payment statuses with the value of the AUTHORIZATION\_EXPIRATION\_DATE is less than or equal to (<=) the current date. The appropriate payment statuses for such orders are:

- AWAIT\_PAY\_INFO

- AWAIT\_AUTH
- REQUESTED\_AUTH
- REQUEST\_CHARGE
- AUTHORIZED, INVOICED
- PAID
- RELEASE\_HOLD
- FAILED\_AUTH
- FAILED\_CHARGE
- VERIFY
- FAILED

## Events Raised

The following events are raised by this time-triggered transaction:

*Table 113. Events Raised by the Payment Collection Transaction*

Transaction/Event	Key Data	Data Published	Template Support?
INCOMPLETE_PAYMENT_INFORMATION	modifyOrder_dbd.txt	YFS_PAYMENT_COLLECTION.INCOMPLETE_PAYMENT_INFORMATION.xml	Yes
PAYMENT_STATUS	YFS_PAYMENT_COLLECTION.PAYMENT_STATUS_dtd.txt	YFS_PAYMENT_COLLECTION.PAYMENT_STATUS.xml	Yes
REQUEST_PAYMENT_STATUS		YFS_PAYMENT_COLLECTION.REQUEST_PAYMENT_STATUS.xml	Yes
ON_LIABILITY_TRANSFER	modifyOrder_dbd.txt	YFS_PAYMENT_COLLECTION.ON_LIABILITY_TRANSFER.xml	Yes
ON_INVOICE_COLLECTION	order_dbd/txt	YFS_CREATE_ORDER_INVOICE.ON_INVOICE_COLLECTION.xml	Yes

## Payment Execution

This transaction processes all requests that are pending authorization and charging.

**Note:** If the charge transaction request has an open authorization reversal, the charges are not collected.

Use this time-triggered transaction for processing all authorization and charge requests.

This transaction requires interfacing with a product that provides financial services.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 114. Payment Execution Attributes for Sales Orders

Attribute	Value
Base Transaction ID	PAYMENT_EXECUTION
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	executeCollection()
User Exits Called	collectionCreditCard, collectionOthers, collectionCustomerAcct

Table 115. Payment Execution Attributes for Return Orders

Attribute	Value
Base Transaction ID	PAYMENT_EXECUTION.0003
Base Document Type	Order
Base Process Type	Reverse Logistics
Abstract Transaction	No
APIs Called	executeCollection()
User Exits Called	collectionCreditCard, collectionOthers, collectionCustomerAcct

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 116. Payment Execution Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ChargeType	Type of credit card process. Valid values are: <ul style="list-style-type: none"><li>• AUTHORIZATION - Validates the credit card account</li><li>• CHARGE - Applies the charge to the credit card</li></ul>
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 117. Payment Execution Statistics

Statistic Name	Description
NumAuthTransProcessed	Number of authorization transaction processed.



Table 117. Payment Execution Statistics (continued)

Statistic Name	Description
NumAuthTransSuccessfullyProcessed	Number of successful returns from user exit for authorization transaction processed.
NumChargeTransProcessed	Number of charge transaction processed.
NumChargeTransSuccessfullyProcessed	Number of successful returns from user exit for charge transaction processed.
NumCollectionValidations	Number of successful returns from the invoked validate collection user exits.
NumCreditCardCollections	Number of credit card collections.
NumCustomerAccountCollections	Number of successful returns from the customer account collection user exits.
NumOtherCollections	Number of successful returns from the other collection user exits.

## Pending Job Count

For this transaction the pending job count is the number of open charge and authorization transactions.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 118. Events Raised by Payment Execution Transaction

Transaction/Event	Key Data	Data Published	Template Support?
CHARGE_FAILED	modifyOrder dbd.txt	PAYMENT_EXECUTION_ CHARGE_FAILED_dbd.txt	No

This transaction raises events as specified under the `executeCollection()` API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Post Inventory Match

This transaction processes all open records in YFS\_INVENTORY\_MATCH table and posts the records to a financial system. An open record in the YFS\_INVENTORY\_MATCH table has the status of 01. After posting, the status is changed to 02.

Use this transaction if any of the configured ship nodes maintain inventory cost.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 119. Post Inventory Match Attributes

Attribute	Value
Base Transaction ID	POST_INVENTORY_MATCH
Base Document Type	General

Table 119. Post Inventory Match Attributes (continued)

Attribute	Value
Base Process Type	General
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 120. Post Inventory Match Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where the YFS_INVENTORY_MATCH table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 121. Post Inventory Match Statistics

Statistic Name	Description
NumInventoryMatchPosted	Number of inventory match records posted.

## Pending Job Count

For this transaction the pending job count is the number of inventory matches with an open status.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 122. Events Raised by the Post Inventory Match Transaction

Transaction/Event	Key Data	Data Published	Template Support?
POST_INVENTORY_MATCH	POST_INVENTORY_MATCH_dbd.txt	YFS_postInventoryMatch_output.xml	No

## Process Order Hold Type

You can create a time-triggered transaction, derived from the PROCESS\_ORDER\_HOLD\_TYPE abstract transaction. It can be configured as the processing transaction for one or more hold types. If an order is associated with a

hold type that has a transaction configured as the processing transaction, a record is created in the YFS\_TASK\_Q table for processing that transaction.

When the processing transaction is triggered, it checks the hold types that it can process based on the hold type configuration. If no hold types can be processed, the YFS\_TASK\_Q record is deleted. If some hold types can be processed, the processOrderHoldType user exit is invoked with the list of hold types to be processed. The processOrderHoldType user exit returns the list of hold types that can be removed from the order.

The transaction then modifies the order and updates the order hold type list based on the output returned by the processOrderHoldType user exit. If no hold types can be processed, the YFS\_TASK\_Q record is deleted. If some hold types can still be processed, YFS\_TASK\_Q is updated with the next available date.

You can also call the processOrderHoldType user exit to add new hold types or change the status of a hold type that is already applied to an order. For more information about the processOrderHoldType user exit, see the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 123. Process Order Hold Type Attributes

Attribute	Value
Base Transaction ID	PROCESS_ORDER_HOLD_TYPE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	Yes
APIs Called	changeOrder

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 124. Process Order Hold Type Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where the YFS_TASK_Q table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

None.

## Pending Job Count

None

## Events Raised

The following events are raised by this time-triggered transaction:

Table 125. Events Raised by Process Order Hold Type Transaction

Transaction/Event	Raised when...	Key Data	Data Published	Template Support?
ON_SUCCESS	On success	modifyOrder_dbd.txt	YFS_ORDER_CHANGE.ON_SUCCESS.xml	Yes *
ON_HOLD_TYPE_STATUS_CHANGE	The status of a hold type is changed.	modifyOrder_dbd.txt	YFS_ON_HOLD_TYPE_STATUS_CHANGE.xml	Yes
ON_ORDER_LINE_HOLD_TYPE_STATUS_CHANGE	The status of a hold type is changed.	modifyOrder_dbd.txt	YFS_ON_ORDER_LINE_HOLD_TYPE_STATUS_CHANGE.xml	Yes

\* Note: Some of the elements and attributes are not template-driven. Refer to the xml for element level details.

## Process Work Order Hold Type

This time-triggered transaction is identical to the Process Order Hold Type transaction, but it is used for work orders instead.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 126. Process Work Order Hold Type Attributes

Attribute	Value
Base Transaction ID	PROCESS_WO_ORDER_HOLD_TYPE
Base Document Type	Work Order
Base Process Type	VAS Process
Abstract Transaction	Yes
APIs Called	modifyWorkOrder

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 127. Process Work Order Hold Type Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.

Table 127. Process Work Order Hold Type Parameters (continued)

Parameter	Description
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

None.

## Pending Job Count

None

## Events Raised

The following events are raised by this time-triggered transaction:

Table 128. Events Raised by Process Work Order Hold Type Transaction

Transaction/Event	Raised when...	Key Data	Data Published	Template Support?
ON_SUCCESS	On success	workOrder_ dbd.txt	VAS_MODIFY_ WORK_ORDER .ON_SUCCESS. xml	Yes *
ON_HOLD_TYPE_ STATUS_ CHANGE	The status of a hold type is changed.	workOrder_  dbd.txt	VAS_ON_HOLD _TYPE_STATUS _CHANGE.xml	Yes
* <b>Note:</b> Some of the elements and attributes are not template driven. Refer to the xml for elements level details.				

## Publish Negotiation Results

This transaction publishes the negotiated terms to the order.

Use this transaction in environments where an order must go through a negotiation phase.

This transaction needs to be run after negotiation is completed.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 129. Publish Negotiation Results Attributes

Attribute	Value
Base Transaction ID	PUBLISH_ORD_NEGOTIATION

Table 129. Publish Negotiation Results Attributes (continued)

Attribute	Value
Base Document Type	Order
Base Process Type	Order Negotiation
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 130. Publish Negotiation Results Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 131. Publish Negotiation Results Statistics

Statistic Name	Description
NumNegotiationsProcessed	Number of negotiations processed.
NumNegotiationsPublished	Number of negotiations published.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 132. Events Raised by Publish Negotiation Results Transaction

Base Transaction	Raised when...	Key Data	Data Published	Template Support?
PUBLISH_ORD_NEGOTIATION/ON_SUCCESS	On success	Negotiation_dbd.txt	YCP_get Negotiation Details_output.xml	Yes *

Table 132. Events Raised by Publish Negotiation Results Transaction (continued)

Base Transaction	Raised when...	Key Data	Data Published	Template Support?
RECEIVE_ORD _NEGOTIATION/ ON_SUCCESS	On success, when DocumentType is 0001, EntityType is ORDER.	Number of concurrent time-triggered transactions running.	receiveOrder Negotiation_dbd. txt	No
* <b>Note:</b> Template used for this event is the same template used by the getNegotiationDetails() API to form the output XML.				

## Release

This transaction releases orders to specific ship nodes, making sure that the scheduled ship nodes have enough inventory to process the order.

This transaction should be invoked after the scheduling process.

For more details, see the information provided under the releaseOrder() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

If you run the combined 'Schedule and Release' agent, do not also run the individual Schedule or the individual Release agents.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 133. Release Attributes

Attribute	Value
Base Transaction ID	RELEASE
Base Document Type	Order
Base Process Type	Order Fulfillment
APIs Called	releaseOrder()

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 134. Release Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
IgnoreReleaseDate	Optional. Determines whether the schedule process should ignore line release date criteria. Valid values are: <ul style="list-style-type: none"> <li>Y - Releases line quantities regardless of release date criteria</li> <li>N - Default value. Releases line quantities only after release date criteria have been met.</li> </ul>

Table 134. Release Criteria Parameters (continued)

Parameter	Description
CheckInventory	Optional. Determine whether inventory should be checked. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Inventory needs to be checked.</li> <li>• N - Inventory does not need to be checked.</li> </ul>
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 135. Release Criteria Statistics

Statistic Name	Description
NumFutureDateFailures	Number of orders did not attempt to release because of future date failures.
NumOrdersAttempted	Number of orders attempted to release.
NumOrdersCannotBeProcessed Failures	Number of orders did not attempt to release because of cannot be processed failures.
NumOrdersProcessed	Number of orders processed.
NumOrdersReleased	Number of orders released.
NumOrdersBackordered	Number of orders backordered.
NumOrderLinesReleased	Number of order lines released.
NumOrderLinesBackordered	Number of order lines backordered.
NumReleasesCreated	Number of order releases created.
NumOrdersCannotBeProcessed Failures	Number of orders that were not released due to process failure.

If the release process results in splitting of an order line, NumOrderLinesReleased, NumOrderLinesBackordered, and NumOfReleasesCreated may result in more than one count.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table, if tasks on hold are not ready to be processed.

## Events Raised

This transaction raises events as specified under the releaseOrder() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.



## Route Shipment

This time-triggered transaction is used to route shipments and belongs to the Outbound Shipment pipeline. It assigns the Carrier and Carrier Service codes for the shipment based on the Routing Guide configured.

The Route Shipment transaction either includes shipments in an existing load or creates a new load and includes the shipments in it.

Shipments can be consolidated to a load, only if the following conditions are met:

- Expected Ship Date - The expected ship date of the shipments must be less than or equal to the must ship before date of the load.
- Expected Load Departure Date - The expected load departure date must be less than or equal to the must ship before date of the shipments in the load.

The must ship before date is a date computed for the load, based on all shipments present in the load. For example, if a load has three shipments with their must ship before dates as 12.22.2005, 12.12.2005, and 12.19.2005 respectively, then the must ship before date of the load is computed as 12.12.2005, as it is the earliest of the three dates.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table 136. Route Shipment*

Attribute	Value
Base Transaction ID	ROUTE_SHIPMENT.0001
Base Document Type	Order
Base Process Type	ORDER_DELIVERY
Abstract Transaction	No
APIs Called	None
User Exits Called	com.yantra.ydm.japi.ue.YDMOverrideDetermineRoutingUE com.yantra.ydm.japi.ue.YDMBeforeDetermineRoutingUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 137. Route Shipment Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where YFS_SHIPMENT table may exist in multiple schemas. Runs the agent for the colony.

Table 137. Route Shipment Criteria Parameters (continued)

Parameter	Description
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 138. Route Shipment Statistics

Statistic Name	Description
NumRouted	Number of shipments routed.

## Pending Job Count

For this transaction the pending job count is the number of records representing the unheld orders that are available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 139. Events Raised by the Route Shipment Transaction

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	shipment_dbd.txt	YDM_ROUTE_SHIPMENT.ON_SUCCESS.xml	Yes
ON_FAILURE	shipment_dbd.txt	YDM_ROUTE_SHIPMENT.ON_FAILURE.xml	Yes

However, note that the template name would read <TransactionId>.ON\_SUCCESS.xml.

## Schedule

This transaction schedules orders to specific ship nodes making sure that the scheduled ship nodes have enough inventory to process the order.

Run this transaction after order creation.

Do not run the individual Schedule or Release agents when running the combined "Schedule and Release" agent.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 140. Schedule Attributes

Attribute	Value
Base Transaction ID	SCHEDULE
Base Document Type	Order
Base Process Type	Order Fulfillment
APIs Called	scheduleOrder()

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 141. Schedule Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
MaximumRecords	Determines the maximum number of possible solutions that the Schedule Agent can find. This parameter may improve the best solution found, but it also impacts the performance of this agent.  If left blank or specified as 0 (zero), it defaults to 5.
OptimizationType	Optional. Determines the optimization rules to apply to the scheduling process. Valid values are: <ul style="list-style-type: none"> <li>• 01 - Optimize on date (Default)</li> <li>• 02 - Optimize on ship node priority</li> <li>• 03 - Optimize on number of shipments</li> </ul>
OrderFilter	Optional. Determines the types of orders to filter. Possible values are: <ul style="list-style-type: none"> <li>• A - All orders (Default)</li> <li>• B - Backorders only</li> <li>• N - New orders only</li> </ul>
ScheduleAndRelease	Optional. Notify the schedule process to release all releasable line quantities. Valid values are: <ul style="list-style-type: none"> <li>• Y - Releases successfully scheduled line quantities.</li> <li>• N - Default value. Only schedules line quantities.</li> </ul> Enabling this parameter does not validate hold types configured for the release transaction.
IgnoreReleaseDate	Optional. Determines whether the schedule process should ignore line release date criteria. Valid values are: <ul style="list-style-type: none"> <li>• Y - Releases line quantities regardless of release date criteria.</li> <li>• N - Releases lines quantities only after release date criteria have been met. Default.</li> </ul>

Table 141. Schedule Criteria Parameters (continued)

Parameter	Description
Next Task Queue Interval	Not used. This agent updates a failed task so that it is suspended for the back order retry interval setup in the appropriately scheduled rule.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 142. Schedule Statistics

Statistic Name	Description
NumFutureDateFailures	<p>Number of orders that Sterling Selling and Fulfillment Foundation did not attempt to schedule because of future date failures.</p> <p>Failures can be caused by any of the following:</p> <ul style="list-style-type: none"> <li>• If the OrderFilter is “B” (Backorders Only) and there are no backordered or unscheduled lines.</li> <li>• If the OrderFilter is “N” (New orders Only) and there are some backordered or unscheduled lines.</li> <li>• If order has order lines within only backordered or unscheduled status and the status modify timestamp is after the current time - the back order wait period specified in the scheduling rule.</li> </ul>
NumOrdersAttempted	Number of orders attempted to schedule. This statistic does not include the values for NumFutureDateFailures and NumOrdersCannotBeProcessedFailures statistics.
NumOrderLinesReleased	Number of order lines that have been released.
NumOrdersCannotBeProcessed Failures	<p>Number of orders that Sterling Selling and Fulfillment Foundation did not attempt to schedule because of cannot be processed failures.</p> <p>Failures can be caused by any of the following:</p> <ul style="list-style-type: none"> <li>• The result of the YFSCheckOrderBeforeProcessingUE user exit returns as false.</li> <li>• The Order has the HoldFlag attribute set to ‘Y’.</li> <li>• The Order has the SaleVoided attribute set to ‘Y’.</li> <li>• The Order does not have PaymentStatus as AUTHORIZED, INVOICED, PAID, nor NOT_APPLICABLE.</li> </ul>
NumOrdersCreated	Number of orders created. This also includes the number of procurement orders created.
NumOrderLinesCreated	Number of order lines created.
NumOrdersProcessed	Number of orders processed.

Table 142. Schedule Statistics (continued)

Statistic Name	Description
<b>NumOrdersScheduled</b>	Number of orders that have at least one line that was scheduled.  This includes scheduled lines in any status except BACKORDER.
<b>NumOrdersProcOrdersCreated</b>	Number of procurement orders created.
<b>NumWorkOrdersCreated</b>	Number of work orders created.
<b>NumOrdersBackordered</b>	Number of orders backordered.
<b>NumOrderLinesScheduled</b>	Number of order lines scheduled.
<b>NumOrderLinesBackordered</b>	Number of order lines backordered.
<b>NumReleasesCreated</b>	Number of order releases created.

## Pending Job Count

For this transaction the pending job count is the number of records representing the unheld orders that are available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table, if tasks on hold are not ready to be processed.

## Events Raised

This transaction raises events as specified under the scheduleOrder() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Providing Oracle Hints

You can provide Oracle Hints to increase the performance of the scheduleOrder agent. The two hints that can be provided for each criteria ID of the scheduleOrder agent are the Outer Hint and the Inner Hint. The Outer Hint is always used for the YFS\_TASK\_Q table. The Inner Hint is used for the YFS\_ORDER\_HEADER table only if the earlier hold functionality is used; otherwise, the Inner Hint is used for the YFS\_ORDER\_RELEASE\_STATUS table.

Insert the following entries in the yfs.properties file in order to enable Oracle Hints:

1. Edit the <INSTALL\_DIR>/properties/yfs.properties file.
2. Insert `yfs.<agent_criteria_id>.getjobs.hint.outer=/** parallel(YFS_TASK_Q 8) full(yfs_task_q) */`  
Insert `yfs.<agent_criteria_id>.getjobs.hint.inner=/** NL_SJ */`

## Send Invoice

This transaction publishes invoice data that can be directed to an external accounts receivable system.

In environments that require an interface with accounts receivable systems, this transaction needs to be scheduled. This transaction raises an event for an invoice based on the following configuration at the following times in the order lifecycle:

- Publish invoice at shipment creation - This implies that your accounts payable system takes care of payment collection. Invoices can be published as soon as they are created.
- Publish invoice after payment collection - This implies that the Console take care of the payment collection. When payment is in the AT\_COLLECT status and the payment is not from an external system, an invoice is published only if the entire payment amount is collected. If the payment is in the AT\_CREATE status or the payment is from an external system, the invoice is published unconditionally.

Many of this transaction's elements and attributes are template driven. Refer to the XML for element level details.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 143. Send Invoice Attributes*

Attribute	Value
Base Transaction ID	SEND_INVOICE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	getOrderInvoiceDetails()

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 144. Send Invoice Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 145. Send Invoice Statistics*

Statistic Name	Description
NumInvoicesSent	Number of invoices sent.

## Pending Job Count

For this transaction the pending job count is the number of order invoices in created ("00") status.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 146. Events Raised by the Send Invoice Transaction

Transaction/Event	Key Data	Data Published	Template Support?
PUBLISH_INVOICE_DETAIL	modifyOrder_dbd.txt and sendInvoice_dbd.txt	YFS_getOrderInvoiceDetails_output.xml	Yes

Additional events may be raised by the `getOrderInvoiceDetails()` API. For detailed information about the events, see the details provided under this API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Send Item Changes

In integrated environments, this transaction publishes item data changes that are directed to an external system.

When item changes occur in Sterling Selling and Fulfillment Foundation, they need to be communicated to the external system.

The business process may require the synchronization of items all at once in a batch. For example, at the end of each business day, the `sendItemChanges` agent can be configured to synchronize items based on the synchronization logic. This transaction retrieves all items that are not logical kit or dynamic physical kit items and whose `SyncTS` is null or `MaxModifyTS` is greater than the `SyncTS`.

The `MaxModifyTS` of an item is updated with the current timestamp whenever an item is modified. The transaction then retrieves detailed information about those items and raises the `ON_SUCCESS` event. This event should be configured to invoke the Send Item Changes action.

For more information about how this integration is implemented, see the *Sterling Selling and Fulfillment Foundation: Integration Guide*.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 147. Send Item Changes Attributes

Attribute	Value
Base Transaction ID	SEND_ITEM_CHANGES
Base Document Type	None
Base Process Type	General
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 148. Send Item Changes Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Organization Code	Optional. The organization from which items are synchronized. This field is blank by default.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

None.

## Pending Job Count

For this transaction the pending job count is the number of items requiring synchronization. This is determined for product items that are not logical kit or dynamic physical kit items and whose SyncTS is null or MaxModifyTS is greater than the SyncTS.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 149. Events Raised by the Send Item Changes Transaction

Transaction/Event	Key Data	Data Published	Template Support?
ON_SUCCESS	None	YCM_SEND_ITEM_CHANGES_ON_SUCCESS.XML	Yes

## Send Customer Changes

In integrated environments, this transaction publishes customer data changes that are directed to an external system.

When customer changes occur in Sterling Selling and Fulfillment Foundation, they need to be communicated to the external system.

The business process may require the synchronization of customers all at once in a batch. For example, at the end of each business day, the sendItemChanges agent can be configured to synchronize items based on the synchronization logic. This transaction retrieves all customers that are consumers, have a user ID present, and are required to synchronize. This transaction can also be used to complete the initial synchronization of users between the two systems. For example, if an



external system is already in place, and Sterling Selling and Fulfillment Foundation is then added, the SendCustomerChanges agent synchronizes the users from the external system.

The sendCustomerChanges agent also serves as a backup mechanism. If a customer synchronization event fails, the agent automatically retries the synchronization after a specified amount of time.

The MaxModifyTS of an customer is updated with the current timestamp whenever an customer is modified, whenever syncTS is less than MaxModifyTS, or when syncTS is null. The transaction then retrieves detailed information about those customers and raises the ON\_SUCCESS event. This event should be configured to invoke the Send Customer Changes action.

For more information about how this integration is implemented, see the *Sterling Selling and Fulfillment Foundation: Integration Guide*.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 150. Send Customer Changes Attributes*

Attribute	Value
Base Transaction ID	SEND_CUSTOMER_CHANGES
Base Document Type	None
Base Process Type	General
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 151. Send Customer Changes Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Organization Code	Optional. The organization from which customers are synchronized. This field is blank by default.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

None.

## Pending Job Count

For this transaction the pending job count is the number of customers requiring synchronization. This is determined for customers that are consumers, have a user

ID present, and are required to synchronize.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 152. Events Raised by the Send Customer Changes Transaction

Transaction/Event	Key Data	Data Published	Template Support?
SEND_CUSTOMER_CHANGES.ON_SUCCESS	None	YSC_SEND_CUSTOMER_CHANGES.ON_SUCCESS.XML	Yes

## Send Order

This transaction tries to raise the ON\_SUCCESS event for an order whose OrderHeaderKey is stored in the task queue object. The event is raised only if all of the order lines of the order reach particular status(es) completely. That is, the entire ORDERED\_QTY of each line must be in the particular status(es). In addition to raising the event, the line statuses are also changed to the drop statuses, corresponding to the pickup statuses. The SendOrder transaction, derived from the abstract transaction SEND\_ORDER, should have the event, pickup, and drop statuses configured. For more information, see the details provided under the sendOrder() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

If an order needs to be communicated to a third party, use this transaction.

The TransactionKey posted in the task object must be an instance of the Abstract Transaction SEND\_ORDER for the ProcessType associated with the Order. Otherwise, an exception is thrown.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 153. Send Order Attributes

Attribute	Value
Base Transaction ID	SEND_ORDER
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	Yes
APIs Called	sendOrder()

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 154. Send Order Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.

Table 154. Send Order Criteria Parameters (continued)

Parameter	Description
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

None.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

## Events Raised

This transaction raises events as specified under the sendOrder() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Send Release

The Send Release Agent dispatches releases to ship nodes.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 155. Send Release Attributes

Attribute	Value
Transaction Name	Send Release
Transaction ID	SHIP_ADVICE
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	com.yantra.yfs.agent.YFSWMSShipAdviceAgent

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 156. Send Release Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.

Table 156. Send Release Criteria Parameters (continued)

Parameter	Description
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 157. Send Release Statistics

Statistic Name	Description
NumReleasesProcessed	Number of order releases processed.
NumReleasesSent	Number of order releases sent.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to ( $\leq$ ) the current date value in the YFS\_Task\_Q table.

## Events Raised

The following events are raised by this time-triggered transaction:

Table 158. Events Raised by the Send Release Transaction

Transaction/Event	Data Published
PUBLISH_SHIP_ADVICE	YFS_publishShipAdvice_output.xml

## Start Order Negotiation

This transaction creates the negotiations for orders that are configured to go through the negotiation process.

Use this transaction in environments where an Order needs to go through a Negotiation phase before it is released.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 159. Start Order Negotiation Attributes

Attribute	Value
Base Transaction ID	START_ORD_NEGOTIATION
Base Document Type	Order
Base Process Type	Order Fulfillment

Table 159. Start Order Negotiation Attributes (continued)

Attribute	Value
Abstract Transaction	No
APIs Called	createNegotiation()
User Exits Called	YCPBeforeCreateNegotiationUE, YCPGetNegotiationNoUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 160. Start Order Negotiation Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
Node	Required. The ship node for which records are being processed.
ColonyID	Required in a multischema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 161. Start Order Negotiation Statistics

Statistic Name	Description
NumOrdersProcessed	Number of orders processed.
NumNegotiationsCreated	Number of negotiations created.

## Pending Job Count

For this transaction the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

## Events Raised

This transaction raises events as specified under the createNegotiation() API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

## Synchronize Colony Map

The Colony Map Synchronizer agent inserts or updates colony mappings of organizations and users in the PLT\_COLONY\_MAP table. When you run the agent

for the first time, it populates this table, which is a necessary step in upgrading to multischema mode after installing or upgrading Sterling Selling and Fulfillment Foundation.

For more information about upgrading to multischema mode, see the *Sterling Selling and Fulfillment Foundation: Multi-Tenant Enterprise Guide*.

## Attributes

The following are attributes for this time-triggered transaction:

Table 162. Colony Map Synchronizer Attributes

Attribute	Value
Base Transaction ID	COLONY_MAP_SYNC
Base Process Type	General
Abstract Transaction	No

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 163. Colony Map Synchronizer Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records to Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	The colony to be synchronized.  Initially, you must run the agent on the DEFAULT colony provided by the Sterling Selling and Fulfillment Foundation installation so that it populates the PLT_COLONY_MAP table. After this, you can run the agent on another ColonyID.
InsertDefaultMappings	If set to Y, users for which the colony cannot be determined will be mapped to the colony for which the Colony Map Synchronizer agent is run.

## Statistics Tracked

None.

## Pending Job Count

None.

## Events Raised

None.

## Tables Purged

None.

## Update Best Match Region

The Update Best Match Region transaction manages the YFS\_REGION\_BEST\_MATCH table, which is used by Data Warehouse Analytics to report best match region data. The best match region is defined by the following five address attributes in person info records:

- ADDRESS\_LINE6
- CITY
- STATE
- SHORT\_ZIP\_CODE
- COUNTRY

**Note:** COUNTRY refers to Country/Region.

The agent for the Update Best Match Region transaction runs in two modes that allow you to set up and update the YFS\_REGION\_BEST\_MATCH table.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table 164. Update Best Match Region Attributes*

Attribute	Value
Base Transaction ID	UPDATE_BEST_MATCH_REGION
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YSCGetShortZipCode UE

### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 165. Update Best Match Region Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If UpdateOnly = N, only distinct records are returned per agent call. If left blank, it defaults to 1000.

Table 165. Update Best Match Region Criteria Parameters (continued)

Parameter	Description
<b>TableType</b>	<p>Required in a multischema deployment when YFS_Person_Info table may exist in multiple schemas.</p> <p>Valid Values: CONFIGURATION, TRANSACTION, MASTER.</p> <p>If set to CONFIGURATION, the agent runs for the YFS_Person_Info records associated with tables that have TableType as CONFIGURATION; for example, YFS_Organization, YFS_Ship_Node, and so forth.</p> <p>If set to TRANSACTION, the agent runs for the YFS_Person_Info records associated with tables that have TableType as TRANSACTION; for example, YFS_Order_Header, YFS_Shipment, and so forth.</p> <p>Note that the agent would run for all TableTypes that exist in the same schema as the one passed. For example, if set to TRANSACTION, the agent would also run for YFS_Person_Info records associated with tables that have TableType as MASTER, since they reside in the same schema.</p>
<b>ColonyID</b>	<p>Required in a multi schema deployment where the YFS_PERSON_INFO table may exist in multiple schemas. Runs the agent for the colony.</p>
<b>UpdateOnly</b>	<p>Mode in which to run. Valid values are:</p> <ul style="list-style-type: none"> <li>• N - Default value. Adds records from the YFS_PERSON_INFO table to the YFS_REGION_BEST_MATCH table and populates the region key in the YFS_BEST_MATCH table. To perform the initial setup of Best Match Region for Analytics, set UpdateOnly to N.</li> <li>• Y - Update mode. Updates region keys based on addresses in YFS_REGION_BEST_MATCH. After performing the initial setup of Best Match Region for Analytics, set this value to Y to specify update mode.</li> </ul>
<b>LastPersonInfoKey</b>	<p>Optional. If UpdateOnly is set to N, LastPersonInfoKey determines the first person info record to populate. If no key is specified, the value defaults to Null.</p>
<b>LastRegionBest MatchKey</b>	<p>Optional. If UpdateOnly is set to Y, LastRegionBestMatchKey determines the first region best match key to update. If no key is specified, the value defaults to Null.</p>

### Statistics Tracked

None.

### Pending Job Count

None.

### Events Raised

None.



## Tables Purged

None.

## PopulateOwnershipTransferSummary

This method updates the YFS\_OWNERSHIP\_TRANSFER\_SUMMARY table.

This transaction updates the YFS\_OWNERSHIP\_TRANSFER\_SUMMARY table by checking the records in YFS\_INV\_OWN\_TRANSFER\_RCD table.

It also updates the IS\_STATISTICS\_UPDATED to 'Y' in YFS\_INV\_OWN\_TRANSFER\_RCD table after the record has been used by the transaction.

## Attributes

Following are the attributes for this time-triggered transaction:

*Table 166. YFSPopulateOwnershipTransfer Attributes*

Attribute	Value
Base Transaction ID	POPULATE_OWN_TRANS_SUMM
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

Following are the criteria parameters for this transaction:

*Table 167. YFSPopulateOwnershipTransfer Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, which is the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multi schema deployment where the YFS_OWNERSHIP_TRANSFER_SUMMARY and YFS_INV_OWN_TRANSFER_RCD tables may exist in multiple schemas. Runs the agent for the colony.

Statistics Tracked

None

Pending Job Count

None

Events Raised

None

---

## Time-Triggered Purge Transactions

There are several transactions that you can use to purge your database tables at specific time intervals.

Purge transactions determine when a table should be purged by determining the current date and subtracting the retention days specified by the purge. If the timestamp on the table is less than or equal to (current day - retention days) the table is purged.

In some cases, a purge may look at another field other than the table's timestamp. These are pointed out in the documentation.

When an entity is being purged, the related or dependent information that is present in other tables should be taken into consideration for purging along with it. For example, if a sales order with live shipments is being purged, any cross reference to that order is not accurate in the Order Shipment Console. As another example, agents such as Purge Order and Purge Order History also purge the records from extension tables, such as `yfs_order_header_extension` and `yfs_order_line_extension`, as well as their associated histories.

Some of the statistics collected and tracked in Release 9.1 for time-triggered transactions, monitors, and integration and application servers may change with the next release of Sterling Selling and Fulfillment Foundation.

All Time-Triggered Purge Transactions have a `CollectPendingJobs` criteria parameter. If this parameter is set to `N`, the agent does not collect information on the pending jobs for that time-triggered transaction. This pending job information is used for monitoring the monitor in the System Management Console.

By default, `CollectPendingJobs` is set to `Y`. It can be helpful to set it to `N` if one particular time-triggered transaction is performing a significant amount of `getPendingJobs` queries, and the overhead cost is too high.

### Purge Strategy

The following recommendations should be taken into consideration when planning a purge strategy for each purge transaction:

- Test purges by setting `Live` to 'N'.
- Turn on logging to test what is purged.
- Set up purge traces in the System Management Console and analyze the information.

### Configuring Purge Transaction Log Files

#### About this task

You can configure purges to write log files to a directory you specify. Each time you run a particular purge, new data is appended to this file. If no file exists, one is created.

To specify a purge log file directory:

## Procedure

1. Configure the `yfs.purge.path` property in the `<INSTALL_DIR>/properties/customer_overrides.properties` file. For example, on UNIX you might specify the log files to be written to the `/app/yfs/logs/purges` directory.  
For additional information about overriding properties using the `customer_overrides.properties` file, see the *Sterling Selling and Fulfillment Foundation: Properties Guide*.
2. Run the `<INSTALL_DIR>/bin/setupfiles.sh` script on UNIX, or the `<INSTALL_DIR>/bin/setupfiles.cmd` script on Windows.

## Available Purges

This section contains details of all purge transactions in alphabetical order.

### Access Token Purge

This purge removes access tokens from the system. If all of the following conditions are met, the `PLT_ACCESS_TOKEN` table is picked up for purge:

- The access token is expired or is in inactive state.
- The last modified date is earlier than or equal to the current date minus the purge criteria's retention days.

### Attributes

The following are the attributes for this time-triggered transaction:

Table 168. Access Token Purge Attributes

Attribute	Value
Base Transaction ID	ACCESSTOKPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

### Criteria Parameters

The following are the criteria parameters for this transaction:

Table 169. Access Token Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the .
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

Table 169. Access Token Purge Criteria Parameters (continued)

Parameter	Description
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.

### Statistics Tracked

The following statistics are tracked for this transaction:

Table 170. Access Token Purge Statistics

Statistic Name	Description
NumAccessTokenPurged	Number of access token records purged.

### Pending Job Count

For this transaction the pending job count is the number of records that can be purged from the PLT\_ACCESS\_TOKEN table.

### Events Raised

None.

### Tables Purged

PLT\_ACCESS\_TOKEN

### Capacity Purge

This purge removes capacity data from the system. This reduces the load on frequently accessed tables.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a capacity data gets picked up for purge:

- All resource pool standard capacity periods with effective to date earlier than or equal to the current date minus the purge criteria's retention days.
- All resource pool overridden capacity with the capacity date earlier than or equal to the current date minus the purge criteria's retention days.
- All resource pool capacity consumption with consumption date less than or equal to the current date minus the purge criteria's retention days.
- All resource pool capacity consumption details where appointment date is earlier than the system date minus the purge criteria's retention days (or ManualReservationPurgeLeadDays for manually created reservations).

- All resource pool capacity consumption details where expiration date has passed and reservation Id is not blank.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 171. Capacity Purge Attributes*

Attribute	Value
Base Transaction ID	CAPACITYPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 172. Capacity Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 173. Capacity Purge Statistics*

Statistic Name	Description
NumStdCapacityPeriodsPurged	Number of standard capacity periods purged.
NumCapacityOverridesPurged	Number of capacity overrides purged.
NumCapacityConsumptionsPurged	Number of capacity consumptions purged.

## Pending Job Count

For this transaction the pending job count is the total number of records that can be purged from the YFS\_RES\_POOL\_STD\_CAPCTY\_PERD, YFS\_RES\_POOL\_CAPCTY\_OVERRIDE, YFS\_RES\_POOL\_CONSMPTN\_DTLS and YFS\_RES\_POOL\_CAPCTY\_CONSMPTN tables.

## Events Raised

None.

## Tables Purged

The YFS\_RES\_POOL\_STD\_CAPCTY\_PERD table is purged when  $EFFECTIVE\_TO\_DATE \leq (CurrentDate - LeadDays)$

The YFS\_RES\_POOL\_CAPCTY\_OVERRIDE table is purged when  $CAPACITY\_DATE \leq (CurrentDate - LeadDays)$

The YFS\_RES\_POOL\_CAPCTY\_CONSMPTN table is purged when  $CONSUMPTION\_DATE \leq (CurrentDate - LeadDays)$ , or if a manual reservation is taken, when  $CONSUMPTION\_DATE \leq (CurrentDate - Manual\ Reservation\ Retention\ Days)$ . When this table is purged, YFS\_RES\_POOL\_CONSMPTN\_DTLS is also purged.

The YFS\_RES\_POOL\_CONSMPTN\_DTLS table is purged when  $RESERVATION\_EXPIRATION\_DATE \leq (CurrentDate - LeadDays)$

## Draft Order History Purge

This purge deletes data from history tables after a specified interval, which in turn, reduces the load on frequently accessed tables.

You can use purge codes' pseudo-logic to analyze the purges. If the following condition is met, a draft order is picked up for history purge:

- The last modified date of the draft order exceeds the retention day period.

All the enterprise using the Console must schedule purge transactions.

For more information about Additional Purge Criteria Based on Line Type, see the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

**Note:** The draft order must be purged and moved to the history tables before you purge the draft order history tables. See "Draft Order Purge" on page 161.

Sterling Selling and Fulfillment Foundation does not provide a transaction for draft order history purges. If you are defining a transaction that purges draft order history tables, refer to the following Criteria Parameters section for information about the transaction criteria.

If you do not want to define your own transaction to purge draft order history tables, you can use the Order Purge transaction and specify DRAFTORDERHISTPRG for the PurgeCode. To configure the Order Purge transaction for draft order history table purges, refer to "Order Purge" on page 189 for more information.

## Criteria Parameters

The following are the criteria parameters for defining a draft order history transaction:

Table 174. Draft Order History Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Required. Enterprise for which the Draft Order History Purge has to be run. If not passed, all the enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Removes qualifying records from the history tables that are listed in Tables Purged.</li><li>• N - Test mode. Determines the rows that are removed without actually removing them.</li></ul>
PurgeCode	Required. Set to DRAFTORDERHISTPRG. Used for internal calculations, such as determining retention days. Corresponds to the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

None.

## Events Raised

None.

## Tables Purged

YFS\_ANSWER\_SET\_TRAN\_H

YFS\_ANSWER\_TRAN\_H

YFS\_CHARGE\_TRAN\_DIST\_H

YFS\_CHARGE\_TRANSACTION\_H

YFS\_CREDIT\_CARD\_TRANSACTION\_H

YFS\_ENTITY\_ADDRESS\_H

YFS\_HEADER\_CHARGES\_H

YFS\_INSTRUCTION\_DETAIL\_H

YFS\_INVOICE\_COLLECTION\_H

YFS\_LINE\_CHARGES\_H

YFS\_NOTES\_H  
YFS\_ORDER\_AUDIT\_DETAIL\_H  
YFS\_ORDER\_AUDIT\_H  
YFS\_ORDER\_AUDIT\_LEVEL\_H  
YFS\_ORDER\_DATE\_H  
YFS\_ORDER\_HEADER\_H  
YFS\_ORDER\_HEADER\_H\_EXTENSION  
YFS\_ORDER\_HOLD\_TYPE\_H  
YFS\_ORDER\_HOLD\_TYPE\_LOG\_H  
YFS\_ORDER\_INVOICE\_DETAIL\_H  
YFS\_ORDER\_INVOICE\_H  
YFS\_ORDER\_KIT\_LINE\_H  
YFS\_ORDER\_KIT\_LINE\_SCHEDULE\_H  
YFS\_ORDER\_LINE\_H  
YFS\_ORDER\_LINE\_H\_EXTENSION  
YFS\_ORDER\_LINE\_OPTION\_H  
YFS\_ORDER\_LINE\_REQ\_TAG\_H  
YFS\_ORDER\_LINE\_SCHEDULE\_H  
YFS\_ORDER\_PROD\_SER\_ASSOC\_H  
YFS\_ORDER\_RELEASE\_H  
YFS\_ORDER\_RELEASE\_STATUS\_H  
YFS\_ORDER\_SER\_PROD\_ITEM\_H  
YFS\_PAYMENT\_H  
YFS\_PROMOTION\_AWARD\_H  
YFS\_PROMOTION\_H  
YFS\_RECEIVING\_DISCREPANCY\_DTL\_H  
YFS\_RECEIVING\_DISCREPANCY\_H  
YFS\_REFERENCE\_TABLE\_H



## Draft Order Purge

This purge archives data into history tables after a specified interval, which in turn, reduces the load on frequently accessed tables. For information about purging draft orders from history tables, see “Draft Order History Purge” on page 158.

Sterling Selling and Fulfillment Foundation does not provide a transaction for draft order purges. If you are defining a transaction that purges draft orders, refer to the following Criteria Parameters section for details about the transaction criteria.

If you do not want to define your own transaction to purge draft orders, you can use the Order Purge transaction and specify DRAFTORDERPRG for the PurgeCode. To configure the Order Purge transaction for draft order purges, refer to “Order Purge” on page 189 for more information.

All the enterprise using the Console must schedule purge transactions.

Draft orders are picked up by the agent for validation when the following conditions are met:

- Draft order flag is set to Y.
- Modifyts is set for the retention date.

After the draft orders are picked up, each draft order is validated for purging based on the following conditions:

- No eligible order release status records (records with a status larger than zero) exist for the order.
- All the open child orders (derived, chained, return, exchange, or refund fulfillment) for the order are already purged.

If a draft order meets the set of conditions for validation listed earlier, the agent continues to verify the draft orders against the following criteria:

- Contains the Draft Created (1000) status, and all the extended Draft Created statuses.
- Does not have an order release status record that does not meet the retention days.
- The order's last modification should be before the lead time (in days) setup.
- In the case when an exchange order is part of a return order, the exchange order should be purged from history tables before the return order is purged.
- In the case of an order line reservation, the draft order cannot be purged.
- If the Draft Order Payment Processing flag is set to N, the draft orders are purged.
- If the Draft Order Payment Processing flag is set to Y and a charge exists on a draft order, the draft order is not purged. However, authorizations are not considered when validating draft orders for purge.
- For order lines, except service order lines:
  - If the Seller inventory update is required, the Status Inventory Type has the Update Seller Supply option turned on, and the Seller Supply Type is Onhand, or blank. (The Seller Supply Type can also be a custom seller supply type, with the Onhand Supply check box enabled.)
  - If the Seller Demand Type is blank.

- If the Buyer inventory update is required, and the Buyer Supply Type is Onhand, or blank.

### Criteria Parameters

The following are the criteria parameters for defining a draft order purge transaction:

*Table 175. Draft Order Purge Criteria Parameters*

Parameter	Description
<b>Action</b>	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
<b>Number of Records To Buffer</b>	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
<b>Next Task Queue Interval</b>	Optional. Specifies (in hours) how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
<b>EnterpriseCode</b>	Required. Enterprise for which the Draft Order Purge has to be run. If not passed, all the enterprises are monitored.  When the EnterpriseCode is blank, the purge criteria configured for the DEFAULT enterprise is used, and not the purge criteria configured for the draft order's enterprise.
<b>Live</b>	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged, to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
<b>PurgeCode</b>	Required. Set to DRAFTORDERPRG. Used for internal calculations, such as determining retention days. Corresponds to the PurgeCode used in Business Rules Purge Criteria.
<b>ColonyID</b>	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

None.

### Events Raised

None.

### Tables Purged

YFS\_ACTIVITY\_DEMAND

YFS\_ANSWER\_SET\_TRAN

YFS\_ANSWER\_TRAN

YFS\_CHARGE\_TRANSACTION

YFS\_CHARGE\_TRAN\_DIST

YFS\_CREDIT\_CARD\_TRANSACTION  
YFS\_ENTITY\_ADDRESS  
YFS\_HEADER\_CHARGES  
YFS\_INSTRUCTION\_DETAIL  
YFS\_INVOICE\_COLLECTION  
YFS\_LINE\_CHARGES  
YFS\_MONITOR\_ALERT  
YFS\_NOTES  
YFS\_ORDER\_AUDIT  
YFS\_ORDER\_AUDIT\_DETAIL  
YFS\_ORDER\_AUDIT\_LEVEL  
YFS\_ORDER\_HEADER  
YFS\_ORDER\_HEADER\_EXTENSION  
YFS\_ORDER\_HOLD\_TYPE  
YFS\_ORDER\_HOLD\_TYPE\_LOG  
YFS\_ORDER\_INVOICE  
YFS\_ORDER\_INVOICE\_DETAIL  
YFS\_ORDER\_KIT\_LINE  
YFS\_ORDER\_KIT\_LINE\_SCHEDULE  
YFS\_ORDER\_LINE  
YFS\_ORDER\_LINE\_EXTENSION  
YFS\_ORDER\_LINE\_OPTION  
YFS\_ORDER\_LINE\_REQ\_TAG  
YFS\_ORDER\_LINE\_RESERVATION  
YFS\_ORDER\_LINE\_SCHEDULE  
YFS\_ORDER\_LINE\_SRC\_CNTRL  
YFS\_ORDER\_PROD\_SER\_ASSOC  
YFS\_ORDER\_RELEASE

YFS\_ORDER\_RELEASE\_STATUS  
 YFS\_ORDER\_SER\_PROD\_ITEM  
 YFS\_ORDER\_DATE  
 YFS\_PAYMENT  
 YFS\_PMNT\_TRANS\_ERROR  
 YFS\_PROMOTION  
 YFS\_PROMOTION\_AWARD  
 YFS\_RECEIVING\_DISCREPANCY  
 YFS\_RECEIVING\_DISCREPANCY\_DTL  
 YFS\_REFERENCE\_TABLE  
 YFS\_TAX\_BREAKUP

### Delivery Plan Purge

This purge deletes delivery plans after they have completed their typical life cycle. All the loads and shipments that are associated with the delivery plans should have been purged before running this purge agent.

It purges all the delivery plans that have been marked as 'Closed' for a period greater than the retention days specified in the criteria parameters and those that do not have any shipments or loads. The order should have been moved to history before the lead time (in days) setup.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a delivery plan is picked up for purge:

- The delivery plan should be in the "Closed" status.
- The delivery plan should not be associated with any load or shipment.
- All orders associated with the delivery plan should be purged.
- The last modification performed on the delivery plan should fall before the lead time (in days) setup.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table 176. Delivery Plan Purge Attributes*

Attribute	Value
Base Transaction ID	DELIVERYPLANPRG
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No

Table 176. Delivery Plan Purge Attributes (continued)

Attribute	Value
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

Table 177. Delivery Plan Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Delivery Plan Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
BatchDelete	Required. The method by which all records are deleted from the table. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Records are deleted in batches.</li> <li>• N - Records are deleted one by one.</li> </ul>
ColonyID	Required in a multi schema deployment where the YFS_DELIVERY_PLAN table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

Table 178. Delivery Plan Purge Statistics

Statistic Name	Description
NumDeliveryPlansPurged	Number of delivery plans purged.

### Pending Job Count

For this transaction the pending job count is the number of records that can be purged from the YFS\_DELIVERY\_PLAN table.

7 Events Raised

None.

## Tables Purged

YFS\_DELIVERY\_PLAN

## Export Table Purge

This purge removes export table data from the system. This reduces the load on frequently accessed tables.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, the YFS\_EXPORT table is picked up for purge:

- YFS\_EXPORT records should be marked as processed (Status = 10).
- The last modified time should fall before the lead time (in days) setup.  
This purge reads only the rules defined by the hub. Enterprise overridden rules are not considered. This purge should be single threaded when you run it in batch delete mode (BatchDelete=Y).

Any enterprise using the Application Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 179. Export Table Purge Attributes*

Attribute	Value
Base Transaction ID	EXPORTTBLPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 180. Export Table Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>

Table 180. Export Table Purge Criteria Parameters (continued)

Parameter	Description
<b>BatchDelete</b>	Required. The method by which all records are deleted from the table. Valid values are: <ul style="list-style-type: none"> <li>• Y - Records are deleted in batches.</li> <li>• N - Default value. Records are deleted one by one.</li> </ul>
<b>PurgeCode</b>	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
<b>CollectPendingJobs</b>	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
<b>ColonyID</b>	Required in a multi schema deployment where the YFS_EXPORT table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 181. Export Table Purge Statistics

Statistic Name	Description
NumExportsPurged	Number of exports purged.

## Pending Job Count

For this transaction the pending job count is the number of records that can be purged from the YFS\_Export table.

## Events Raised

None.

## Tables Purged

YFS\_EXPORT

## Import Table Purge

This purge removes import table data from the system. This reduces the load on frequently accessed tables.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, the YFS\_IMPORT table is picked up for purge:

- YFS\_IMPORT records should be marked as processed (Status = "10").
- The "last modified time" should fall before the lead time (in days) setup. This purge reads only the rules defined by the hub. Enterprise overridden rules are not considered. This purge should be single threaded when you run it in batch delete mode(BatchDelete=Y).

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 182. Import Table Purge Attributes*

Attribute	Value
Base Transaction ID	IMPORTTBLPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 183. Import Table Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
BatchDelete	Required. The method by which all records are deleted from the table. Valid values are: <ul style="list-style-type: none"> <li>• Y - Records are deleted in batches.</li> <li>• N - Default value. Records are deleted one by one.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where the YFS_IMPORT table may exist in multiple schemas. Runs the agent for the colony.



## Statistics Tracked

The following statistics are tracked for this transaction:

Table 184. Import Table Purge Statistics

Statistic Name	Description
NumImportsPurged	Number of import tables purged.

## Pending Job Count

For this transaction the pending job count is the number of records that can be purged from the YFS\_Import table.

## Events Raised

None.

## Tables Purged

YFS\_IMPORT

## Inventory Audit Purge

This purge removes inventory audit data from the system. This reduces the load on frequently accessed tables.

Any enterprise using the Console must schedule purge transactions.

All inventory audits of the provided organization with modify timestamp earlier than the current date minus the purge criteria's retention days can be configured to be picked up by the Inventory Audit Purge.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, an inventory audit record is picked up for purge:

- The inventory audit record's last modification is earlier than the current timestamp minus the retention days.  
Number of threads for this purge's agent criteria details must be set to 1. For more information about agent criteria, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

The following are the attributes for this time-triggered transaction:

Table 185. Inventory Audit Purge Attributes

Attribute	Value
Base Transaction ID	INVENTORYAUDITPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 186. Inventory Audit Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. The inventory organization for which the Inventory Audit Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Table Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 187. Inventory Audit Statistics

Statistic Name	Description
NumInventoryAuditsPurged	Number of inventory audits purged.

## Pending Job Count

For this transaction the pending job count is the number of records that can be purged from the YFS\_Inventory\_Audit table.

## Events Raised

None.

## Table Purged

YFS\_INVENTORY\_AUDIT

## Inventory Purge

This purge removes inventory data from the system. This reduces the load on frequently accessed tables. This purge does not take retention days into account when purging.

You can use purge codes pseudo-logic to analyze purges.

For YFS\_INVENTORY\_SUPPLY, if the following conditions are met, an inventory supply is picked up for purge:

- Supply record has the same availability type as the node. For example, TRACK or INFINITE.
- Supply record has 0 quantity.
- Supply record does not contain the supply type “INFO”.

For YFS\_INVENTORY\_DEMAND, if the following conditions are met, an inventory demand is picked up for purge:

- Demand record has 0 quantity or lesser.
- Demand record does not have demand details as well as matching demand record in YFS\_INVENTORY\_DEMAND\_ADDNL tables.

For YFS\_INVENTORY\_TAG, it is purged if the INVENTORY\_TAG\_KEY is not used by any of the existing supply and demand.

For YFS\_INVENTORY\_RESERVATION, an inventory reservation is picked up for purge if it meets the following conditions:

- Inventory reservation record has 0 quantity or ship date is earlier than the system date minus the purge criteria's retention days.

For YFS\_INVENTORY\_NODE\_CONTROL, it is purged if the INV\_PIC\_INCORRECT\_TILL\_DATE is earlier than the current time stamp minus the purge criteria's retention days.

For YFS\_IBA\_TRIGGER, it is purged if IBA\_REQUIRED = 'N', IBA\_RUN\_REQUIRED = 'N', and LAST\_IBA\_PROCESSED\_TS is earlier than the current time stamp minus the purge criteria's retention days.

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 188. Inventory Purge Attributes*

Attribute	Value
Base Transaction ID	INVENTORYPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 189. Inventory Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	The inventory organization for which the Inventory Purge needs to be run.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 190. Inventory Purge Statistics

Statistic Name	Description
NumInventoryDemandsPurged	Number of inventory demands purged.
NumInventoryNodeControlsPurged	Number of inventory node controls purged.
NumInventoryReservationsPurged	Number of inventory reservations purged.
NumInventoryTagsPurged	Number of inventory tags purged.
NumItemBasedAllocationTriggers Purged	Number of item based allocation triggers purged.

## Pending Job Count

For this transaction, the pending job count is the total number of records that can be purged from the YFS\_Inventory\_Supply, YFS\_Inventory\_Demand, YFS\_Inventory\_Tag, YFS\_Inventory\_Reservation, YFS\_IBA\_Trigger, and YFS\_Inventory\_Node\_Control tables.

## Events Raised

None.

## Tables Purged

YFS\_IBA\_TRIGGER

YFS\_INVENTORY\_DEMAND

YFS\_INVENTORY\_TAG

YFS\_INVENTORY\_RESERVATION

YFS\_INVENTORY\_SUPPLY

YFS\_INVENTORY\_NODE\_CONTROL

## Inventory Supply Temp Purge

The Inventory Supply Temp purge agent cleans up the contents in the temporary inventory tables generated by the process of synchronizing the Sterling Selling and Fulfillment Foundation inventory picture with the actual inventory picture at the nodes.

The node inventory picture is stored during the loading process into the YFS\_INVENTORY\_SUPPLY\_TEMP table. Once the synchronization phase is complete and the YFS\_INVENTORY\_SUPPLY table has been updated, the YFS\_INVENTORY\_SUPPLY\_TEMP table needs to be purged, which is done through this agent.

For more information about configuring the synchronization with node inventory, see the *Sterling Selling and Fulfillment Foundation: Global Inventory Visibility Configuration Guide*.

The Inventory Supply Temp purge agent is used to purge all records in the YFS\_INVENTORY\_SUPPLY\_TEMP table whose modify timestamp is less than current time minus the purge criteria's retention days for a group of YantraMessageGroupID.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 191. Inventory Supply Temp Purge Attributes

Attribute	Value
Base Transaction ID	SUPPLYTEMPPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 192. Inventory Supply Temp Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
EnterpriseCode	Optional. The inventory organization for which the Inventory Supply Temp Purge needs to be run. If not passed, then all enterprises are monitored.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_INVENTORY_SUPPLY_TEMP table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 193. Inventory Supply Temp Purge Statistics

Statistic Name	Description
NumInventorySupplyTempsPurged	Number of entries in the YFS_INVENTORY_SUPPLY_TEMP table purged.

## Pending Job Count

Number of unique YantraMessageGroupIDs from YFS\_INVENTORY\_SUPPLY\_TEMP table whose maximum modify timestamp is less than current timestamp minus purge criteria's lead day.

## Events Raised

None.

## Tables Purged

YFS\_INVENTORY\_SUPPLY\_TEMP

## Item Audit Purge

This purge removes the YFS\_AUDIT table data from the system, which reduces the load on frequently accessed tables. It purges records in the YFS\_AUDIT and the YFS\_AUDIT\_HEADER tables that meet the following conditions:

- YFS\_AUDIT records that have 'modifyts' greater than the retention days specified and the records have the table name as 'YFS\_ITEM'.
- The last modified time is before the lead time (in days) setup.

When the enterprise modifies records in the YFS\_ITEM table through the Applications Manager, the YFS\_ITEM is audited and the audit records are inserted in the YFS\_AUDIT table. In order to clean up the audit records, this purge transaction can be used.

Any enterprise using the Console must schedule purge transactions accordingly.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table 194. Item Audit Purge Attributes*

Attribute	Value
Base Transaction ID	YFS_ITEM_AUDIT_PURGE
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 195. Item Audit Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, the value defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), this value defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Production mode. Deletes records from the regular tables.</li><li>• N - Test mode.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_AUDIT and YFS_AUDIT_HEADER tables may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 196. Item Audit Purge Statistics*

Statistic Name	Description
NumItemAuditRecords Purged	Number of item audit records purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_AUDIT table that match the criteria values.

## Events Raised

None.

## Tables Purged

YFS\_AUDIT, YFS\_AUDIT\_HEADER

## Load History Purge

This purge deletes the load data from history tables after it completes its typical lifecycle. This reduces the load on frequently accessed tables.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the last modification made to the load is before the lead time (in days) is met, a load is picked up for purge.

Before you run this transaction, ensure to purge loads and move them to history tables. For more information about purging loads, see “Load Purge” on page 178.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 197. Load History Purge Attributes*

Attribute	Value
Base Transaction ID	LOADHISTPRG
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE



## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 198. Load History Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Load Purge needs to be run. If not passed, all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
Purge Code	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 199. Load History Purge Statistics

Statistic Name	Description
NumLoadHistoriesPurged	Number of load histories purged.
NumLoadShipment HistoriesPurged	Number of load shipment histories purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Load\_H table.

## Events Raised

None.

## Tables Purged

YFS\_LOAD\_H

YFS\_LOAD\_STOP\_H

YFS\_LOAD\_SHIPMENT\_CHARGE\_H

YFS\_LOAD\_STATUS\_AUDIT\_H

YFS\_SHIPMENT\_CONTAINER\_H

YFS\_CONTAINER\_ACTIVITY\_H

YFS\_LOADED\_CONTAINER\_H

YFS\_LOAD\_SHIPMENT\_H

YFS\_ADDITIONAL\_DATE\_H

YFS\_LOAD\_HOLD\_TYPE\_H

YFS\_LOAD\_HOLD\_TYPE\_LOG\_H

## Load Purge

This purge removes load data from the system. It picks up all loads that have been marked as 'Closed' and purges them. Empty Loads (for example, loads with no shipments) are not considered for purge. As a part of this purge, the associated child tables are also purged.

This is not a pipeline transaction. It also does not work from the task queue.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, a load is picked up for purge:

- The Load's last modification should fall before the lead time (in days) setup.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 200. Load Purge Attributes*

Attribute	Value
Base Transaction ID	LOADPRG
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 201. Load Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

Table 201. Load Purge Criteria Parameters (continued)

Parameter	Description
EnterpriseCode	Optional. Enterprise for which the Load Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

Table 202. Load Purge Statistics

Statistic Name	Description
NumLoadShipmentsPurged	Number of load shipments purged.
NumLoadsPurged	Number of loads purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Load table.

### Events Raised

None.

### Tables Purged

YFS\_ADDITIONAL\_DATE

YFS\_LOAD

YFS\_LOAD\_HOLD\_TYPE

YFS\_LOAD\_HOLD\_TYPE\_LOG

YFS\_LOAD\_STOP

YFS\_LOAD\_SHIPMENT

YFS\_LOAD\_SHIPMENT\_CHARGES (charges that pertain to this load)

YFS\_LOAD\_STATUS\_AUDIT

YFS\_LOADED\_CONTAINER

YFS\_SHIPMENT\_CONTAINER

YFS\_CONTAINER\_ACTIVITY

## Negotiation History Purge

This purge deletes negotiation history data from the system. This reduces the load on frequently accessed tables. It purges data from the order negotiation history tables.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, a negotiation is picked up for history purge:

- The last modified date of the negotiation exceeds the retention day period.

Any enterprise using the Console must schedule purge transactions.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table 203. Negotiation History Purge Attributes*

Attribute	Value
Base Transaction ID	NEGOTIATIONHISTPRG
Base Document Type	Order
Base Process Type	Order Negotiation
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 204. Negotiation History Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Negotiation History Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>

Table 204. Negotiation History Purge Criteria Parameters (continued)

Parameter	Description
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 205. Negotiation History Purge Statistics

Statistic Name	Description
NumNegotiationHistoriesPurged	Number of negotiation histories purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Negotiation\_Hdr\_H table.

## Events Raised

None.

## Tables Purged

YFS\_AUDIT

YFS\_NEGOTIATION\_HDR\_H

YFS\_NEGOTIATION\_LINE\_H

YFS\_RESPONSE\_H

YFS\_RESPONSE\_HDR\_H

YFS\_RESPONSE\_LINE\_H

YFS\_RESPONSE\_LINE\_DTL\_H

## Negotiation Purge

This purge archives data into history tables after it completes its typical lifecycle. This reduces the load on frequently accessed tables. It works from the task queue (YFS\_TASK\_Q) table.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, a negotiation is picked up for purge:

- The last modification performed on the negotiation falls before the lead time (in days) setup.
- The negotiation is in pickable status.

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 206. Negotiation Purge Attributes*

Attribute	Value
Base Transaction ID	ORD_NEGOTIATION_PURGE
Base Document Type	Order
Base Process Type	Order Negotiation
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 207. Negotiation Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Negotiation Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 208. Negotiation Purge Statistics*

Statistic Name	Description
NumOrderNegotiationsPurged	Number of order negotiations purged.

## Pending Job Count

For this transaction, the pending job count is the number of records available to be processed by the transaction with the `AVAILABLE_DATE` value less than or equal to (`<=`) the current date value in the `YFS_Task_Q` table.

## Events Raised

None

## Tables Purged

YFS\_AUDIT

YFS\_NEGOTIATION\_HDR

YFS\_NEGOTIATION\_LINE

YFS\_RESPONSE

YFS\_RESPONSE\_HDR

YFS\_RESPONSE\_LINE

YFS\_RESPONSE\_LINE\_DTL

## Opportunity History Purge

This transaction deletes tasks previously archived by the Opportunity Purge. See “Opportunity Purge” on page 185.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, an opportunity that is previously purged by the opportunity purge agent is picked up for history purge:

- The last modified date of the opportunity should exceed the retention day period.
- The quote history is purged.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 209. Opportunity History Purge Attributes*

Attribute	Value
Base Transaction ID	OPPORTUNITYHISTPRG
Base Document Type	Opportunity
Base Process Type	Opportunity Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 210. Opportunity History Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Live	Optional. Mode in which to run. Defaults to N. <ul style="list-style-type: none"><li>• Y - Default value. Removes qualifying records from the history tables listed under Tables Purged.</li><li>• N- Test mode. Determines the rows that are removed without actually removing them.</li></ul>
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Opportunity History Purge needs to be run. If not passed, then all enterprises are monitored.  When the EnterpriseCode is blank, the purge criteria configured for the DEFAULT enterprise is used; not the purge criteria configured for the opportunity's enterprise.
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 211. Opportunity History Purge Statistics

Statistic Name	Description
NumOpportunityHistory Purged	Number of opportunity histories purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_OPPORTUNITY\_H table.

## Events Raised

None.



## Tables Purged

YFS OPPORTUNITY\_H

YFS OPPORTUNITY\_H\_EXTENSION

## Opportunity Purge

This time-triggered transaction purges all the opportunities for a period greater than the retention days specified in the Opportunity Purge criteria, and those which are either in the status of cancelled or completed.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, an opportunity is picked up for purge:

- The last modified date of the opportunity should exceed the retention day period.
- The quote associated with the opportunity should be purged.
- The opportunity should be in pickable status by the purge transaction.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 212. Opportunity Purge Attributes*

Attribute	Value
Base Transaction ID	OPPORTUNITYPRG
Base Document Type	Opportunity
Base Process Type	Opportunity Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 213. Opportunity Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Live	Optional. Mode in which to run. Defaults to Y. <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

Table 213. Opportunity Purge Criteria Parameters (continued)

Parameter	Description
EnterpriseCode	Optional. Enterprise for which the Opportunity Purge needs to be run. If not passed, then all enterprises are monitored.  When the EnterpriseCode is blank, the purge criteria configured for the DEFAULT enterprise is used; not the purge criteria configured for the opportunity's enterprise.
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 214. Opportunity Purge Statistics

Statistic Name	Description
NumOpportunityPurged	Number of opportunities purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_OPPORTUNITY table.

## Events Raised

None.

## Tables Purged

YFS\_OPPORTUNITY

YFS\_OPPORTUNITY\_EXTENSION

## Order History Purge

This purge deletes data from history tables after it completes its typical lifecycle. This reduces the load on frequently accessed tables.

The order should have been purged and moved into the history tables before you can run this transaction. For more information about this, see "Order Purge" on page 189.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, an order is picked up for history purge:

- The last modified date of the order exceeds the retention day period.

Any enterprise using the Console must schedule purge transactions.

For more information about Additional Purge Criteria Based on Line Type, see the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 215. Order History Purge Attributes

Attribute	Value
Base Transaction ID	ORDERHISTPRG
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 216. Order History Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Order History Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Removes qualifying records from the history tables listed under Tables Purged.</li> <li>N- Test mode. Determines the rows that are removed without actually removing them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 217. Order History Purge Statistics

Statistic Name	Description
NumOrderHistoriesPurged	Number of order histories purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Order\_HEADER\_H table.

## Events Raised

None.

## Tables Purged

YFS\_ANSWER\_SET\_TRAN\_H  
YFS\_ANSWER\_TRAN\_H  
YFS\_CHARGE\_TRAN\_DIST\_H  
YFS\_CHARGE\_TRAN\_REQUEST\_H  
YFS\_CHARGE\_TRAN\_RQ\_MAP\_H  
YFS\_CHARGE\_TRANSACTION\_H  
YFS\_CREDIT\_CARD\_TRANSACTION\_H  
YFS\_ENTITY\_ADDRESS\_H  
YFS\_HEADER\_CHARGES\_H  
YFS\_INSTRUCTION\_DETAIL\_H  
YFS\_INVOICE\_COLLECTION\_H  
YFS\_LINE\_CHARGES\_H  
YFS\_NOTES\_H  
YFS\_ORDER\_AUDIT\_DETAIL\_H  
YFS\_ORDER\_AUDIT\_H  
YFS\_ORDER\_AUDIT\_LEVEL\_H  
YFS\_ORDER\_DATE\_H  
YFS\_ORDER\_HEADER\_H  
YFS\_ORDER\_HEADER\_H\_EXTENSION  
YFS\_ORDER\_HOLD\_TYPE\_H  
YFS\_ORDER\_HOLD\_TYPE\_LOG\_H  
YFS\_ORDER\_INVOICE\_DETAIL\_H  
YFS\_ORDER\_INVOICE\_H

YFS\_ORDER\_KIT\_LINE\_H  
YFS\_ORDER\_KIT\_LINE\_SCHEDULE\_H  
YFS\_ORDER\_LINE\_H  
YFS\_ORDER\_LINE\_H\_EXTENSION  
YFS\_ORDER\_LINE\_OPTION\_H  
YFS\_ORDER\_LINE\_REQ\_TAG\_H  
YFS\_ORDER\_LINE\_SCHEDULE\_H  
YFS\_ORDER\_PROD\_SER\_ASSOC\_H  
YFS\_ORDER\_RELEASE\_H  
YFS\_ORDER\_RELEASE\_STATUS\_H  
YFS\_ORDER\_SER\_PROD\_ITEM\_H  
YFS\_PAYMENT\_H  
YFS\_PROMOTION\_AWARD\_H  
YFS\_PROMOTION\_H  
YFS\_RECEIVING\_DISCREPANCY\_DTL\_H  
YFS\_RECEIVING\_DISCREPANCY\_H  
YFS\_REFERENCE\_TABLE\_H  
YFS\_TAX\_BREAKUP\_H  
YIC\_BOM\_HEADER\_H  
YIC\_BOM\_LINE\_H  
YIC\_BOM\_MESSAGE\_H  
YIC\_BOM\_PROP\_H

### **Order Purge**

This purge archives data into history tables after it completes its typical lifecycle. To purge orders from history tables, see “Order History Purge” on page 186. This reduces the load on frequently accessed tables. It works on a task queue. It picks up the orders from YFS\_TASK\_Q table that are available for the transaction PURGE.

If purge criteria are not met, AVAILABLE\_DATE is calculated based on the modify time stamp of the order in YFS\_ORDER\_HEADER table as well as the YFS\_TASK\_Q table, whichever is maximum. To this value, retention days is added to the new AVAILABLE\_DATE.

This transaction depends on all lines of an order being in a status pickable by the Purge transaction.

The following statuses are available for configuration to be picked up by Order Purge:

- Draft Created (1000) and all extended Draft Created Statuses.
- Created (1100) and all extended Created statuses. These statuses are available only for document types Sales Order, Purchase Order and Transfer Order.
- Released (3200) and all extended Released statuses.
- Shipped (3700) and all extended Shipped statuses.
- Completed (3700) and all extended Completed statuses. These statuses are available only for the document type Master Order.
- Received (3900) and all extended Received statuses.
- Cancelled (9000) and all extended Cancelled statuses.
- Shorted (9020) and all extended Shorted statuses.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, an order is picked up for purge:

- All open child orders (derived, chained, return, exchange, procurement, or refund fulfillment) for the order must already be purged.
- No pending transfer-out charges to another order exceeding the transfer-ins.
- No pending adjustment invoices.

An order is purged immediately if it meets the above three criteria and is completely cancelled with payment collection complete.

For the purge agent to pick up a cancelled order, the payment status of the order must be one of the following:

- Paid
- Not Applicable

If an order does not meet any of the above criteria, continue checking for the criteria given below:

- No order release status record that does not meet the retention days.
- It should be in the correct status for purge. For example,
  - All service requests for the order should have Shipped or extended Shipped status.
  - The payment status for the order should be Paid or Not Applicable.
  - It must not have any unpurged negotiations.
- For all order lines other than service request lines:
  - If the Seller inventory update is required, the Status Inventory Type has the "Update Seller Supply" option turned on, and the Seller Supply Type is "Onhand", or blank. (The Seller Supply Type can also be a custom seller supply type with the "Onhand Supply" checkbox enabled.)
  - If the Seller Demand Type is blank.
  - If the Buyer inventory update is required and the Buyer Supply Type is "Onhand", or blank.
- The order's last modification should fall before the lead time (in days) setup.
- Any enterprise using the Console must schedule purge transactions.
- The order must not have a undelivered service line.

- In the case of an exchange order for processing a return order, the exchange order should be purged from history before the return order can be purged.

With no change to status inventory type, a in Shipped (3700) status or its extended status is purged if the Buyer is not passed.

An order in Shipped status or extended Shipped status in the default pipeline is not purged if the Buyer passed on the is tracking inventory. This prevents the purging of the order relating to the pending supply for the Buyer tracking inventory.

To purge such orders, the status inventory type for the Shipped or extended Shipped status should be configured such that the Buyer Supply Type is ONHAND for the status inventory type.

When the purge agent is run, the draft order without lines are purged to the order history table. Once the purge history agent is run, the draft orders without lines gets deleted permanently.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table 218. Order Purge Attributes*

Attribute	Value
Base Transaction ID	PURGE
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 219. Order Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Next Task Queue Interval	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
EnterpriseCode	Optional. Enterprise for which the Order Purge needs to be run. If not passed, then all enterprises are monitored.  When the EnterpriseCode is blank, the purge criteria configured for the DEFAULT enterprise is used; not the purge criteria configured for the order's enterprise.

Table 219. Order Purge Criteria Parameters (continued)

Parameter	Description
<b>Live</b>	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
<b>PurgeCode</b>	Required. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria. You can set this parameter to the following values: <ul style="list-style-type: none"> <li>• DRAFTORDERHISTPRG to purge draft order information from the order history tables.</li> <li>• DRAFTORDERNOLINEHISTPRG to purge draft orders without order lines from the order history tables.</li> <li>• DRAFTORDERNOLINEPRG to purge draft orders that have no order lines.</li> <li>• DRAFTORDERPRG to purge draft order information and archive it in the order history tables.</li> </ul> PurgeCode cannot be set to the value ORDER_RELEASE_STATUS_PURGE.
<b>AdditionalPurgeCode</b>	Optional. To purge order release status records, set this parameter to ORDER_RELEASE_STATUS_PURGE.  For more information, see "Order Release Status Purge" on page 194.
<b>ColonyID</b>	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

Table 220. Order Purge Statistics

Statistic Name	Description
<b>NumOrdersProcessed</b>	Number of order processed.
<b>NumOrdersPurged</b>	Number of orders purged.

### Pending Job Count

For this transaction, the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

### Events Raised

None.



## Tables Purged

YFS\_ACTIVITY\_DEMAND  
YFS\_ANSWER\_SET\_TRAN  
YFS\_ANSWER\_TRAN  
YFS\_CHARGE\_TRANSACTION  
YFS\_CHARGE\_TRAN\_DIST  
YFS\_CHARGE\_TRAN\_REQUEST  
YFS\_CHARGE\_TRAN\_RQ\_MAP  
YFS\_CREDIT\_CARD\_TRANSACTION  
YFS\_ENTITY\_ADDRESS  
YFS\_HEADER\_CHARGES  
YFS\_INSTRUCTION\_DETAIL  
YFS\_INVOICE\_COLLECTION  
YFS\_LINE\_CHARGES  
YFS\_MONITOR\_ALERT  
YFS\_NOTES  
YFS\_ORDER\_AUDIT  
YFS\_ORDER\_AUDIT\_DETAIL  
YFS\_ORDER\_AUDIT\_LEVEL  
YFS\_ORDER\_HEADER  
YFS\_ORDER\_HEADER\_EXTENSION  
YFS\_ORDER\_HOLD\_TYPE  
YFS\_ORDER\_HOLD\_TYPE\_LOG  
YFS\_ORDER\_INVOICE  
YFS\_ORDER\_INVOICE\_DETAIL  
YFS\_ORDER\_KIT\_LINE  
YFS\_ORDER\_KIT\_LINE\_SCHEDULE  
YFS\_ORDER\_LINE

YFS\_ORDER\_LINE\_EXTENSION  
YFS\_ORDER\_LINE\_OPTION  
YFS\_ORDER\_LINE\_REQ\_TAG  
YFS\_ORDER\_LINE\_RESERVATION  
YFS\_ORDER\_LINE\_SCHEDULE  
YFS\_ORDER\_LINE\_SRC\_CNTRL  
YFS\_ORDER\_PROD\_SER\_ASSOC  
YFS\_ORDER\_RELEASE  
YFS\_ORDER\_RELEASE\_STATUS  
YFS\_ORDER\_SER\_PROD\_ITEM  
YFS\_ORDER\_DATE  
YFS\_PAYMENT  
YFS\_PMNT\_TRANS\_ERROR  
YFS\_PROMOTION  
YFS\_PROMOTION\_AWARD  
YFS\_RECEIVING\_DISCREPANCY  
YFS\_RECEIVING\_DISCREPANCY\_DTL  
YFS\_REFERENCE\_TABLE  
YFS\_TAX\_BREAKUP  
YIC\_BOM\_HEADER  
YIC\_BOM\_LINE  
YIC\_BOM\_MESSAGE  
YIC\_BOM\_PROP

### **Order Release Status Purge**

The Order Release Status Purge agent extends the Order Purge agent's capabilities by purging order release status records before the Order Purge agent completely purges data to history tables.

If an order meets the criteria for purging, the order release status records with quantities of 0 are deleted from the YFS\_ORDER\_RELEASE\_STATUS table and are not put into the history table.

When the Order Release Status Purge agent has completed, the task queue's AVAILABLE\_DATE is reset to the date specified by the purge criteria for Order Purge. This enables the Order Purge agent to pick up and process an order as necessary. Order Purge will continue to purge order release status records as usual.

If the following conditions are met, the Order Purge agent purges order release status records:

- All conditions for Order Purge have been met. See “Order Purge” on page 189 for information about conditions for Order Purge.
- Order release records have 0 quantity.
- AdditionalPurgeCode in the Order Purge criteria is set to ORDER\_RELEASE\_STATUS\_PURGE.
- The order has been modified within the Order Purge lead days AdditionalPurgeCode.

### Criteria Parameters

The following are the criteria parameters for Order Release Status Purge:

*Table 221. Order Release Status Purge Criteria Parameters*

Parameter	Description
<b>Action</b>	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
<b>Number of Records To Buffer</b>	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
<b>Next Task Queue Interval</b>	Optional. Specifies in hours how long a failed task should be suspended before it is considered for reprocessing. Defaults to 5 hours.
<b>EnterpriseCode</b>	Optional. Enterprise for which the Order Purge needs to be run. If not passed, then all enterprises are monitored.  When the EnterpriseCode is blank, the purge criteria configured for the DEFAULT enterprise is used; not the purge criteria configured for the order's enterprise.
<b>Live</b>	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
<b>PurgeCode</b>	Required. To extend the Order Purge agent to purge order release status records, set to ORDERPRG. Used for internal calculations, such as determining retention days. You must also set AdditionalPurgeCode.
<b>AdditionalPurgeCode</b>	Required. To purge order release status records, set this parameter to ORDER_RELEASE_STATUS_PURGE.
<b>ColonyID</b>	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

None.

## Pending Job Count

The pending job count is the number of records available to be processed by Order Purge with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

## Events Raised

None.

## Tables Purged

YFS\_ORDER\_RELEASE\_STATUS

## Order Status Audit Purge

This purge removes order status audit data from the system. This reduces the load on frequently accessed tables.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, an order status audit is picked up for history purge:

- The last modified time falls before the lead time (in days) setup.

Any enterprise using the Console must schedule purge transactions.

This transaction needs to be run after negotiation is completed.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 222. Order Status Audit Purge Attributes*

Attribute	Value
Base Transaction ID	STATUSAUDITPRG
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 223. Order Status Audit Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

Table 223. Order Status Audit Purge Criteria Parameters (continued)

Parameter	Description
<b>EnterpriseCode</b>	Optional. Enterprise for which the Order Status Audit Purge needs to be run. If not passed, then all enterprises are monitored.
<b>Live</b>	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
<b>PurgeCode</b>	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
<b>ColonyID</b>	Required in a multi schema deployment where the YFS_STATUS_AUDIT Table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

Table 224. Order Status Audit Purge Statistics

Statistic Name	Description
<b>NumStatusAuditsPurged</b>	Number of status audits purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Status\_Audit table.

### Events Raised

None.

### Tables Purged

YFS\_STATUS\_AUDIT

### Organization Audit Purge

This purge removes the YFS\_AUDIT table data from the system, which reduces the load on frequently accessed tables. It purges records in the YFS\_AUDIT and the YFS\_AUDIT\_HEADER tables that meet the following conditions:

- The YFS\_AUDIT records that have 'modifyts' greater than the retention days specified and the records have the table name as 'YFS\_ORGANIZATION'.
- The last modified time is before the lead time (in days) setup.

When the enterprise modifies records in the YFS\_ORGANIZATION table through the Applications Manager, the YFS\_ORGANIZATION is audited and the audit records are inserted in the YFS\_AUDIT table. In order to clean up the audit records, this purge transaction can be used.

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 225. Organization Audit Purge Attributes*

Attribute	Value
Base Transaction ID	YFS_ORGANIZATION_AUDIT_PURGE
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 226. Organization Audit Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, the value defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), this value defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Production mode. Deletes records from the regular tables.</li> <li>• N - Test mode.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds to the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_AUDIT and YFS_AUDIT_HEADER tables may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 227. Organization Audit Purge Statistics*

Statistic Name	Description
NumOrganizationAudit RecordsPurged	Number of organization audit records purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_AUDIT table that match the criteria values.

## Events Raised

None.

## Tables Purged

YFS\_AUDIT

YFS\_AUDIT\_HEADER

## Person Info Purge

This purge gets a list of dates with the person info record count and sorts them by date in ascending order. Then, based on the specified number of records to buffer and the modify timestamp, it purges the applicable records and places them in the YFS\_PERSON\_INFO\_H table.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 228. PersonInfo Purge Attributes*

Attribute	Value
Base Transaction ID	PERSONINFOPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 229. PersonInfo Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. <ul style="list-style-type: none"><li>• If left blank or the number specified is less than 10000, it defaults to 10000.</li><li>• If the number specified is greater than 10000, then that value is used.</li></ul>
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>

Table 229. PersonInfo Purge Criteria Parameters (continued)

Parameter	Description
<b>PurgeCode</b>	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
<b>CollectPendingJobs</b>	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
<b>EnterpriseCode</b>	Optional. Enterprise for which the PersonInfo Purge needs to be run. If not passed, then all enterprises are monitored.
<b>TableType</b>	<p>Required in a multi schema deployment when YFS_Person_Info table may exist in multiple schemas.</p> <p>Valid Values: CONFIGURATION, TRANSACTION, MASTER.</p> <p>If set to CONFIGURATION, purge runs for the YFS_Person_Info records associated with tables that have TableType as CONFIGURATION; for example, YFS_Organization, YFS_Ship_Node, and so forth.</p> <p>If set to TRANSACTION, purge runs for the YFS_Person_Info records associated with tables that have TableType as TRANSACTION; for example, YFS_Order_Header, YFS_Shipment, and so forth.</p> <p>Note that purge would run for all TableTypes that exist in the same schema as the one passed. For example, if set to TRANSACTION, purge would also run for YFS_Person_Info records associated with tables that have TableType as MASTER, since they reside in the same schema.</p>
<b>ColonyID</b>	Required in a multi schema deployment where the YFS_PERSON_INFO table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

If it is left blank or any number less than 10,000 is specified, then it defaults to 10,000. But if any number > 10,000 is specified, then that value would be used.

Table 230. PersonInfo Purge Statistics

Statistic Name	Description
<b>NumPersonInfoPurged</b>	Number of person info records purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_PERSON\_INFO table.

## Events Raised

None.



## Tables Purged

YFS\_PERSON\_INFO

### Person Info History Purge

This purge deletes records from the YFS\_PERSON\_INFO\_H table based on the purge criteria.

#### Attributes

The following are the attributes for this time-triggered transaction:

*Table 231. PersonInfo History Purge Attributes*

Attribute	Value
Base Transaction ID	PERSONINFOHISTPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

#### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 232. PersonInfo History Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. <ul style="list-style-type: none"><li>• If left blank or the number specified is less than 10000, it defaults to 10000.</li><li>• If the number specified is greater than 10000, then that value is used.</li></ul>
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
EnterpriseCode	Optional. Enterprise for which the PersonInfo Purge needs to be run. If not passed, then all enterprises are monitored.

Table 232. PersonInfo History Purge Criteria Parameters (continued)

Parameter	Description
<b>TableType</b>	<p>Required in a multi schema deployment when YFS_Person_Info table may exist in multiple schemas.</p> <p>Valid Values: CONFIGURATION, TRANSACTION, MASTER.</p> <p>If set to CONFIGURATION, purge runs for the YFS_Person_Info records associated with tables that have TableType as CONFIGURATION; for example, YFS_Organization, YFS_Ship_Node, and so forth.</p> <p>If set to TRANSACTION, purge runs for the YFS_Person_Info records associated with tables that have TableType as TRANSACTION; for example, YFS_Order_Header, YFS_Shipment, and so forth.</p> <p>Note that purge would run for all TableTypes that exist in the same schema as the one passed. For example, if set to TRANSACTION, purge would also run for YFS_Person_Info records associated with tables that have TableType as MASTER, since they reside in the same schema.</p>
<b>ColonyID</b>	<p>Required in a multi schema deployment where the YFS_PERSON_INFO_H table may exist in multiple schemas. Runs the agent for the colony.</p>

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 233. PersonInfo History Purge Statistics

Statistic Name	Description
<b>NumPersonInfoHistoryRecords Purged</b>	Number of person info history records purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_PERSON\_INFO\_H table.

## Events Raised

None.

## Tables Purged

YFS\_PERSON\_INFO\_H

## Picklist Purge

This purge picks up all picklists that have been existing for a period greater than the retention days specified in the criteria parameters and those that do not have any shipments.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a picklist is picked up for purge:

- The picklist should exist for more than the specified retention period.
- The picklist should not be associated with any shipment.

All shipments associated with the picklists should have been purged before running this purge agent.

#### Attributes

The following are the attributes for this time-triggered transaction:

*Table 234. Picklist Purge Attributes*

Attribute	Value
Base Transaction ID	PICKLISTPRG
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

#### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 235. Picklist Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_PICK_LIST table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 236. Picklist Purge Statistics*

Statistic Name	Description
NumPickListsPurged	Number of picklists purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_PICK\_LIST table.

## Events Raised

None.

## Tables Purged

YFS\_PICK\_LIST

## Price List Purge

This purge removes price list data from the system. This reduces the load on frequently accessed tables.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, a price list is picked up for purge:

- The price list has valid date less than the current date minus the purge criteria's retention days.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 237. Price List Purge Attributes*

Attribute	Value
Base Transaction ID	PRICELISTPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 238. Price List Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 239. Price List Purge Statistics

Statistic Name	Description
NumPriceSetsPurged	Number of price sets purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Price\_Set table.

## Events Raised

None.

## Tables Purged

YFS\_PRICE\_SET table with VALID\_TILL\_DATE less than or equal to (CurrentDate - LeadDays)

YFS\_PRICE\_PROGRAM\_DEFN

YFS\_ITEM\_PRICE\_SET

YFS\_ITEM\_PRICE\_SET\_DTL

## Purge Catalog Mass Audits

This purge removes old audit records from the YFS\_CATALOG\_MASS\_AUDIT table. This table contains data about changes to the catalog due to assignment of attributes and attribute values to categories and items. It also contains information about inherited attributes and attribute values. The purge transaction finds mass audit records that have not been modified in a specified number of days and removes those records from the database.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table 240. Purge Catalog Mass Audits Attributes*

Attribute	Value
Base Transaction ID	CATALOG_MASS_AUDIT_PURGE
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 241. Purge Catalog Mass Audits Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_CATALOG_MASS_AUDIT table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 242. Purge Catalog Mass Audits Statistics

Statistic Name	Description
NumCatalogMassAuditsPurged	Number of mass audit records purged.

## Pending Job Count

For this transaction the pending job count is the total number of records that can be purged from the YFS\_CATALOG\_MASS\_AUDIT table.

## Events Raised

None.

## Tables Purged

The YFS\_CATALOG\_MASS\_AUDIT table is purged when  $\text{MODIFYTS} < (\text{CurrentDate} - \text{LeadDays})$

## Receipt History Purge

This transaction deletes receipts previously archived by the Receipt Purge. See “Receipt Purge” on page 209.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a receipt that is previously purged by the receipt purge agent is picked up for history purge:

- The last modified date of the receipt should exceed the retention day period.
- The shipment associated with the receipt should be purged from the history table.

To purge a receipt history, ensure that the Receipts are closed and Shipments are purged.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 243. Receipt History Purge Attributes

Attribute	Value
Base Transaction ID	RECEIPHISTPRG
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 244. Receipt History Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Receipt History Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Removes qualifying records from the history tables listed under Tables Purged.</li><li>• N- Test mode. Determines the rows that are removed without actually removing them.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 245. Receipt History Purge Statistics

Statistic Name	Description
NumReceiptLineHistoriesPurged	Number of receipt line histories purged.
NumReceiptHistoriesPurged	Number of receipt histories purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Receipt\_Header\_H table.

## Events Raised

None.

## Tables Purged

YFS\_RECEIPT\_HEADER\_H

YFS\_RECEIPT\_LINE\_H

YFS\_RECEIPT\_STATUS\_AUDIT\_H

YFS\_INSTRUCTION\_DETAIL\_H



## Receipt Purge

This purge removes receipt data from the system. This reduces the load on frequently accessed tables. This transaction picks up all receipts that are not open and not pending inspection and archives them into their history tables. See "Receipt History Purge" on page 207. It also archives and purges the receipt's child tables.

This is a pipeline transaction and works from a task queue.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a receipt is picked up for purge:

- The last modified date of the receipt should exceed the retention day period.
- The shipment associated with the receipt should be purged.
- The receipt should be in pickable status for the purge transaction.
- The value of the OpenReceiptFlag field should be set to "N".
- The receipt should not have pending inspections.
- There is no inventory in the warehouse for the receipt.

To purge a receipt, ensure that the receipts are closed and Shipments are purged.

### Attributes

The following are the attributes for this time-triggered transaction:

Table 246. Receipt Purge Attributes

Attribute	Value
Base Transaction ID	RECEIPTPRG
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

Table 247. Receipt Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Receipt Purge needs to be run. If not passed, then all enterprises are monitored.

Table 247. Receipt Purge Criteria Parameters (continued)

Parameter	Description
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

Table 248. Receipt Purge Statistics

Statistic Name	Description
NumReceiptLinesPurged	Number of Receipt Lines purged.
NumReceiptsPurged	Number of receipts purged.

### Pending Job Count

For this transaction, the pending job count is the number of records available to be processed by the transaction with the AVAILABLE\_DATE value less than or equal to (<=) the current date value in the YFS\_Task\_Q table.

### Events Raised

None.

### Tables Purged

YFS\_RECEIPT\_HEADER

YFS\_RECEIPT\_LINE

YFS\_RECEIPT\_STATUS\_AUDIT

YFS\_INSTRUCTION\_DETAIL

### Reprocess Error Purge

This purge deletes reprocess errors from the system. This reduces the load on frequently accessed tables.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a YFS\_REPROCESS\_ERROR table is picked up for purge:

- YFS\_REPROCESS\_ERROR records with State = Fixed or Ignored is processed.
- The last modified time is earlier than the lead time (in days) setup.

This purge reads only the rules defined by the hub. Enterprise overridden rules are not considered.

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 249. Reprocess Error Purge Attributes*

Attribute	Value
Base Transaction ID	REPROCESSPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 250. Reprocess Error Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_REPROCESS_ERROR table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 251. Reprocess Error Purge Statistics*

Statistic Name	Description
NumReprocessErrorsPurged	Number of reprocess errors purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_REPROCESS\_ERROR table.

## Events Raised

None.

## Tables Purged

YFS\_REPROCESS\_ERROR

## Reservation Purge

This purge deletes expired inventory reservations from the system. This reduces the load on frequently accessed tables as well as free up demands that are consumed by expired reservations.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, all records in the YFS\_INVENTORY\_RESERVATION tables are picked up for purge:

- EXPIRATION\_DATE is earlier than the current date or quantity is less than or equal to 0

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 252. Reservation Purge Attributes*

Attribute	Value
Base Transaction ID	RESERVATIONPRG
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 253. Reservation Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.

Table 253. Reservation Purge Criteria Parameters (continued)

Parameter	Description
<b>Live</b>	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
<b>PurgeCode</b>	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
<b>ColonyID</b>	Required in a multi schema deployment where the YFS_INVENTORY_RESERVATION table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 254. Reservation Purge Statistics

Statistic Name	Description
NumReservationsPurged	Number of reservations purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_INVENTORY\_RESERVATION table.

## Events Raised

None.

## Tables Purged

YFS\_INVENTORY\_RESERVATION

## Shipment History Purge

This transaction deletes shipments previously archived by the Shipment Purge. See “Shipment Purge” on page 215.

Any enterprise using the Console must schedule purge transactions.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, all records archived in the history table are picked up for purge:

- The last modification performed on the shipment falls before the lead time (in days) setup.

Orders related to the shipments should have been purged by order purge. Shipments should have been closed by the Close Shipment transaction. See “Close Shipment” on page 102.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 255. Shipment History Purge Attributes

Attribute	Value
Base Transaction ID	SHIPMENTHISTPRG
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 256. Shipment History Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Shipment History Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Removes qualifying records from the history tables listed under Tables Purged.</li><li>• N- Test mode. Determines the rows that are removed without actually removing them.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 257. Shipment History Purge Statistics

Statistic Name	Description
NumShipmentHistoriesPurged	Number of shipment histories purged.
NumShipmentLineHistoriesPurged	Number of shipment line histories purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Shipment\_H table.

## Events Raised

None.

## Tables Purged

YFS\_ADDITIONAL\_ATTRIBUTE\_H

YFS\_ADDITIONAL\_DATE\_H

YFS\_AUDIT

YFS\_CONTAINER\_DETAILS\_H

YFS\_CONTAINER\_STS\_AUDIT\_H

YFS\_INSTRUCTION\_DETAIL\_H

YFS\_SHIPMENT\_CONTAINER\_H

YFS\_SHIPMENT\_H

YFS\_SHIPMENT\_LINE\_H

YFS\_SHIPMENT\_LINE\_REQ\_TAG\_H

YFS\_SHIPMENT\_STATUS\_AUDIT\_H

YFS\_SHIPMENT\_TAG\_SERIAL\_H

YFS\_CONTAINER\_ACTIVITY\_H

## Shipment Purge

This purge removes shipment data from the system. This reduces the load on frequently accessed tables. This transaction picks up all shipments that have been marked as 'Closed' and archives them into their history tables. See "Shipment History Purge" on page 213. It also archives and purges the shipment's child tables.

This is not a pipeline transaction. It also does not work from the task queue.

Any enterprise using the Console must schedule purge transactions.

Orders related to the shipments should have been purged by order purge. Shipments should have been closed by the Close Shipment transaction. For more information, see "Close Shipment" on page 102.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a shipment is picked up for purge:

- The last modification performed on the shipment should fall before the lead time (in days) setup.
- The value of the ShipmentClosedFlag field should be set to "Y".
- The order record should already be purged for all shipment lines.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 258. Shipment Purge Attributes

Attribute	Value
Base Transaction ID	SHIPMENTPRG
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 259. Shipment Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Number of Days To Execute	Optional. Maximum number of days before the lead days the agent will look for shipment records to purge.
EnterpriseCode	Optional. Enterprise for which the Shipment Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 260. Shipment Purge Statistics

Statistic Name	Description
NumShipmentsPurged	Number of Shipments purged.
NumShipmentLinesPurged	Number of Shipment Lines purged.



## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_Shipment table.

## Events Raised

None.

## Tables Purged

YFS\_ADDITIONAL\_ATTRIBUTES

YFS\_ADDITIONAL\_DATE

YFS\_AUDIT

YFS\_CONTAINER\_DETAILS

YFS\_LOAD\_SHIPMENT\_CHARGE

YFS\_MONITOR\_ALERT

YFS\_SHIPMENT\_CONTAINER

YFS\_SHIPMENT\_STATUS\_AUDIT

YFS\_SHIPMENT

YFS\_INSTRUCTION\_DETAIL

YFS\_SHIPMENT\_MONITOR\_ALERT

YFS\_HEADER\_CHARGES

YFS\_LINE\_CHARGES

YFS\_TAX\_BREAKUP

YFS\_SHIPMENT\_HOLD\_TYPE

YFS\_SHIPMENT\_HOLD\_TYPE\_LOG

YFS\_SHIPMENT\_TAG\_SERIALS

YFS\_SHIPMENT\_LINE

YFS\_SHIPMENT\_LINE\_REQ\_TAG

YFS\_ACTIVITY\_DEMAND

YFS\_CONTAINER\_STS\_AUDIT

YFS\_CONTAINER\_ACTIVITY

## Shipment Statistics Purge

This transaction deletes the shipment statistics from the table older than the specified retention days.

This agent should be used whenever shipment statistics records need to be removed, such as after application server restart.

You can use purge codes pseudo-logic to analyze purges. If the following condition is met, the shipment statistics are picked up for purge:

- The last modification performed on the shipment statistics should fall before the lead time (in days) setup.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table 261. Shipment Statistics Purge Attributes*

Attribute	Value
Base Transaction ID	PRG_SHIP_STATS
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

### Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 262. Shipment Statistics Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Shipment Statistics Purge needs to be run. If not passed, then all enterprises are monitored.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the YFS_SHIPMENT_STATISTICS table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Parameters

The following are the statistics parameters for this transaction:

*Table 263. Shipment Statistics Purge Statistics*

Parameter	Description
<b>NumShipmentStatisticsPurged</b>	Number of shipment statistics purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_SHIPMENT\_STATISTICS table.

## Events Raised

None.

## Tables Purged

YFS\_SHIPMENT\_STATISTICS

## User Activity Purge

This purge deletes the user activity data from the system. It purges all records older than the specified retention days, and those records which have a logged out status. This purge must be single threaded when you run it in batch delete mode (BatchDelete=Y).

The following limitation is assumed when purging records:

This purge do not purge any record if the Application server goes down abruptly because the audit records of users connected to the application server at the time when the server went down cannot be updated. As a result, the last activity time or the logout time is not populated. The purge does not know whether the user has logged out or still logged in. Therefore, you need to manually delete these records.

The following are the attributes for this time-triggered transaction:

*Table 264. User Activity Purge Attributes*

Attribute	Value
<b>Base Transaction ID</b>	USERACTIVITYPRG
<b>Base Document Type</b>	None
<b>Base Process Type</b>	None
<b>APIs Called</b>	None
<b>User Exits Called</b>	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 265. User Activity Purge Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
Number of Records To Buffer	Required. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 100.
BatchDelete	Required. The method by which all records are deleted from the table. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Records are deleted in batches.</li><li>• N - Records are deleted one by one.</li></ul>
ColonyID	Required in a multi schema deployment where the YFS_USER_ACTIVITY table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 266. Statistics Purge Statistics

Statistic Name	Description
NumStatisticsPurged	Number of statistics purged

## Pending Job Count

None.

## Events Raised

None.

## Tables Purged

YFS\_USER\_ACTIVITY

## User Activity Audit Purge

This purge removes user activity audit data from the system. It purges all records older than the specified retention days. It purges only those records which have a logged out status (records with a Login\_Type of 'T' or 'N'). This purge should be single threaded when you run it in batch delete mode(BatchDelete=Y).

The following limitation is assumed when purging records:

- This purge does not purge any records if the Application server goes down abruptly because the audit records of users connected to application servers at the time the server went down cannot be updated. As a result, the last activity time or the logout time does not get populated and the purge does not know whether the user was logged out or was still logged in. These records have to be deleted manually.

The following are the attributes for this time-triggered transaction:

Table 267. User Activity Audit Purge Attributes

Attribute	Value
Base Transaction ID	USERACTAUDPURGE
Base Document Type	None
Base Process Type	None
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 268. User Activity Audit Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
CollectPendingJobs	If this parameter is set to "N", the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
Number of Records To Buffer	Required. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 100.

Table 268. User Activity Audit Purge Criteria Parameters (continued)

Parameter	Description
BatchDelete	Required. The method by which all records are deleted from the table. Valid values are: <ul style="list-style-type: none"> <li>• Y - Default value. Records are deleted in batches.</li> <li>• N - Records are deleted one by one.</li> </ul>
ColonyID	Required in a multi schema deployment where the YFS_USER_ACT_AUDIT table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 269. Statistics Purge Statistics

Statistic Name	Description
NumStatisticsPurged	Number of statistics purged

## Pending Job Count

None.

## Events Raised

None.

## Tables Purged

YFS\_USR\_ACT\_AUDIT

## Work Order History Purge

This transaction deletes tasks previously archived by the Work Order Purge. See “Work Order Purge” on page 224.

You can use purge codes pseudo-logic to analyze purges. If the last modified date of the work order exceeds the retention day period, a work order that is previously purged by the work order purge agent is picked up for history purge.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 270. Work Order History Purge Attributes

Attribute	Value
Base Transaction ID	WORK_ORDER_HISTORY_PURGE
Base Document Type	Work Order
Base Process Type	VAS
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 271. Work Order History Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Live	Optional. Mode in which to run. Defaults to N. <ul style="list-style-type: none"><li>• Y - Default value. Removes qualifying records from the history tables listed under Tables Purged.</li><li>• N- Test mode. Determines the rows that are removed without actually removing them.</li></ul>
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Node	Optional. Node for which the Work Order History Purge needs to be run. If not passed, then all nodes are monitored.
AgentCriteriaGroup	Optional. Used to classify nodes. This value can be accepted by Sterling Selling and Fulfillment Foundation time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 272. Work Order History Purge Statistics

Statistic Name	Description
NumWorkOrderHistoriesPurged	Number of work order histories purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_WORK\_ORDER\_H table.

## Events Raised

None.

## Tables Purged

YFS\_AUDIT

YFS\_WO\_APPT\_USER\_H

YFS\_WORK\_ORDER\_H

YFS\_WORK\_ORDER\_APPT\_H

YFS\_WORK\_ORDER\_ACTIVITY\_H

YFS\_WORK\_ORDER\_ACTY\_DTL\_H

YFS\_WORK\_ORDER\_AUDT\_DTL\_H

YFS\_WORK\_ORDER\_COMPONENT\_H

YFS\_WORK\_ORDER\_COMP\_TAG\_H

YFS\_WORK\_ORDER\_HOLD\_TYPE\_H

YFS\_WORK\_ORDER\_HOLD\_TYPE\_LOG\_H

YFS\_WORK\_ORDER\_PROD\_DEL\_H

YFS\_WORK\_ORDER\_SERVICE\_LINE\_H

YFS\_WORK\_ORDER\_STS\_AUDIT\_H

YFS\_WORK\_ORDER\_TAG\_H

## Work Order Purge

This time-triggered transaction purges all the work orders for a period greater than the retention days specified in the Work Order Purge criteria and those, which are either in the status of cancelled or completed.

You can use purge codes pseudo-logic to analyze purges. If the following conditions are met, a work order is picked up for purge:

- The last modified date of the work order should exceed the retention day period.
- The order associated with the work order should be purged.
- The work order should be in pickable status by the purge transaction.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 273. Work Order Purge Attributes*

Attribute	Value
Base Transaction ID	WORK_ORDER_PURGE
Base Document Type	Work Order
Base Process Type	VAS



Table 273. Work Order Purge Attributes (continued)

Attribute	Value
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 274. Work Order Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Live	Optional. Mode in which to run. Defaults to Y. <ul style="list-style-type: none"> <li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li> <li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li> </ul>
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Node	Optional. Node for which the Work Order Purge needs to be run. If not passed, then all nodes are monitored.
AgentCriteriaGroup	Optional. Used to classify nodes. This value can be accepted by Sterling Selling and Fulfillment Foundation time-triggered transactions that only perform their tasks on the nodes with a matching node transactional velocity value.  Valid values are: LOW, HIGH, and any additional values defined by the Hub from Application Platform > System Administration > Agent Criteria Groups.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 275. Work Order Purge Statistics

Statistic Name	Description
NumWorkOrdersPurged	Number of work orders purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_WORK\_ORDER table.

## Events Raised

None.

## Tables Purged

YFS\_AUDIT

YFS\_WO\_APPT\_USER

YFS\_WORK\_ORDER

YFS\_WORK\_ORDER\_ACTIVITY

YFS\_WORK\_ORDER\_ACTY\_DTL

YFS\_WORK\_ORDER\_HOLD\_TYPE

YFS\_WORK\_ORDER\_HOLD\_TYPE\_LOG

YFS\_WORK\_ORDER\_APPT

YFS\_WORK\_ORDER\_AUDT\_DTL

YFS\_WORK\_ORDER\_COMPONENT

YFS\_WORK\_ORDER\_COMP\_TAG

YFS\_WORK\_ORDER\_PROD\_DEL

YFS\_WORK\_ORDER\_SERVICE\_LINE

YFS\_WORK\_ORDER\_STS\_AUDIT

YFS\_WORK\_ORDER\_TAG

## YFS Audit Purge

This purge removes the YFS\_AUDIT table data from the system, which reduces the load on frequently accessed tables. It purges records in the YFS\_AUDIT and the YFS\_AUDIT\_HEADER tables that meet the following conditions:

- YFS\_AUDIT records that have 'modifyts' greater than the retention days specified and the value of table name matches in the YFS\_AUDIT table.
- The last modified time is before the lead time (in days) setup.

The way you configure the YFS Audit Purge may have some effect on the functioning of the Configuration Data Versioning Tool. For more information about configuration of the Data Versioning Tool, see the *Sterling Selling and Fulfillment Foundation: Configuration Deployment Tool Guide*.

When the enterprise extends the entities and sets the extended entities attribute AuditTable="Y", the extended tables are audited and the audit records are inserted in the YFS\_AUDIT table. In order to clean up the audit records, this purge transaction can be used.

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 276. YFS Audit Purge Attributes

Attribute	Value
Base Transaction ID	YFS_AUDIT_PURGE
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	YFSBeforePurgeUE

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 277. YFS Audit Purge Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, this value defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), this value defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Production mode. Deletes records from the regular tables.</li><li>• N - Test mode.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
Table Name	Required. The table name for which the audit records need to be purged.
TableType	Required in a multischema deployment when YFS_AUDIT table may exist in multiple schemas.  Valid Values: CONFIGURATION, TRANSACTION, MASTER.  If set to CONFIGURATION, the agent runs for the YFS_AUDIT records associated with tables that have TableType as CONFIGURATION; for example, YFS_Organization, YFS_Ship_Node, and so forth.  If set to TRANSACTION, the agent runs for the YFS_AUDIT records associated with tables that have TableType as TRANSACTION; for example, YFS_Order_Header, YFS_Shipment, and so forth.  Note that the agent would run for all TableTypes that exist in the same schema as the one passed. For example, if set to TRANSACTION, the agent would also run for YFS_AUDIT records associated with tables that have TableType as MASTER, since they reside in the same schema.

Table 277. YFS Audit Purge Criteria Parameters (continued)

Parameter	Description
ColonyID	Required in a multi schema deployment where the YFS_AUDIT and YFS_AUDIT_HEADER tables may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

Table 278. YFS Audit Purge Statistics

Statistic Name	Description
NumAuditRecordsPurged	Number of audit records purged.

### Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the YFS\_AUDIT table that match the criteria values.

### Events Raised

None.

### Tables Purged

YFS\_AUDIT, YFS\_AUDIT\_HEADER

### YFSInventoryOwnershipAudit Purge

This transaction purges all the records from YFS\_INV\_OWN\_TRANSFER\_RCD prior to the lead days specified in criteria parameters.

### Attributes

Following are the attributes for this time-triggered transaction:

Table 279. YFSInventoryOwnership Purge Attributes

Attribute	Value
Base Transaction ID	PURGE_INV_TRANSFR_RECORD
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

Following are the criteria parameters for this transaction:

*Table 280. YFSInventoryOwnership Purge Criteria Parameters*

Parameter	Description
<b>Action</b>	Required. Triggers the transaction. If left blank, this value defaults to Get, which is the only valid value.
<b>Number of Records To Buffer</b>	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), this value defaults to 5000.
<b>EnterpriseCode</b>	Optional. The inventory organization for which the YFSInventoryOwnership Audit Purge needs to run. If not passed, all the enterprises are monitored.
<b>Live</b>	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Production mode. Deletes records from the regular tables.</li><li>• N - Test mode.</li></ul>
<b>PurgeCode</b>	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds to the PurgeCode used in the Business Rules Purge Criteria.
<b>Lead Days</b>	Number of days before the present date, the agent will purge the records.
<b>ColonyID</b>	Required in a multi schema deployment where the YFS_INV_OWN_TRANSFER_RCD table may exist in multiple schemas. Runs the agent for the colony.

Statistics Tracked

None.

Pending Job Count

None.

Tables Purged

YFS\_INV\_OWN\_TRANSFER\_RCD

## Password Reset Request Purge

This purge deletes password reset request data from the system.

You can use purge codes pseudo-logic to analyze purges.

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 281. Password Reset Request Purge Attributes*

Attribute	Value
Base Transaction ID	None
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 282. Password Reset Request Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the PLT_PWD_REQ table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 283. Password Reset Request Purge Statistics*

Statistic Name	Description
NumPasswordRequestPurged	Number of password requests purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the PLT\_PWD\_REQ table.

## Events Raised

None.

## Tables Purged

PLT\_PWD\_REQ

## User Login Failure Purge

This purge deletes data on number of failed login attempts of users from the system.

You can use purge codes pseudo-logic to analyze purges.

Any enterprise using the Console must schedule purge transactions.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 284. User Login Failure Purge Attributes*

Attribute	Value
Base Transaction ID	None
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 285. User Login Failure Purge Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
Live	Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"><li>• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.</li><li>• N - Test mode. Determines the rows that are moved to history tables without actually moving them.</li></ul>
PurgeCode	Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds with the PurgeCode used in Business Rules Purge Criteria.
ColonyID	Required in a multi schema deployment where the PLT_USER_LOGIN_FAILED table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 286. User Login Failure Purge Statistics*

Statistic Name	Description
NumUserLoginFailPurged	Number of failed login attempts purged.

## Pending Job Count

For this transaction, the pending job count is the number of records that can be purged from the PLT\_USER\_LOGIN\_FAILED table.

## Events Raised

None.

## Tables Purged

PLT\_USER\_LOGIN\_FAILED

---

## Task Queue Syncher Time-Triggered Transactions

Many transactions use the task queue as their work repository. The workflow manager automatically creates tasks for transactions to handle the next processing step, as configured in your pipeline.

In some situations, the task queue repository may become out of date. For example, when reconfiguring the processing pipeline while the pipeline is active, the queue may go out of synch with the new pipeline configuration.

Alerts that indicate a halt in the lifecycle of a business document may indicate an outdated task queue repository.

The task queue syncher transactions are designed to update the task queue repository with the latest list of open tasks to be performed by each transaction, based on the latest pipeline configuration.

Some of the statistics collected and tracked in Release 9.1 for time-triggered transactions, monitors, and integration and application servers may change with the next release.

## Load Execution Task Queue Syncher

This transaction synchronizes the task queue for the load execution process type.

You can use the following pseudo-logic to analyze this time-triggered transaction. If the following conditions are met, a task queue for the load execution process type is synchronized:

- LOAD\_CLOSED\_FLAG of Load should not be 'Y'.
- Load should be in a status that is pickable by a transaction in the pipeline.



- There should not be any Task Q record for the load, transaction combination in the Task Q table. In this case, the system inserts one Task Q record for this load, transaction combination with the current database time as the available date.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 287. Load Execution Task Queue Syncher Attributes*

Attribute	Value
Base Transaction ID	TASK_QUEUE_SYNCER_L_D
Base Document Type	Load
Base Process Type	Load Execution
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 288. Load Execution Task Queue Syncher Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 289. Load Execution Task Queue Syncher Statistics*

Statistic Name	Description
NumTasksCreated	Number of tasks created.

## Pending Job Count

None.

## Events Raised

None.

## Order Delivery Task Queue Syncher

This transaction synchronizes the order delivery process type.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 290. Order Delivery Task Queue Syncher Attributes*

Attribute	Value
Base Transaction ID	TASK_QUEUE_SYNCER_O_D
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 291. Order Delivery Task Queue Syncher Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 292. Order Delivery Task Queue Syncher Statistics*

Statistic Name	Description
NumTasksCreated	Number of tasks created.

## Pending Job Count

None.

## Events Raised

None.

## Order Fulfillment Task Queue Syncher

This transaction synchronizes the order fulfillment process type.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 293. Order Fulfillment Task Queue Syncher Attributes*

Attribute	Value
Base Transaction ID	TASK_QUEUE_SYNCER_O_F
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 294. Order Fulfillment Task Queue Syncher Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 295. Order Fulfillment Task Queue Syncher Statistics*

Statistic Name	Description
NumTasksCreated	Number of tasks created.

## Pending Job Count

None.

## Events Raised

None.

## Order Negotiation Task Queue Syncher

This transaction synchronizes the order negotiation process type.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 296. Order Negotiation Task Queue Syncher Attributes*

Attribute	Value
Base Transaction ID	TASK_QUEUE_SYNCER_O_N
Base Document Type	Order
Base Process Type	Order Negotiation
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 297. Order Negotiation Task Queue Syncher Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 298. Order Negotiation Task Queue Syncher Statistics*

Statistic Name	Description
NumTasksCreated	Number of tasks created.

## Pending Job Count

None.

## Events Raised

None.

## Quote Fulfillment Task Queue Syncher

This transaction synchronizes the quote fulfillment process type.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 299. Quote Fulfillment Task Queue Syncher Attributes*

Attribute	Value
Base Transaction ID	TASK_QUEUE_SYNCER_Q_F
Base Document Type	Order
Base Process Type	Quote Fulfillment
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

*Table 300. Quote Fulfillment Task Queue Syncher Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

*Table 301. Quote Fulfillment Task Queue Syncher Statistics*

Statistic Name	Description
NumTasksCreated	Number of tasks created.

## Pending Job Count

None.

## Events Raised

None.

---

## Monitors

Monitors are transactions that watch for processes or circumstances that are out of bounds and then raise alerts.

Some of the statistics collected and tracked in Release 9.1 for time-triggered transactions, monitors, and integration and application servers may change with the next release of Sterling Selling and Fulfillment Foundation.

All Monitors have a CollectPendingJobs criteria parameter. If this parameter is set to N, the agent does not collect information on the pending jobs for that monitor. This pending job information is used for monitoring the monitor in the System Management Console. By default, CollectPendingJobs is set to Y. It can be helpful to set it to N if one monitor is performing a significant amount of getPendingJobs queries and the overhead cost is too high.

## Availability Monitor

This time-triggered transaction monitors inventory availability. The Availability Monitor raises global alerts when the available inventory falls below the configured quantities on the current day, on subsequent days within the ATP time frame, and on subsequent days outside of the ATP time frame. The quantities for the days outside of the ATP time frame are determined by the maximum monitoring days. Unlike the schedule and release transactions, the Availability Monitor calculates the actual availability beyond the ATP horizon and does not assume infinite inventory.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table 302. Availability Monitor Attributes*

Attribute	Value
Base Transaction ID	ATP_MONITOR
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None

### Criteria Parameters

The following are the criteria parameters for this monitor:

*Table 303. Availability Monitor Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
MonitorOption	Optional. Specifies how to monitor inventory. Valid values are: <ul style="list-style-type: none"> <li>• 1 - current inventory</li> <li>• 0 - inventory within and outside of the ATP time frame. This is the default value.</li> </ul>
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
InventoryOrganizationCode	Optional. Valid owner inventory organization. Organization to process in this run. If not passed, all inventory organizations are processed.

Table 303. Availability Monitor Criteria Parameters (continued)

Parameter	Description
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
Status	The negotiation status you are monitoring.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

None.

## Pending Job Count

None.

## Events Raised

No events are raised. Individual actions associated with the monitoring rule are run.

Data published to the actions is AVAILABILITY\_MONITOR\_dbd.txt.

## Exception Monitor

This time-triggered transaction monitors exceptions in your system as noted below. It monitors the exceptions logged in the system and escalates these exceptions:

- If an exception has not been assigned to a user by a certain time
- If an exception has not been resolved by a certain time
- If the active size of the queue is more than a certain maximum size

In order to prevent re-alerts on exceptions during every run of the Exception Monitor, specify a re-alert interval through Alert Management in the Applications Manager. This attribute is associated with a queue and can be configured for each queue.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 304. Exception Monitor Attributes

Attribute	Value
Base Transaction ID	EXCEPTION_MONITOR
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this monitor:

Table 305. Exception Monitor Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
QueueID	Optional. Defines the Alert Queue into which exceptions from this monitor are stored.
OrganizationCode	Optional. Organization to process in this run. If not passed, all inventory organizations are processed.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
QueueGroup	Optional. Defines the set of Queues for which the exceptions will be monitored. If both QueueId and QueueGroup are supplied, QueueId is ignored.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 306. Exception Monitor Statistics

Statistic Name	Description
NumInboxProcessed	Number of alerts processed.
NumExceededQueueSizeAlerts	Number of actions raised when the number of unresolved alerts exceeds the queue's maximum active size.
NumUnResolvedAlerts	Number of actions raised when the unresolved alert time of an alert exceeds the queue's resolution time.
NumUnAssignedAlerts	Number of actions raised when the unassigned alert time of an alert exceeds the queue's assignment time.

## Pending Job Count

None.

## Events Raised

No events are raised. Individual actions associated with the monitoring rule are run.



# Inventory Monitor

This time-triggered transaction monitors inventory availability at ship node level. It raises alerts at the ship node level when the available inventory exceeds or drops below the configured quantities.

This monitor uses the OPEN\_ORDER demand type to calculate available inventory at a given node. All supplies assigned to a supply type that is considered by the OPEN\_ORDER demand type are considered. For more information about configuring inventory supply and demand considerations, refer to the *Sterling Selling and Fulfillment Foundation: Global Inventory Visibility Configuration Guide*.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 307. Inventory Monitor Attributes

Attribute	Value
Base Transaction ID	INVENTORY_MONITOR
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	checkAvailability()

## Criteria Parameters

The following are the criteria parameters for this monitor:

Table 308. Inventory Monitor Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
InventoryOrganizationCode	Optional. Valid inventory owner organization. Organization to process in this run. If not passed, all inventory organizations are processed.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
AllowedOverriddenCriteria	<p>If this parameter is set to Y, the overriding value for the agent criteria parameters can be provided in the command line in the following format when triggering the agent:</p> <pre>&lt;AgentCriteriaAttribute&gt; &lt;OverriddenValue&gt;</pre> <p>For more information about passing these attributes, see the <i>Sterling Selling and Fulfillment Foundation: Installation Guide</i></p>

Table 308. Inventory Monitor Criteria Parameters (continued)

Parameter	Description
ShipNodes	Optional. Comma-separated list of valid ship nodes that should be processed in this run. If not passed, all the ship nodes are processed.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

None.

## Pending Job Count

None.

## Events Raised

No events are raised. Individual actions associated with the monitoring rule are run.

Data published to the actions is <INSTALL\_DIR>/xapidocs/api\_javadocs/dbd/INVENTORY\_MONITOR\_dbd.txt.

## Negotiation Monitor

This time-triggered transaction alerts the Enterprise when a negotiation remains in a particular status for a specific amount of time. This also monitors the negotiation expiration date. This time-triggered transaction invokes the actions configured against the negotiation statuses. Configure status Expired (2000) to monitor negotiation expiration date.

Use this monitor in environments where Order or order release has to go through a negotiation phase and you want to monitor the negotiation.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 309. Negotiation Monitor Attributes

Attribute	Value
Base Transaction ID	ORD_NEGOTIATION_MONITOR
Base Document Type	Order
Base Process Type	Order Negotiation
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this monitor:

Table 310. Negotiation Monitor Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Negotiation Monitor needs to be run. If not passed, then all enterprises are monitored.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
Status	The negotiation status you are monitoring.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 311. Negotiation Monitor Statistics

Statistic Name	Description
NumNegotiationsProcessed	Number of negotiations processed.
NumNegotiationsRequiringAlert	Number of negotiations which have at least one alert raised.

## Pending Job Count

None.

## Events Raised

This invokes the actions configured against the negotiation statuses.

Key Data - Not Applicable.

Data Published - YCP\_getNegotiationDetails\_output.xml

## Enhanced Order Monitor

The enhanced order monitor enables you to monitor the following situations:

- Milestone x has not been reached y hours before a given date type.
- Milestone x has not been reached within y hours of a given date type.
- Milestone x has not been reached within y hours of milestone z.
- Milestone x has been reached y hours before a given date type.
- Milestone x has been reached within y hours of a given date type.
- Milestone x has been reached within y hours after milestone z.

- The order has been in status x for y hours.
- Date type x is y hours before date type z.
- Date type x is y hours after date type z.
- The order has been in hold type x for y hours.
- The order has been in hold type x for y hours before date type z.

The order monitor can be configured to monitor the following system date types for and Purchase Order document types:

- Actual Order Date - Read from the ORDER\_DATE column of the YFS\_ORDER\_HEADER table.
- Actual Next Iteration Date - Read from the NEXT\_ITER\_DATE column of the YFS\_ORDER\_HEADER table.
- Requested Ship Date - If there is an order release, read from the REQ\_SHIP\_DATE column of the YFS\_ORDER\_RELEASE table. Otherwise, read from the REQ\_SHIP\_DATE of the YFS\_ORDER\_LINE table.
- Expected Ship Date - Read from the EXPECTED\_SHIPMENT\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If it is null, uses the same logic as Requested Ship Date.
- Actual Ship Date - If the date is before 01/01/2500, read from the EXPECTED\_SHIPMENT\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If the date is on or after 01/01/2500, this date type is returned as null.
- Requested Delivery Date - If there is a release, read from the REQ\_DELIVERY\_DATE column of the YFS\_ORDER\_RELEASE table.
- Expected Delivery Date - Read from the EXPECTED\_DELIVERY\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If it is null, uses the same logic as Requested Delivery Date.
- Actual Delivery Date - If the date is before 01/01/2500, read from the EXPECTED\_DELIVERY\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If the date is on or after 01/01/2500, this date type is returned as null. For Order Fulfillment, Planned Order Execution, Reverse Logistics, and Purchase Order Execution pipelines, the system defined dates such as Shipment and Delivery are stored without a time component. Therefore when you configure a rule using these dates, all time computations are carried out assuming they are always 12:00:00 AM.

For more information about milestones, date types, and monitoring rules, refer to the *Sterling Selling and Fulfillment Foundation: Supply Collaboration Configuration Guide*, the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*, and the *Sterling Selling and Fulfillment Foundation: Reverse Logistics Configuration Guide*.

If you run the Enhanced Order Monitor, you must configure and run the Close Order time-triggered transaction in all applicable pipelines. For more information about the Close Order time-triggered transaction, see “Close Order” on page 99.

The same relog interval is used for all document types.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 312. Enhanced Order Monitor Attributes

Attribute	Value
Base Transaction ID	ORDER_MONITOR_EX
Base Document Type	Order
Base Process Type	Order Fulfillment
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this monitor:

Table 313. Enhanced Order Monitor Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Order Monitor needs to be run. If not passed, then all enterprises are monitored.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this monitor:

Table 314. Enhanced Order Monitor Statistics

Statistic Name	Description
NumOrdersProcessed	Number of orders processed.
NumAlertsRaised	Number of alerts raised.

## Pending Job Count

For this transaction the pending job count is the number of open orders with the value of NEXT\_ALERT\_TS less than or equal to ( $\leq$ ) the current date.

## Events Raised

The Enhance Order Monitor transaction raises the ON\_AUTO\_CANCEL event, but does not cancel the order. A service on this event should be configured to cancel the order.

Table 315. Events Raised by the Enhanced Order Monitor Transaction

Transaction/Event	Key Data	Data Published*	Template Support?
ON_AUTO_CANCEL	ORDER_MONITOR_dbd.txt	YFS_ORDER_MONITOR_EX.ON_AUTO_CANCEL.html	Yes
<p>* These files are located in the following directory:                      &lt;INSTALL_DIR&gt;/xapidocs/api_javadocs/XSD/HTML</p>			

## Monitor Rule's Condition Template

If a monitor rule contains a condition, the <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX\_CONDITION.xml template file is used to obtain both the order details and the evaluating monitor rule details. See the provided <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX\_CONDITION.xml.sample file for more details.

If the <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX\_CONDITION.xml template file does not exist, the MonitorConsolidation->Order element of the default monitor template, the <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX.xml file, is used.

If the default monitor template is used, the MonitorConsolidation->Order->OrderStatuses-> OrderStatus-> MonitorRule element is ignored and is not passed into the condition.

## Enhanced Quote Monitor

The enhanced quote monitor enables you to monitor the following situations:

- Milestone x has not been reached y hours before a given date type.
- Milestone x has not been reached within y hours of a given date type.
- Milestone x has not been reached within y hours of milestone z.
- Milestone x has been reached y hours before a given date type.
- Milestone x has been reached within y hours of a given date type.
- Milestone x has been reached within y hours after milestone z.
- The order has been in status x for y hours.
- Date type x is y hours before date type z.
- Date type x is y hours after date type z.

The quote monitor can be configured to monitor the following system date types:

- Actual Expiration Date - Read from the EXPIRATION\_DATE column of the YFS\_ORDER\_HEADER table.

For more information about milestones, date types, and monitoring rules, refer to the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

If you run the Enhanced Quote Monitor, you must configure and run the Close Order time-triggered transaction in all applicable pipelines. For more information about the Close Order time-triggered transaction, see “Close Order” on page 99.

The same relog interval is used for all document types.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 316. Enhanced Quote Monitor Attributes*

Attribute	Value
Transaction ID	ORDER_MONITOR_EX.0015
Document Type	Quote
Process Type	Quote Fulfillment
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this monitor:

*Table 317. Enhanced Quote Monitor Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Quote Monitor needs to be run. If not passed, then all enterprises are monitored.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this monitor:

*Table 318. Enhanced Quote Monitor Statistics*

Statistic Name	Description
NumOrdersProcessed	Number of quotes processed.
NumAlertsRaised	Number of alerts raised.

## Pending Job Count

For this transaction the pending job count is the number of open orders with the value of NEXT\_ALERT\_TS less than or equal to ( $\leq$ ) the current date.

## Events Raised

No events are raised. Individual actions associated with the monitoring rule are run.

The data published is ORDER\_MONITOR\_EX.0015.xml.

## Monitor Rule's Condition Template

If a monitor rule contains a condition, the <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX\_CONDITION.xml template file is used to obtain both the order details and the evaluating monitor rule details. See the provided <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX\_CONDITION.xml.sample file for more details.

If the <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX\_CONDITION.xml template file does not exist, the MonitorConsolidation->Order element of the default monitor template, the <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER\_MONITOR\_EX.xml file, is used.

If the default monitor template is used, the MonitorConsolidation->Order->OrderStatuses->OrderStatus->MonitorRule element is ignored and is not passed into the condition.

## Enhanced Return Monitor

The enhanced return monitor allows you to monitor the following situations:

- Milestone x has not been reached y hours before a given date type.
- Milestone x has not been reached within y hours of a given date type.
- Milestone x has not been reached within y hours of milestone z.
- Milestone x has been reached y hours before a given date type.
- Milestone x has been reached within y hours of a given date type.
- Milestone x has been reached within y hours after milestone z.
- The order has been in status x for y hours.
- Date type x is y hours before date type z.
- Date type x is y hours after date type z.

The enhanced return monitor can be configured to monitor the following system date types:

- Actual Order Date - Read from the ORDER\_DATE column of the YFS\_ORDER\_HEADER table
- Requested Ship Date - If there is an order release, read from the REQ\_SHIP\_DATE column of the YFS\_ORDER\_RELEASE table. Otherwise, read from the REQ\_SHIP\_DATE of the YFS\_ORDER\_LINE table.
- Expected Ship Date - Read from the EXPECTED\_SHIPMENT\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If it is null, uses the same logic as Requested Ship Date.
- Actual Ship Date - If the date is before 01/01/2500, read from the EXPECTED\_SHIPMENT\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If the date is on or after 01/01/2500, this date type is returned as null.



- Requested Delivery Date - If there is a release, read from the REQ\_DELIVERY\_DATE column of the YFS\_ORDER\_RELEASE table. Otherwise, read from the REQ\_DELIVERY\_DATE of the YFS\_ORDER\_LINE table.
- Expected Delivery Date - Read from the EXPECTED\_DELIVERY\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If it is null, uses the same logic as Requested Delivery Date.
- Actual Delivery Date - If the date is before 01/01/2500, read from the EXPECTED\_DELIVERY\_DATE column of the YFS\_ORDER\_LINE\_SCHEDULE table. If the date is on or after 01/01/2500, this date type is returned as null. For Order Fulfillment, Planned Order Execution, Reverse Logistics, and Purchase Order Execution pipelines, the system defined dates such as Shipment and Delivery are stored without a time component. Therefore when you configure a rule using these dates, all time computations are carried out assuming they are always 12:00:00 AM.

For more information about milestones, date types, and monitoring rules, refer to the *Sterling Selling and Fulfillment Foundation: Supply Collaboration Configuration Guide*, the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*, and the *Sterling Selling and Fulfillment Foundation: Reverse Logistics Configuration Guide*.

If you run the Enhanced Return Monitor, you must configure and run the Close Order time-triggered transaction in all applicable pipelines. For more information about the Close Order time-triggered transaction, see “Close Order” on page 99.

The same relog interval is used for all document types.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 319. Enhanced Order Monitor Attributes*

Attribute	Value
Base Transaction ID	RETURN_MONITOR_EX
Base Document Type	Return Order
Base Process Type	Reverse Logistics
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this monitor:

*Table 320. Enhanced Order Monitor Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Order Monitor needs to be run. If not passed, then all enterprises are monitored.

Table 320. Enhanced Order Monitor Criteria Parameters (continued)

Parameter	Description
FromStatus	Optional. Statuses to monitor that are greater than or equal to the passed status.
ToStatus	Optional. Statuses to monitor that are less than or equal to the passed status.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this monitor:

Table 321. Enhanced Order Monitor Statistics

Statistic Name	Description
NumOrdersProcessed	Number of orders processed.
NumAlertsRaised	Number of alerts raised.

## Pending Job Count

For this transaction the pending job count is the number of open orders with the value of NEXT\_ALERT\_TS less than or equal to ( $\leq$ ) the current date.

## Events Raised

No events are raised. Individual actions associated with the monitoring rule are run.

The data published is RETURN\_MONITOR\_EX.xml.

## Monitor Rule's Condition Template

If a monitor rule contains a condition, the `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER_MONITOR_EX_CONDITION.xml` template file is used to obtain both the order details and the evaluating monitor rule details. See the provided `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER_MONITOR_EX_CONDITION.xml.sample` file for more details.

If the `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER_MONITOR_EX_CONDITION.xml` template file does not exist, the `MonitorConsolidation->Order` element of the default monitor template, the `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/ORDER_MONITOR_EX.xml` file, is used.

If the default monitor template is used, the `MonitorConsolidation->Order->OrderStatuses->OrderStatus->MonitorRule` element is ignored and is not passed into the condition.

## Real-time Availability Monitor

The Real-time Availability Monitor time-triggered transaction monitors the inventory availability of inventory items. It can be configured to raise the `REALTIME_AVAILABILITY_CHANGE` event when the inventory level for a given item changes between the thresholds defined in the Applications Manager in the Global Inventory Visibility module.

It can be run in three modes:

- Activity Based: Raises the event in real time every time an item goes above or below one of the thresholds.
- Quick Sync: Re-sends the most recently published inventory availability information.
- Full Sync: Monitors all of the items regardless of activity and publishes the inventory information for all of the items.

In all cases, the percentage of future inventory availability is used for considering inventory availability at retrieval time. For more information about future inventory availability, see the *Sterling Selling and Fulfillment Foundation: Global Inventory Visibility Configuration Guide*.

Demand of type `OPEN_ORDER` is used in getting the inventory availability picture. If sourcing is maintained, the Real-time Availability Monitor can either monitor the total availability across nodes or the availability at individual nodes. Inventory items without an Availability Monitor rule, or with a rule that is disabled, are unable to be processed by this time-triggered transaction.

If item information is not available on the system, such as when inventory between databases is not in sync, default inventory monitor rules can be configured so that monitoring will proceed. Setting these default inventory monitor rules also ensures that inventory activity will be recorded when the system is running in activity-based mode.

When monitoring the total availability across nodes, the Real-time Availability Monitor monitors all nodes in the default distribution group of the inventory organization.

When monitoring the availability at individual nodes, the Real-time Availability Monitor monitors all nodes in a specified distribution group. For more information about configuring distribution groups and node-level inventory monitoring, see the *Sterling Selling and Fulfillment Foundation: Global Inventory Visibility Configuration Guide*.

If configured, the Real-time Availability Monitor also considers the onhand and future inventory availability safety factor during monitoring. For more information about the inventory availability safety factors and the `findInventory()` API, see the *Sterling Selling and Fulfillment Foundation: Global Inventory Visibility Configuration Guide* and the *Sterling Selling and Fulfillment Foundation: Javadocs*.

When the onhand quantity is greater than the configured low threshold, the `REALTIME_ONHAND` alert type is raised, and the alert level is based on the onhand quantity.

When the onhand quantity falls below the configured low threshold, the `REALTIME_FUTURE_MAX` alert type is raised, and the alert level is based on the total

future supply (FutureAvailableQuantity) with FirstFutureAvailableDate set to the date on which the first future supply is available, and FutureAvailableDate set to the date on which the maximum future supply is available.

When the Real-time Availability Monitor is run in activity based mode, changing one of the thresholds of an inventory item does not cause the agent to monitor it unless there is a change in activity. For example, if item I with available quantity 700 is being monitored with a low threshold of 600, and the low threshold is then changed to 1000, no event is published unless there is change in I's activity. In order to ensure that in such a scenario I is not left unmonitored, call the createInventoryActivity API when changing a monitoring rule for an item.

## Computing and Publishing the Maximum Ship Dates for Available Quantities

If enabled, the Real-Time Availability Monitor computes and publishes a matrix of maximum ship dates for available quantities, which includes the following information:

- Available Quantity - Refers to the number of items that are available for shipping on the maximum ship date.
- Maximum Ship Date - Refers to the time and date when available quantities are shipped by.
- Effective Until Date - Refers to the last time and date that an order can be placed if it is to be shipped by the maximum ship date.

The matrix is published to the REALTIME\_AVAILABILITY\_CHANGE event and stored in XML format in the AVAILABILITY\_INFO field of the YFS\_INVENTORY\_ALERTS table. The monitorItemAvailability() API can be used to update the matrix. For more information about the monitorItemAvailability() API, refer to the *Sterling Selling and Fulfillment Foundation: Javadocs*.

For information about using the Real-Time Availability Monitor to calculate and publish a matrix of maximum ship dates for available quantities, refer to the chapter on Configuring Inventory Rules in the *Sterling Selling and Fulfillment Foundation: Global Inventory Visibility Configuration Guide*.

### Computing the Maximum Ship Date

The maximum ship date is equal to the maximum expected ship date across all the nodes being considered. For information about calculating the expected ship date, refer to the *Sterling Selling and Fulfillment Foundation: Product Concepts Guide*. Additionally, the following options can be configured as part of the maximum ship date:

- Maximum Ship Date Time
- Number of Days To Offset the Maximum Ship Date

**Maximum Ship Date Time** - If you specify a time for the maximum ship date, the Real-Time Availability Monitor calculates the maximum ship date, as described earlier, and then applies the following logic:

- If the time specified for the maximum ship date occurs later in the day than the calculated ship date, the Real-Time Availability Monitor resets the maximum ship date to the specified time. For example, if the Real-Time Availability

Monitor calculates the maximum ship date to be 10 a.m. on July 21 and Maximum Ship Date Time is set to 11 a.m., the maximum ship date is recalculated to be 11 a.m. on July 21.

- If the time specified for the maximum ship date occurs earlier in the day than the calculated ship date, the maximum ship date is incremented by one day and reset to the specified time. For example, if the maximum ship date is calculated to be 11 a.m. on July 21 and Maximum Ship Date Time is set to 10 a.m., the Real-Time Availability Monitor recalculates the maximum ship date to be 10 a.m. on July 22.

**Number of Days To Offset the Maximum Ship Date** - You can specify a number of days to offset the maximum ship date. The Real-Time Availability Monitor calculates the maximum ship date, including the maximum ship date time, and then increments the maximum ship date by the number of days specified by the offset number. For example, if the Real-Time Availability Monitor has calculated a maximum ship date to be 11 a.m. on July 19 and Number of Days to Offset the Maximum Ship Date is set to 1, the maximum ship date is recalculated to be 11 a.m. on July 20.

### Calculating the Effective Until Date

The Real-Time Availability Monitor calculates the effective until date by subtracting the node's minimum notification time from the maximum ship date and then adjusting for the preceding notification time on the node's notification schedule. The effective until date is only valid while supplies are available at the node.

For example, if an available quantity has a maximum ship date of 4 p.m. on July 19 and the shipping node has the following notification schedule, the effective until date is calculated to be 3 p.m. on July 18:

- 24-hour minimum notification time
- 3 p.m. and 5 p.m. notification times

In this example, the effective until date is calculated by first subtracting the 24-hour minimum notification time from the 4 p.m., July 19 maximum ship date and then adjusting for the 3 p.m. notification time. If an order is not placed before 3 p.m. on July 18, the July 19 maximum ship date is no longer available because the node must be notified at least 24 hours before shipping the items, by 4 p.m. on July 19. Also, if a different order reduces available quantities at the node before the order is placed at 3 p.m. on July 19, the maximum ship date cannot be met and the effective until date becomes invalid.

Additionally, offset days are not considered when calculating the effective until date. Thus, if the maximum ship date in the earlier example is updated to 4 p.m. July 20 by setting Number of Days to Offset Maximum Ship Date to 1, the effective until date is updated to 3 p.m., July 19.

### Example 1: Computing Maximum Ship Dates for Available Quantities

Node 1 has the following supply picture:

- 24-hour minimum notification time
- Notification times are 3 p.m. and 5 p.m. daily
- Work Days are 24 hours-a-day, 7 days-a-week

Node 2 has the following supply picture:

- 48-hour minimum notification time
- Notification times are 2 p.m. and 5 p.m. daily
- Work Days are 24 hours-a-day, 7 days-a-week

The following table shows the availability matrix for Node 1 and Node 2, where the following conditions are true:

- Current date is July 19
- Estimated time of arrival (ETA) equals the date that the quantity is expected to be available at the node
- Maximum Ship Date Time is set to 4 p.m.
- Number of Days to Offset the Maximum Ship Date is set to 0

*Table 322. Example: Availability Matrix of Maximum Ship Dates for Available Quantities*

ETA	Quantity	Maximum Ship Date	Effective Until Date
Node 1			
7/19/2010	80	4 p.m., July 20	3 p.m., July 19
7/22/2010	10	4 p.m. July 22	3 p.m., July 21
Node 2			
7/19/2010	100	4 p.m., July 21	2 p.m., July 19
7/22/2010	20	4 p.m., July 22	2 p.m., July 20

In this example, July 19 is the ETA for a quantity of 80 items at Node 1 and 100 items at Node 2. The matrix shows a 4 p.m., July 20 maximum ship date for the 80 available items from Node 1 and a 4 p.m., July 21 maximum ship date for the 100 available items from Node 2. For Node 1, the maximum ship date is calculated by adding the 24-hour minimum notification time to the 3 p.m. notification time on July 19, and then adjusting for the 4 p.m. maximum ship date time. The effective until date is calculated by subtracting the 24-hour minimum notification time from the maximum ship date and then adjusting for the 3 p.m. notification time. For Node 2, the maximum ship date and effective until date are calculated similarly, with the exception that Node 2 has a 48-hour minimum notification time and a 2 p.m. notification time.

Additionally, the example shows July 22 as the ETA for a quantity of 10 items at Node 1 and 20 items at Node 2. The maximum ship date is 4 p.m., July 22 for the 10 items at Node 1 and 4 p.m., July 22 for the 20 items at Node 2. If the difference between the current date and the ETA is greater than the node's minimum notification time, the ETA date is used for the maximum ship date. In this example, the difference between the current date, July 19, and the ETA date, July 22, is greater than the minimum notification times at both nodes. Thus, the maximum ship date is set to the maximum ship date time on the ETA date at the nodes, which is 4 p.m., July 22 at Node 1 and 4 p.m., July 22 at Node 2.

### **Example 2: Computing the Maximum Ship Date at Nodes With Non-Working Days**

The following table displays the availability matrix for Node 1 and Node 2 when the supply picture and conditions from Example 1 are applied. However, in this scenario, July 19 and July 20 are nonworking days.

Table 323. Example: Availability Matrix for Nodes with Non-Working Days

ETA	Quantity	Maximum Ship Date	Effective Until Date
Node 1			
7/19/2010	80	4 p.m., July 22	3 p.m., July 21
Node 2			
7/19/2010	100	4 p.m., July 23	2 p.m., July 21

In the example, Node 1 has an available quantity of 80 on July 19 and a minimum notification time of 24 hours. Because July 19 and July 20 are nonworking days at Node 1, the 80 items are not considered available until July 21. In this case, the maximum ship date is calculated by adding the 24-hour minimum notification time to July 21 and adjusted for the 4 p.m. maximum ship date time. For Node 2, the maximum ship date is calculated similarly, with the exception of a 48-hour minimum notification time.

### Example 3: Offsetting the Maximum Ship Date

The following table displays the availability matrix for Node 1 and Node 2 when the supply picture and conditions from Example 2 are applied. However, in this scenario, Number of Days To Offset the Maximum Ship Date is set to 1.

Table 324. Example: Availability Matrix When Offsetting the Maximum Ship Date

ETA	Quantity	Maximum Ship Date	Effective Until Date
Node 1			
7/19/2010	80	4 p.m., July 23	3 p.m., July 22
Node 2			
7/19/2010	100	4 p.m., July 24	2 p.m., July 22

In the example, the maximum ship dates for Nodes 1 and 2 are calculated similarly to Example 2. However, the maximum ship dates are incremented by 1 because Number of Days to Offset the Maximum Ship Date is set to 1. In this example, the effective until date is set to 3 p.m., July 22 for Node 1 and 2 p.m., July 22 for Node 2 because the offset days are not considered when calculating the effective until date.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 325. Real-time Availability Monitor Attributes

Attribute	Value
Base Transaction ID	REALTIME_ATP_MONITOR
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	FindInventory

## Criteria Parameters

The following are the criteria parameters for this monitor:

Table 326. Real-time Availability Monitor Criteria Parameters

Parameter	Description
<b>Action</b>	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
<b>Number of Records To Buffer</b>	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
<b>InventoryOrganizationCode</b>	Inventory organization code to use when MonitorOption is passed as 3. The inventory organization has to be an enterprise.  If this is not passed, the monitor runs for all inventory organizations.
<b>MonitorOption</b>	1 - Activity Based (Monitor based on distinct inventory items in YFS_INVENTORY_ACTIVITY table).  2 – Quick Sync (Re-raise event to publish information from the YFS_INVENTORY_ALERTS table).  3 – Full Sync (Monitor based on all inventory items maintained by the inventory organization provided. If no InventoryOrganizationCode is provided, all inventory item is monitored).  If not provided, default value is 1.
<b>ItemStatuses</b>	List of valid statuses of items to be processed. Statuses must be separated by a , for example 3000,2000. This is only used when MonitorOption is passed as 2 or 3. If provided, only items with the matching statuses is monitored.
<b>FromAlertTimestamp</b>	This is only used when MonitorOption is passed as 2. If provided, the agent raises the REALTIME_AVAILABILITY_CHANGE event to republish inventory availability information which was published between the time that the agent started and FromAlertTimestamp.  If not provided, all inventory availability information published before the time that the agent started is republished.
<b>AllowedOverriddenCriteria</b>	If set to Y, the overridden value for the agent criteria parameters can be provided at the command line while triggering the agent in the following format:  <AgentCriteriaAttribute> <OverriddenValue>  For more information about passing these attributes, see the <i>Sterling Selling and Fulfillment Foundation: Installation Guide</i> .



Table 326. Real-time Availability Monitor Criteria Parameters (continued)

Parameter	Description
<b>FromLastNumberOfHours</b>	This is only used when MonitorOption is passed as 2 to calculate the FromAlertTimestamp parameter, if necessary.  If the FromAlertTimestamp parameter is not provided, it is calculated as current timestamp minus FromLastNumberOfHours.
<b>CollectPendingJobs</b>	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
<b>RaiseEventsOnAllAvailability Changes</b>	When set to Y, REALTIME_AVAILABILITY_CHANGE event is raised on all availability changes regardless of whether availability exceeds or falls below specified thresholds. This is only used when MonitorOption is passed as 1. Valid values: Y or N. Default value: N.
<b>ColonyID</b>	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

**Note:** Distribution Rule Id is used for internal purposes and should be left blank.

### Statistics Tracked

None.

### Pending Job Count

None.

### Events Raised

The following events are raised by this time-triggered transaction:

Table 327. Events Raised by the Realtime Availability Monitor Transaction

Transaction/Event	Key Data	Data Published*	Template Support?
<b>REALTIME_AVAILABILITY_CHANGE</b>	None	YFS_REALTIME_ATP_MONITOR. REALTIME_AVAILABILITY_CHANGE.html	Yes

\* These files are located in the following directory:  
<INSTALL\_DIR>\xapidocs/api\_javadocs/XSD/HTML

Although described as 'real time', availability changes may not be triggered immediately as inventory changes occur if the agent has a backlog of messages to process. Furthermore, this monitor exists as a time-triggered transaction, and thus monitors availability of inventory items only when the monitor is triggered based on the configured runtime properties.

# Shipment Monitor

This time-triggered transaction reports the states of a shipment, based on rules in the YFS\_MONITOR\_RULE table. This transaction enables you to monitor the following situations:

- If the Shipment has been in a status for more than a specified amount of time.
- If a specified date that is associated with the shipment is:
  - n hours before another specified date
  - n hours after another specified date
  - n hours not before another specified date
  - n hours not after another specified date
- If the Shipment has been in a hold type for a specified amount of time.
- If the Shipment has been in a hold type for n hours before a specified date.

Monitoring rules can be configured for shipment's origin and destination points.

Monitoring rules cannot be configured for a shipment's intermediate pickup and drop off points. A shipment has intermediate pickup or drop off only if it has multiple pickup or drop off points. For example, a shipment has more than one loads carrying it. The shipment status on first load deposit, second load deposit, and so forth cannot be monitored. Once the last load deposits the shipment at its destination, then the shipment status can be marked and monitored.

This is not a pipeline transaction. It also does not work from the task queue.

For more information about milestones, date types, and monitoring rules, see the *Sterling Selling and Fulfillment Foundation: Supply Collaboration Configuration Guide*, the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*, and the *Sterling Selling and Fulfillment Foundation: Reverse Logistics Configuration Guide*.

## Attributes

The following are the attributes for this time-triggered transaction:

Table 328. Shipment Monitor Attributes

Attribute	Value
Base Transaction ID	SHIPMENT_MONITOR
Base Document Type	Order
Base Process Type	Order Delivery
Abstract Transaction	No
APIs Called	None

## Criteria Parameters

The following are the criteria parameters for this monitor:

Table 329. Shipment Monitor Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.

Table 329. Shipment Monitor Criteria Parameters (continued)

Parameter	Description
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Shipment Monitor needs to be run. If not passed, then all enterprises are monitored.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 330. Shipment Monitor Statistics

Statistic Name	Description
NumShipmentsMonitored	Number of shipments monitored.

## Pending Job Count

For this transaction the pending job count is the number of open shipments with the value of NEXT\_ALERT\_TS less than or equal to (<=) the current date.

## Events Raised

This invokes the actions configured against shipment statuses.

Key Data - Not Applicable.

Data Published - SHIPMENT\_MONITOR.xml

## Monitor Rule's Condition Template

If a monitor rule contains a condition, the <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/SHIPMENT\_MONITOR\_CONDITION.xml template file is used to obtain the shipment details and the evaluating monitor rule details. See the provided <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/SHIPMENT\_MONITOR\_CONDITION.xml.sample file for more details.

If the <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/SHIPMENT\_MONITOR\_CONDITION.xml template file does not exist, the MonitorConsolidation->Shipment element of the default monitor template, the <INSTALL\_DIR>/repository/xapi/template/source/smcfs/monitor/SHIPMENT\_MONITOR.xml file, is used.

If the default monitor template is used, the MonitorConsolidation->Shipment->MonitorRule element is ignored and is not passed into the condition.

## Work Order Monitor

This time-triggered transaction alerts the enterprise when a work order remains in a particular state or hold type for a specific amount of time.

Use this monitor to track how long work orders stay in a particular state or hold type.

### Attributes

The following are the attributes for this time-triggered transaction:

*Table 331. Work Order Monitor Attributes*

Attribute	Value
Base Transaction ID	WORK_ORDER_MONITOR
Base Document Type	Work Order
Base Process Type	VAS Process
Abstract Transaction	No

### Criteria Parameters

The following are the criteria parameters for this monitor:

*Table 332. Work Order Monitor Criteria Parameters*

Parameter	Description
Action	Required. Triggers the transaction. If left blank it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
EnterpriseCode	Optional. Enterprise for which the Work Order Monitor needs to be run. If not passed, then all enterprises are monitored.
Node	Optional. Node for which the Work Order Monitor needs to be run. If not passed, then all nodes are monitored.
CollectPendingJobs	If this parameter is set to N, the agent does not collect information on the pending jobs for this monitor. This pending job information is used for monitoring the monitor in the System Management Console.
ColonyID	Required in a multi schema deployment where a table may exist in multiple schemas. Runs the agent for the colony.

### Statistics Tracked

The following statistics are tracked for this transaction:

*Table 333. Work Order Monitor Statistics*

Statistic Name	Description
NumWorkOrdersMonitored	Number of work orders monitored.

## Pending Job Count

For this transaction the pending job count is the number of Work Orders that are monitored, where NEXT\_ALERT\_TS less than or equal to ( $\leq$ ) current date.

## Events Raised

No events are raised. Individual actions associated with the monitoring rule are run. Data published to the actions is workOrder\_dbd.txt.

## Monitor Rule's Condition Template

If a monitor rule contains a condition, the `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/monitor/WORK_ORDER_MONITOR_CONDITION.xml` template file is used to obtain the work order details and the evaluating monitor rule details. See the provided `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/WORK_ORDER_MONITOR_CONDITION.xml.sample` file for more details.

If the `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/WORK_ORDER_MONITOR_CONDITION.xml` template file does not exist, the MonitorConsolidation->WorkOrder element of the default monitor template, the `<INSTALL_DIR>/repository/xapi/template/source/smcfs/monitor/WORK_ORDER_MONITOR.xml` file, is used.

If the default monitor template is used, the MonitorConsolidation-> WorkOrder-> MonitorRule element is ignored and is not passed into the condition.



---

## Chapter 20. Condition Builder Attributes

---

### Condition Builder Attributes

Statements in the condition builder are built using attributes that are defined throughout the Applications Manager.

These attributes are grouped as follows:

#### **Sales Order**

- Order fulfillment
- Order negotiation
- Outbound shipment
- Receipt

#### **Planned Order**

- Planed order execution
- Planned order negotiation

#### **Return Order**

- Reverse logistics
- Return shipment
- Return receipt

#### **Template Order**

- Template order

#### **Purchase Order**

- Purchase order execution
- Purchase order negotiation
- Inbound shipment
- Purchase order receipt

#### **Transfer Order**

- Transfer order execution
- Transfer order delivery
- Transfer order receipt

#### **Master Order**

- Master order fulfillment

#### **Quote**

- Quote fulfillment

#### **Load**

- Load execution

## General

- General
- WMS putaway
- WMS layout definition
- WMS inventory
- Trailer loading
- Task execution
- Move request execution
- Manifesting
- Over pack build

## Count

- Count execution

## Container

- Pack process

## Wave

- Outbound picking

## Work Order

- VAS process

## Opportunity

- Opportunity fulfillment

## Item-Based Allocation (IBA)

- Item-based allocation (IBA) order

---

## Sales Order

### Order Fulfillment

The Condition Builder attributes for Order Fulfillment, Order Execution, Quote Fulfillment, Transfer Order Execution, and Template Order are identical.

*Table 334. Order Fulfillment Condition Builder Attributes*

Attribute	Description
<b>Order Attributes</b>	
<b>Condition Variable 1</b>	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
<b>Condition Variable 2</b>	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
<b>Delivery Method</b>	The delivery method of the order (shipment, pickup or delivery).



Table 334. Order Fulfillment Condition Builder Attributes (continued)

Attribute	Description
<b>Disposition Code</b>	The disposition code of the item. This field is only applicable for Reverse Logistics and Supply Collaboration.
<b>Line Type</b>	The type of the order line. Sterling Selling and Fulfillment Foundation has no application logic associated with the order line type. This field can be set up as per your business practices.
<b>Order Type</b>	The type of the order. Sterling Selling and Fulfillment Foundation has no application logic associated with the order type. This field can be set up as per your business practices.
<b>Payment Status</b>	The payment status of the order.
<b>Sale Voided</b>	The flag indicating whether the order is voided.
<b>Transaction ID</b>	The ID of the last transaction that was run on the order.
<b>Participant Attributes</b>	
<b>Bill To ID</b>	The ID of the bill to address for the order.
<b>Buyer Organization Code</b>	The code of the organization that is buying the goods or services.
<b>Enterprise Code</b>	The code of the enterprise on the order.
<b>Receiving Node</b>	The node that receives the shipment for the order.
<b>Seller Organization Code</b>	The code of the organization that is selling the goods or services.
<b>Ship Node</b>	The node that ships the shipment for the order.
<b>Ship Node Interface Type</b>	The interface type of the ship node on the order (External Application, Console, Sterling Store Inventory Management, or WMS 6.2).
<b>Ship To ID</b>	The ID of the ship to address for the order.
<b>Supplier Code</b>	The code of the supplier for the order.
<b>Item Attributes</b>	
<b>Item ID</b>	The ID of the item on the order line.
<b>Item Group Code</b>	The group code of the service item. For example, if the service is a provided service item, then the item group code is PS.
<b>Product Line</b>	The product line of the item on the order line.
<b>Sourcing Attributes</b>	
<b>Fulfillment Type</b>	The fulfillment type of the order.
<b>Intentional Backorder</b>	The flag indicating whether the order was intentionally dropped into backordered status at order creation.
<b>Is Firm Predefined Node</b>	The flag indicating whether the node on the order is a firm predefined node.
<b>Order Sourcing Classification</b>	The order sourcing classification of the order.
<b>Reservation Mandatory</b>	The flag indicating whether the reservation is mandatory.
<b>Related Order Attributes</b>	
<b>Chain Type</b>	The chain type of the order.
<b>Is Chained Line</b>	The flag indicating whether the order line is chained with another order line.

Table 334. Order Fulfillment Condition Builder Attributes (continued)

Attribute	Description
<b>Is Derived Line</b>	The flag indicating whether the order line is derived from another order line.
<b>Order Purpose</b>	The purpose of the order. If this is an exchange order, this field is set to EXCHANGE.
<b>{Enter Your Own Attribute}</b>	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## Order Negotiation

The Condition Builder attributes for Order Negotiation and Planned Order Negotiation are identical.

Table 335. Order Negotiation Condition Builder Attributes

Attribute	Description
<b>Enterprise Code</b>	The code of the enterprise on the order.
<b>Initiator Organization Code</b>	The code of the organization that initiates the negotiation.
<b>Negotiator Organization Code</b>	The code of the organization that can accept, counter-offer, or reject the initiator's offer.
<b>Negotiation Pipeline Key</b>	The key of the negotiation pipeline this order is going through.
<b>Negotiation Number</b>	The negotiation number of this order.
<b>Negotiation Rule Key</b>	The key of the negotiation rule for this order.
<b>Header Entity</b>	The entity for which the negotiation was initiated. Currently, the only applicable entity is Order.
<b>Negotiation Status</b>	The status of the negotiation for this order.
<b>Document Type</b>	The document type for this order. Typical value is Sales Order.
<b>Freight Terms</b>	The freight terms for this order.
<b>Payment Terms</b>	The payment terms for this order.
<b>{Enter Your Own Attribute}</b>	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## Outbound Shipment

The condition builder attributes for Outbound Shipment, Inbound Shipment, Transfer Order Delivery, and Return Shipment are identical.

Table 336. Outbound Shipment Condition Builder Attributes

Attribute	Description
<b>Enterprise Code</b>	The code of the enterprise on the outbound shipment.
<b>Buyer Organization Code</b>	The code of the organization that is buying the goods or services.
<b>Seller Organization Code</b>	The code of the organization that is selling the goods or services.
<b>Ship Node</b>	The node that ships this shipment.
<b>Ship Node Interface Type</b>	The interface type of the ship node on the order (External Application, Console, Sterling Store Inventory Management, or WMS 6.2).
<b>Receiving Node</b>	The node that receives this shipment.
<b>Ship Mode</b>	The shipment mode that is used for the shipment. For example, Parcel, Truck Load, Less-Than Truck Load.
<b>Freight Terms</b>	The freight terms for this shipment.
<b>Carrier Type</b>	The shipment's carrier type for this shipment.
<b>Hazardous Materials Flag</b>	The flag indicating whether these materials are hazardous.
<b>ESP Check Required</b>	The flag indicating whether an Economic Shipping Parameters check is required at shipment consolidation time.
<b>Is Appointment Required</b>	The flag indicating whether an appointment is required for a service execution.
<b>Routing Guide Maintained</b>	The flag indicating whether a routing guide is maintained for this shipment.
<b>Carrier</b>	The carrier for the shipment.
<b>Real-time Integration with WMS 6.2</b>	The flag indicating whether the node this shipment is shipping from is integrating with the Sterling Store Inventory Management. Setting this field to N means that you are integrating with WMS 6.2, or any other warehouse management system.
<b>Manually Entered</b>	The flag indicating whether or not the shipment was entered through the Console.
<b>Delivery Code</b>	The code of the entity that pays for the transportation costs.
<b>Country/Region</b>	The country or region that the shipment is being shipped to.
<b>Delivery Method</b>	The delivery method of the shipment (shipment, pickup or delivery).
<b>Is Serial Requested</b>	The flag indicating whether the shipment has any line with a specific serial number passed. If that is the case, a different outbound shipment process can be selected in the pipeline.
<b>Is Provided Service</b>	The flag indicating whether the shipment has an associated provided service item.
<b>Shipment Type</b>	Indicates a set of shipments that are of the same nature.

Table 336. Outbound Shipment Condition Builder Attributes (continued)

Attribute	Description
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## Receipt

The Receipt condition builder attributes are identical to the Return Receipt attributes.

---

## Planned Order

### Planned Order Execution

The Planned Order Execution condition builder attributes are identical to the Order Fulfillment attributes.

### Planned Order Negotiation

The Planned Order Negotiation condition builder attributes are identical to the Order Negotiation attributes.

---

## Return Order

### Reverse Logistics

Table 337. Return Fulfillment Condition Builder Attributes

Attribute	Description
<b>Order Attributes</b>	
<b>Condition Variable 1</b>	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
<b>Condition Variable 2</b>	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
<b>Delivery Method</b>	The delivery method of the return (shipment, pickup or delivery).
<b>Disposition Code</b>	The disposition code of the item.
<b>Line Type</b>	The type of the return line. Sterling Selling and Fulfillment Foundation has no application logic associated with the return line type. This field can be set up as per your business practices.

Table 337. Return Fulfillment Condition Builder Attributes (continued)

Attribute	Description
<b>Order Type</b>	The type of the return. Sterling Selling and Fulfillment Foundation has no application logic associated with the return type. This field can be set up as per your business practices.
<b>Payment Status</b>	The payment status of the return.
<b>Sale Voided</b>	The flag indicating whether the return is voided.
<b>Transaction ID</b>	The ID of the last transaction that was run on the return.
<b>Participant Attributes</b>	
<b>Bill To ID</b>	The ID of the bill to address for the return.
<b>Buyer Organization Code</b>	The code of the organization that is buying the goods or services.
<b>Enterprise Code</b>	The code of the enterprise on the return.
<b>Receiving Node</b>	The node that receives the shipment for the return.
<b>Seller Organization Code</b>	The code of the organization that is selling the goods or services.
<b>Ship Node</b>	The node that be ships the shipment for the return.
<b>Ship Node Interface Type</b>	The interface type of the ship node on the return (External Application, Console, Sterling Store Inventory Management, or WMS 6.2).
<b>Ship To ID</b>	The ID of the ship to address for the return.
<b>Supplier Code</b>	The code of the supplier for the return.
<b>Item Attributes</b>	
<b>Item ID</b>	The ID of the item on the return line.
<b>Item Group Code</b>	The group code of the service item. For example, if the service is a provided service item, then the item group code is PS.
<b>Product Line</b>	The product line of the item on the return line.
<b>Sourcing Attributes</b>	
<b>Fulfillment Type</b>	The fulfillment type of the return.
<b>Intentional Backorder</b>	The flag indicating whether the return was intentionally dropped into backordered status at return creation.
<b>Is Firm Predefined Node</b>	The flag indicating whether the node on the return is a firm predefined node.
<b>Order Sourcing Classification</b>	The order sourcing classification of the return.
<b>Reservation Mandatory</b>	The flag indicating whether the reservation is mandatory.
<b>Related Order Attributes</b>	
<b>Chain Type</b>	The chain type of the return.
<b>Is Chained Line</b>	The flag indicating whether the return line is chained with another return line.
<b>Is Derived Line</b>	The flag indicating whether the return line is derived from another return line.
<b>Order Purpose</b>	This field is only applicable to sales orders.

Table 337. Return Fulfillment Condition Builder Attributes (continued)

Attribute	Description
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## Return Shipment

The Return Shipment condition builder attributes are identical to the Outbound Shipment attributes.

## Return Receipt

The Condition Builder attributes for Receipt, Purchase Order Receipt, Return Receipt, Transfer Order Receipt are identical.

Table 338. Return Receipt Condition Builder Attributes

Attribute	Description
Document Type	The document type on the receipt. Typical value is Return Order.
Enterprise Code	The code of the enterprise that owns the receipt.
Seller Organization Code	The code of the organization that is selling the goods or services.
Ship Node	The node where the shipment was shipped out of.
Buyer Organization Code	The code of the organization that is buying the goods or services.
Receiving Node	The node where the shipment was received.
Receiving Node Interface Type	The interface type of the receiving node on the order (External Application, Console, Sterling Store Inventory Management, or WMS 6.2).
Ship Mode	The shipment mode that is used for the shipment. For example, Parcel, Truck Load, Less-Than Truck Load.
Freight Terms	The freight terms on the receipt.
Carrier Type	The carrier type on the receipt.
Is Hazardous Material	The flag indicating whether there are hazardous materials that are being received.
Is Inspection Pending	The flag indicating whether there is an inspection pending on this return.
Is Receiving Node Integrated Real Time	The flag indicating whether the receiving node is integrating with WMS 6.2, or with another WMS system.

Table 338. Return Receipt Condition Builder Attributes (continued)

Attribute	Description
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## Template Order

The Template Order condition builder attributes are identical to the Order Fulfillment attributes.

## Purchase Order

### Purchase Order Execution

Table 339. Purchase Order Execution Condition Builder Attributes

Attribute	Description
<b>Order Attributes</b>	
Condition Variable 1	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
Condition Variable 2	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
Delivery Method	The delivery method of the inbound order (shipment, pickup or delivery).
Disposition Code	The disposition code of the item.
Line Type	The type of the inbound order line. Sterling Selling and Fulfillment Foundation has no application logic associated with the inbound order line type. This field can be set up as per your business practices.
Order Type	The type of the inbound order. Sterling Selling and Fulfillment Foundation has no application logic associated with the inbound order type. This field can be set up as per your business practices.
Payment Status	The payment status of the inbound order.
Sale Voided	The flag indicating whether the inbound order is voided.
Transaction ID	The ID of the last transaction that was run on the inbound order.
<b>Participant Attributes</b>	
Bill To ID	The ID of the bill to address for the inbound order.
Buyer Organization Code	The code of the organization that is buying the goods or services.

Table 339. Purchase Order Execution Condition Builder Attributes (continued)

Attribute	Description
Enterprise Code	The code of the enterprise on the inbound order.
Receiving Node	The node that receives the shipment for the inbound order.
Seller Organization Code	The code of the organization that is selling the goods or services.
Ship Node	The node that ships the shipment for the inbound order.
Ship Node Interface Type	The interface type of the ship node on the inbound order (External Application, Console, Sterling Store Inventory Management, or WMS 6.2).
Ship To ID	The ID of the ship to address for the inbound order.
Supplier Code	The code of the supplier for the inbound order.
<b>Item Attributes</b>	
Item ID	The ID of the item on the inbound order line.
Item Group Code	The group code of the service item. For example, if the service is a provided service item, then the item group code is PS.
Product Line	The product line of the item on the inbound order line.
<b>Sourcing Attributes</b>	
Fulfillment Type	The fulfillment type of the inbound order.
Intentional Backorder	The flag indicating whether the inbound order was intentionally dropped into backordered status at inbound order creation.
Is Firm Predefined Node	The flag indicating whether the node on the inbound order is a firm predefined node.
Order Sourcing Classification	The order sourcing classification of the inbound order.
Reservation Mandatory	The flag indicating whether the reservation is mandatory.
<b>Related Order Attributes</b>	
Chain Type	The chain type of the inbound order.
Is Chained Line	The flag indicating whether the inbound order line is chained with another inbound order line.
Is Derived Line	The flag indicating whether the inbound order line is derived from another inbound order line.
Order Purpose	This field is only applicable to sales orders.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>



## Purchase Order Negotiation

Table 340. Purchase Order Negotiation Condition Builder Attributes

Attribute	Description
Enterprise Code	The code of the enterprise on the inbound order.
Initiator Organization Code	The code of the organization that initiates the negotiation.
Negotiator Organization Code	The code of the organization that can accept, counter-offer, or reject the initiator's offer.
Negotiation Pipeline Key	The key of the negotiation pipeline this inbound order is going through.
Negotiation Number	The negotiation number of this inbound order.
Negotiation Rule Key	The key of the negotiation rule for this inbound order.
Header Entity	The entity for which the negotiation was initiated. Currently, the only applicable entity is Order.
Negotiation Status	The status of the negotiation for this inbound order.
Document Type	The document type for this inbound order. Typical value is Purchase Order.
Freight Terms	The freight terms for this inbound order.
Payment Terms	The payment terms for this inbound order.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## Inbound Shipment

The Inbound Shipment condition builder attributes are identical to the Outbound Shipment attributes.

## Purchase Order Receipt

The Purchase Order Receipt condition builder attributes are identical to the Return Receipt attributes.

---

## Transfer Order

### Transfer Order Execution

The Transfer Order Execution condition builder attributes are identical to the Order Fulfillment attributes.

### Transfer Order Delivery

The Transfer Order Delivery condition builder attributes are identical to the Outbound Shipment attributes.

## Transfer Order Receipt

The Transfer Order Receipt condition builder attributes are identical to the Return Receipt attributes.

### Master Order Fulfillment

Table 341. Master Order Fulfillment Condition Builder Attributes

Attribute	Description
<b>Master Order Attributes</b>	
<b>Condition Variable 1</b>	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
<b>Condition Variable 2</b>	A variable that can be used for condition building. This is an existing field in the YFS_ORDER_LINE database table, and can be used to create conditions without extending the database.
<b>Delivery Method</b>	The delivery method of the order (shipment, pickup or delivery).
<b>Disposition Code</b>	The disposition code of the item. This field is only applicable for Reverse Logistics and Supply Collaboration.
<b>Line Type</b>	The type of the order line. Sterling Selling and Fulfillment Foundation has no application logic associated with the order line type. This field can be set up as per your business practices.
<b>Order Type</b>	The type of the order. Sterling Selling and Fulfillment Foundation has no application logic associated with the order type. This field can be set up as per your business practices.
<b>Payment Status</b>	The payment status of the order.
<b>Sale Voided</b>	The flag indicating whether the order is voided.
<b>Transaction ID</b>	The ID of the last transaction that was run on the order.
<b>Participant Attributes</b>	
<b>Bill To ID</b>	The ID of the bill to address for the order.
<b>Buyer Organization Code</b>	The code of the organization that is buying the goods or services.
<b>Enterprise Code</b>	The code of the enterprise on the order.
<b>Receiving Node</b>	The node that receives the shipment for the order.
<b>Seller Organization Code</b>	The code of the organization that is selling the goods or services.
<b>Ship Node</b>	The node that ships the shipment for the order.
<b>Ship Node Interface Type</b>	The interface type of the ship node on the order (External Application, Console, Sterling Store Inventory Management, or WMS 6.2).
<b>Ship To ID</b>	The ID of the ship to address for the order.
<b>Supplier Code</b>	The code of the supplier for the order.
<b>Item Attributes</b>	
<b>Item ID</b>	The ID of the item on the order line.

Table 341. Master Order Fulfillment Condition Builder Attributes (continued)

Attribute	Description
<b>Item Group Code</b>	The group code of the service item. For example, if the service is a provided service item, then the item group code is PS.
<b>Product Line</b>	The product line of the item on the order line.
<b>Sourcing Attributes</b>	
<b>Fulfillment Type</b>	The fulfillment type of the order.
<b>Intentional Backorder</b>	The flag indicating whether the order was intentionally dropped into backordered status at order creation.
<b>Is Firm Predefined Node</b>	The flag indicating whether the node on the order is a firm predefined node.
<b>Order Sourcing Classification</b>	The order sourcing classification of the order.
<b>Reservation Mandatory</b>	The flag indicating whether the reservation is mandatory.
<b>Related Master Order Attributes</b>	
<b>Chain Type</b>	The chain type of the order.
<b>Is Chained Line</b>	The flag indicating whether the order line is chained with another order line.
<b>Is Derived Line</b>	The flag indicating whether the order line is derived from another order line.
<b>Order Purpose</b>	The purpose of the order. If this is an exchange order, this field is set to EXCHANGE.
<b>{Enter Your Own Attribute}</b>	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

---

## Quote

### Quote Fulfillment

The Quote Fulfillment condition builder attributes are identical to the Order Fulfillment condition builder attributes.

---

## Load Execution

Table 342. Load Execution Condition Builder Attributes

Attribute	Description
<b>Load Type</b>	The type of the load document.
<b>Enterprise Code</b>	The code of the enterprise on the load document.
<b>Owner Organization Code</b>	The code of the organization that owns the load document.
<b>Carrier</b>	The carrier used to carry the load.
<b>Carrier Service Code</b>	The code of the carrier service used to carry the load.

Table 342. Load Execution Condition Builder Attributes (continued)

Attribute	Description
<b>Ship Mode</b>	The shipment mode that is used for the shipment. For example, Parcel, Truck Load, Less-Than Truck Load.
<b>Hazardous Material</b>	The flag indicating whether hazardous materials are being carried in this load.
<b>Origin Node</b>	The node where the load originated from.
<b>Destination Node</b>	The node where the load is being shipped to.
<b>Multiple Load Stop</b>	The flag indicating whether or not a shipment goes through multiple stops to load or unload additional shipments.
<b>{Enter Your Own Attribute}</b>	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## General

The following Condition Builder attributes are identical to those for WMS Putaway, WMS Layout Definition, WMS Inventory, Trailer Loading, Task Execution, Move Request Execution, Manifesting, and Over Pack Build.

Table 343. General Condition Builder Attributes

Attribute	Description
<b>Enterprise Code</b>	The code of the enterprise.
<b>Organization Code</b>	The code of the organization.
<b>Provider Organization Code</b>	The code of the organization that provides the service.
<b>Ship Node</b>	The node that ships this shipment.
<b>Supply Type</b>	The supply type associated with the inventory status. Typical values are Onhand, Held, etc.
<b>Item ID</b>	The ID of the item on the order line.
<b>Unit Of Measure</b>	The unit of measure of the item.
<b>Product Class</b>	The inventory classification of an item based on the product's characteristics. Typical values are FQ - First Quality, SQ - Second Quality, etc.
<b>Inventory Status</b>	The inventory sub classification of the product, based on the results of the inventory control processes within the warehouse. Typical values are Good - Good Inventory, Damaged - Damaged inventory, Qlty-Hold - Quality Hold, etc.
<b>Adjustment Type</b>	The type of inventory adjustment. Typical values are Cycle Count, Receipt, Picking, Packing, Shipping, etc.
<b>Alert Type</b>	The type of alert raised when an exception occurs.
<b>Carrier</b>	The carrier used to carry the shipment.

Table 343. General Condition Builder Attributes (continued)

Attribute	Description
Task Type	The Task Type applicable to a task. Typical values are Receipt, QC, Count, Replenishment, Retrieval, Putaway, VAS, Pack, Shipping, and Picking.
Assigned To User ID	The ID of the user to whom the task is assigned.
Task Status	The Task Status within the pipeline that the task travels through. Typical values are Open, Suggested, In Progress, Held, Completed, Canceled, etc.
Document Type	The document type for this order. Typical values are Sales Order, Purchase Order, Transfer Order, and Return Order.
SC UI Client Version	The Rich Client Platform application version number.
Activity Group ID	The identifier for the activity group.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

---

## WMS Putaway

The WMS Putaway condition builder attributes are identical to the General attributes.

---

## WMS Layout Definition

The WMS Layout Definition condition builder attributes are identical to the General attributes.

---

## WMS Inventory

The WMS Layout Inventory condition builder attributes are identical to the General attributes.

---

## Trailer Loading

The Trailer Loading condition builder attributes are identical to the General attributes.

---

## Task Execution

The Task Execution condition builder attributes are identical to the General attributes.

---

## Move Request Execution

The Move Request Execution condition builder attributes are identical to the General attributes.

---

## Manifesting

The Manifesting condition builder attributes are identical to the General attributes.

---

## Over Pack Build

The Over Pack Build condition builder attributes are identical to the General attributes.

---

## Count Execution

*Table 344. Count Execution Condition Builder Attributes*

<b>Attribute</b>	<b>Description</b>
<b>Enterprise Code</b>	The code of the enterprise for which the count request is created.
<b>Request Type</b>	The type of count requested.
<b>Count Program Name</b>	The name of the count program for which the count request is created.
<b>Node Key</b>	The node where the count request is processed.
<b>Zone ID</b>	The zone where the count must be performed.
<b>Location Size Code</b>	The capacity of the location where the count must be performed.
<b>Is LPN Level</b>	The flag indicating whether the count tasks are be performed at the LPN level.
<b>Is Case Level</b>	The flag indicating whether the count tasks are be performed at the case level.
<b>Is Pallet Level</b>	The flag indicating whether the count tasks are be performed at the pallet level.
<b>Is Item Level</b>	The flag indicating whether the count tasks are be performed at the item level.
<b>Is Resolvable</b>	The flag indicating whether variance can be resolved for this count result.
<b>Product Class</b>	The inventory classification of an item based on the product's characteristics. Typical values are FQ - First Quality, SQ - Second Quality, etc.
<b>Unit Of Measure</b>	The unit of measure of the item that was counted.
<b>Item Classification 1</b>	The first item classification attribute for determining the Count Strategy.
<b>Item Classification 2</b>	The second item classification attribute for determining the Count Strategy.
<b>Item Classification 3</b>	The third item classification attribute for determining the Count Strategy.
<b>Has Variance</b>	The flag indicating whether the count request has a variance.

Table 344. Count Execution Condition Builder Attributes (continued)

Attribute	Description
Has Absolute Variance	The flag indicating whether the count request has an absolute variance.
Variance Quantity	The difference in quantity (+/-) between the count result and system quantity.
Absolute Variance Quantity	The absolute difference between the count result and system quantity.
Variance Value	The difference in cost/value (+/-) between the count result and system quantity.
Absolute Variance Value	The absolute difference in cost/value between the count result and system quantity.
Has Variance With Previous Count	The flag indicating whether the variance between the current count result and previous count results displays.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## Pack Process

Table 345. Pack Process Condition Builder Attributes

Attribute	Description
<b>Node Attributes</b>	
Ship Node	The node that ships this shipment.
Receiving Node	The node that receives this shipment.
Ship from Ship Node Interface Type	The interface type of the ship node from which the shipment is shipped (External Application, Console, Sterling Store Inventory Management, or WMS 6.2).
Ship from Supplier Code	The code of the supplier that is shipping the shipment.
Ship from DCM Integration Real Time	The flag indicating whether the node from which the shipment is shipped uses WMS 6.2.
Ship from Country/Region	The code of the country or region from which the shipment is being shipped.
Ship to Ship Node Interface Type	The interface type of the ship node to which the shipment is shipped (External Application, Console, Sterling Store Inventory Management, or WMS 6.2).
Ship to Supplier Code	The code of the supplier to whom the shipment is being shipped.
Ship to DCM Integration Real Time	The flag indicating whether the node to which the shipment is shipped uses WMS 6.2.
Ship to Country/Region	The code of the country or region to which the shipment is being shipped.
<b>Organization Attributes</b>	
Enterprise Code	The code of the enterprise that owns the shipment.

Table 345. Pack Process Condition Builder Attributes (continued)

Attribute	Description
<b>Buyer Organization Code</b>	The code of the organization that is buying the goods or services.
<b>Seller Organization Code</b>	The code of the organization that is selling the goods or services.
<b>Shipment Attributes</b>	
<b>Ship Mode</b>	The shipment mode that is used for the shipment. For example, Parcel, Truck Load, Less-Than Truck Load.
<b>Carrier</b>	The carrier used to carry the shipment.
<b>Freight Terms</b>	The freight terms of the shipment.
<b>Delivery Code</b>	The code of the entity that pays for the transportation costs.
<b>Pack And Hold</b>	The flag indicating whether the shipment needs to be packed and put away for retrieval at a later date.
<b>Shipment Container Count</b>	The number of containers in the shipment.
<b>Shipment Containerized Flag</b>	The flag indicating the containerization state of the shipment. The values are: 01 - not containerized, 02 - containerization in progress and 03 - containerization completed.
<b>Container Attributes</b>	
<b>Is Shipment Container</b>	The flag indicating whether the container belongs to a shipment.
<b>Is Load Container</b>	The flag indicating whether the container is part of a load.
<b>Is Inventory Pallet</b>	The flag indicating whether the container is an inventory pallet.
<b>Is Converted From LPN</b>	The flag indicating whether the inventory container has been converted to a shipment container.
<b>Is Serial Capture Pending</b>	The flag indicating whether the serial capture is pending for the container.
<b>Is Pack Process Complete</b>	The flag indicating whether any more pack activities are pending for the container.
<b>Is Product Placing Complete</b>	The flag indicating whether placing the product into the container according to the system's suggestion has been completed.
<b>Requires VAS</b>	The flag indicating whether the container requires value added services.
<b>Has Child Containers</b>	The flag indicating whether a container is a parent container having other containers.
<b>Number of Items</b>	The number of items contained in the container.
<b>Container Type</b>	The attribute that specifies whether a shipment container is a case or pallet.
<b>{Enter Your Own Attribute}</b>	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>



## Outbound Picking

Table 346. Outbound Picking Condition Builder Attributes

Attribute	Description
Activity Group ID	The identifier for the activity group.
Shipment Group ID	The identifier for the shipment group.
{Enter Your Own Attribute}	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## VAS Process

Table 347. VAS Process Condition Builder Attributes

Attribute	Description
Enterprise Code	The code of the enterprise that owns the item or license plate.
Provider Organization Code	The code of the organization that provides the service.
Node Key	The node, where the work orders are run.
Purpose	The purpose for the work order (ORDER / STOCK / SHIP)
Service Item Group Code	The code of the service item group (KIT/DKIT/COMPL/INVC/PS)
Service Item ID	The identifier for the service item.
Segment Type	The type of segment. This may be MTO (made to order) or MTC (made to customer).
Segment	The segment to which the inventory involved in the work order belongs.
Has Components	The flag indicating whether the work order has component items.
Status	The status of the work order.
Pre Call Status	The flag indicating the status of the pre-call process.
Appt Status	The status of the appointment. This is in sync with the service order line. The appointment status is used in case of provided service work order.
Number Of Attempts	The number of attempts made to run the work order.
Number Of Hours until Appointment	The number of hours left before the appointment for the service item.
Number Of Hours After Appointment	The number of hours after the last appointment for the service item.
Number Of Hours After Last Execution	The number of hours after the last attempt to run the service.
Last Execution Success	The flag indicating whether the last attempt to run the service was successful or not.

Table 347. VAS Process Condition Builder Attributes (continued)

Attribute	Description
<b>Open Work Order Flag</b>	The flag indicating whether the execution of the work order has ended or not.
<b>{Enter Your Own Attribute}</b>	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

## Opportunity

### Opportunity Fulfillment

Table 348. Opportunity Fulfillment Condition Builder Attributes

Attribute	Description
<b>Opportunity Attributes</b>	
<b>Opportunity ID</b>	The ID of the opportunity.
<b>Opportunity Name</b>	The name of the opportunity.
<b>Status</b>	The status of the opportunity.
<b>Currency Value</b>	The currency value of the opportunity.
<b>Probable Success Rate</b>	The likelihood of whether an order will be created from the opportunity.
<b>Participant Attributes</b>	
<b>Bill To ID</b>	The ID of the bill to address for the opportunity.
<b>Buyer Organization Code</b>	The code of the organization that may buy the goods or services.
<b>Enterprise Code</b>	The code of the enterprise for the opportunity.
<b>Owner User ID</b>	The user ID of the opportunity owner.
<b>Co-Owner User ID</b>	The user ID of the opportunity co-owner.
<b>Customer Contact ID</b>	The ID of the customer contact for the opportunity.
<b>Team Code</b>	The code of the team that manages the opportunity.
<b>{Enter Your Own Attribute}</b>	<p>A customizable condition builder attribute. For more information about customizing this field, see the <i>Sterling Selling and Fulfillment Foundation: Extending the Condition Builder</i>.</p> <p>This field is limited only to unexposed key attributes that are predefined by Sterling Selling and Fulfillment Foundation as opposed to any XML attribute that you can enter.</p>

---

## Item Based Allocation

The Item Based Allocation transaction allocates unpromised and promised demands of existing orders to more suitable supplies based upon inventory items and nodes which have been triggered for the Item Based Allocation process in the YFS\_IBA\_TRIGGER table.

The Item Based Allocation agent obtains and processes all Item Based Allocation triggers from the YFS\_IBA\_TRIGGER table that meet the following conditions:

- IBA\_RUN\_REQUIRED = "Y"
- LAST\_IBA\_PROCESSED\_TS was 'x' hours before current time, where 'x' is from the 'Item Based Allocation Agent Execution Interval (in hours)' rule in the Installation rules. For more information about installation rules, refer to the topic "System Administration Components: Defining Installation Rules" in the *Sterling Selling and Fulfillment Foundation: Configuration Guide*. This rule is used to indicate the interval that the Item Based Allocation agent should not reprocess the triggers in the YFS\_IBA\_TRIGGER table, which were processed earlier. This prevents the IBA agent from over-processing the item and node combination in the given time interval to avoid any high loads on the system.
- PROCESSING\_BY\_AGENT="N" or PROCESS\_OVER\_BY\_TS is before the current timestamp. The PROCESSING\_BY\_AGENT field is used to prevent the picking up of the IBA trigger which is being processed by another instance of the agent.

If InventoryOrganizationCode is specified in the agent criteria, only the IBA trigger with inventory items of that inventory organization is retrieved.

For each triggered item and node combination, the agent finds all of the applicable order lines or order line reservations that contain the item and node and tries to move their unpromised and promised demands to more suitable available supplies based on user-configured IBA selection rules or FIFO (First-In-First-Out) IBA selection rules.

Sterling Selling and Fulfillment Foundation creates new positive order line reservations with the matched supply's first ship date and negative order line reservations for the existing demand ship date. Once all orders are processed, they are placed on hold to be rescheduled if changes are detected in the order line reservations.

The following configuration is required for the Item Based Allocation process:

- The Use Item Based Allocation rule needs to be enabled.
- Item and node need to have Item Based Allocation Allowed enabled.
- A hold type is required to be set up for the change order line reservations modification type so that the order can be placed on hold for rescheduling. For more information, refer to the *Sterling Selling and Fulfillment Foundation: Javadocs*.

The 'When a line is backordered, backorder against the highest priority ship node' rule should be checked in order to reallocate backordered demand. For more information, see the Fulfillment Rules section in the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

Before processing the Item Based Allocation logic, the Item Based Allocation agent updates the following fields on the Item Based Allocation trigger:

- PROCESSING\_BY\_AGENT = "Y". This indicates that an instance of the agent is currently processing this trigger.

- PROCESS\_OVER\_BY\_TS = current time + 1 hr. This indicates the expected time that the agent should finish with processing this IBA trigger. One hour is the fixed window and cannot be changed. Sterling Selling and Fulfillment Foundation treats the PROCESSING\_BY\_AGENT flag as "N" regardless of the actual value when current timestamp is after this timestamp.
- IBA\_RUN\_REQUIRED = "N". This resets the IBA\_RUN\_REQUIRED flag back to "N".

## **Obtaining a List of Demands Based on Applicable Order Release Statuses and Order Line Reservations to be Allocated**

A list of demands is derived from applicable order release statuses and order line reservations, which have the item and node in the IBA trigger. The following types of demands are retrieved:

- Demands of chained orders
- Demands of orders with chained order already created
- Demands of orders with procurement node but chained order creation is not yet created
- Demands of orders without procurement node
- Demands from order line reservations

The demand quantity is derived based on the order release status quantity with the status from the Status Inventory Type configuration that has a demand type, which considers the supply type with 'Use Consider Demand Type for Item Based Allocation' enabled. For more information, refer to the *Sterling Selling and Fulfillment Foundation: Global Inventory Visibility Configuration Guide*.

## **Obtaining a List of Available Supplies for Allocation**

Sterling Selling and Fulfillment Foundation obtains the available supply based on the availability of the item at the node by ignoring unpromised and promised demands. If the inventory organization maintains its inventory externally, the external availability can be read by the YFSGetExternalInventoryUE user exit. Only the availability of supplies that consider the 'Demand Type Look for Availability during Item Based Allocation' are used in the allocation logic. For more information, refer to the *Sterling Selling and Fulfillment Foundation: Global Inventory Visibility Configuration Guide*.

Allocated demands should be matched with the same supplies as "Demand to look for during release".

## **Matching Demands Against Supplies in FIFO (First-In-First-Out) Order**

Sterling Selling and Fulfillment Foundation sorts the list of available supplies in the order of the first shippable date (ETA), and matches the obtained list of demands using the top-down logic (unlike the normal matching logic for obtaining availability, where matches are based on the closest ETA). Demands are allocated in the following orders:

- Demands of chained orders - first based on user-configured sequencing rules, and then in ascending order of order creation date. (These types of demands are matched based on the closest ETA to avoid any changes in the chained orders).

- Demands of orders with a chained order already created - first based on user-configured sequencing rules, then in ascending order of product availability date. (These types of demands are matched based on the closest ETA to avoid any changes in the orders).
- Demands of orders for which procurement node and chained order creation is imminent (within the advanced notification time window) - first based on user-configured sequencing rules, then in order of order creation date.
- Demands of orders without a procurement node and within the release window (advanced notification time window) - first based on user-configured sequencing rules, then in order of order creation date.
- Demands from order line reservations on the order lines in the order of requested reservation date, and left-over demands (outside of the advanced notification time window) of orders with or without a procurement node, first based on user-configured sequencing rules and then in the order of order creation date.
- Demands from inventory reservations in the order of ship date.

Notice that different types of demands are given different priorities based on their significance. The demands of chained orders or orders related to chained orders are treated with a higher priority than the demands of normal orders. Furthermore, the demands with a ship date within the advanced notification time window also have a higher priority than the demands with a date outside of the advanced notification time window.

## Updating Order Reservations for the Matched Demands

After matching the available supply and demand in user-configured sequencing and then in FIFO order, the system builds up a list of order line reservation changes and inventory demand changes (corresponding to the order line reservation changes) and summarize them to optimize the number of order reservation updates and inventory updates. Negative order line reservations are added for the matched demands. Positive order reservations are added for the matched demands with the product availability date set to the matched supplies' first ship date.

After the Item Based Allocation agent completes its tasks for an Item Based Allocation trigger, it updates the fields of the trigger with the following values:

- IBA\_REQUIRED = "N"
- LAST\_IBA\_PROCESSED\_TS = current timestamp.
- PROCESS\_OVER\_BY\_TS = current timestamp.
- PROCESSING\_BY\_AGENT = "N"

The Item Based Allocation agent should be used in conjunction with the rescheduling process as the rescheduling process reschedules the affected orders by utilizing the order line reservations created by the Item Based Allocation process.

## Attributes

The following are the attributes for this time-triggered transaction:

*Table 349. Item Based Allocation Attributes*

Attribute	Value
Base Transaction ID	ITEM_BASED_ALLOCATION

Table 349. Item Based Allocation Attributes (continued)

Attribute	Value
Base Document Type	General
Base Process Type	General
Abstract Transaction	No
APIs Called	changeOrder – for updating the order line reservations created as part of the Item Based Allocation process.
User Exits Called	None

## Criteria Parameters

The following are the criteria parameters for this transaction:

Table 350. Item Based Allocation Criteria Parameters

Parameter	Description
Action	Required. Triggers the transaction. If left blank, it defaults to Get, the only valid value.
Number of Records To Buffer	Optional. Number of records to retrieve and process at one time. If left blank or specified as 0 (zero), it defaults to 5000.
InventoryOrganization Code	The inventory organization code of the inventory items which are processed by the Item Based Allocation agent. If provided, only the IBA triggers with the inventory item that belongs to this inventory organization are processed.
ColonyID	Required in a multischema deployment where the YFS_IBA_TRIGGER table may exist in multiple schemas. Runs the agent for the colony.

## Statistics Tracked

The following statistics are tracked for this transaction:

Table 351. Item Based Allocation Statistics

Statistic Name	Description
NumOrdersProcessed	Number of orders processed by the Item Based Allocation agent.
NumOrdersRequiredReschedule	Number of orders required rescheduling as the result of Item Based Allocation process.

## Pending Job Count

None.

## Events Raised

This transaction raises events as specified under the changeOrder API in the *Sterling Selling and Fulfillment Foundation: Javadocs*.

---

# Index

## A

- actions 39, 44
  - creation 44
- activity codes
  - transportation
    - creating 31
    - deleting 32
    - modifying 31
- Allow Addition of Shipments not Available on System to a Load field 29
- application rules side panel 8
- Applications Manager
  - actions 18
    - document types 19
    - entering dates/times 21
    - lists 21
    - lookup functionality 19
    - special characters 22
    - troubleshooting 22
    - users 20
  - layout 7
  - starting 7
  - work area 14

## B

- base repositories 39
- Break Bulk Node field 82
- building
  - catalog index 109
- business models 1
- buyers 2

## C

- Carrier Service Code field 23, 81
- carrier service codes
  - creating 23
  - deleting 24
  - modifying 24
- Carrier Service Description field 23
- carrier special service codes
  - creating 27
  - defining 27
  - deleting 28
  - modifying 28
- Carrier Special Services Code field 27
- Carrier Special Services Description field 27
- Carrier/Service field 82
- catalog
  - index building 109
- charge categories
  - adding charge names 47
  - creating 47
  - deleting 49
  - deleting charge names 48
  - modifying 48
  - modifying charge names 48
- charge definitions 47

- common codes 45
- condition builder 282
- conditions 39, 43
  - creating 43
- configuration screens
  - accessing 9
- Consider Buyer's Routing Guide field 76
- Consolidate upto volume threshold of field 78
- Consolidate upto weight threshold of field 78
- Consolidator field 81
- Country/Region field 81
- Customer PO # field 77

## D

- Delay shipment by not more than field 78
- Delivery plans 2
- DELIVERYPLANPRG purge rule 51
- Department Code field 77
- Description field 51
- Destinations 3
- Distance Per Day field 23
- drop statuses 42

## E

- Economic Shipping Parameters 77
- Economic shipping parameters
  - maintained field 77
- enterprises 2
- events 44

## F

- freight terms
  - creating 75
  - defining 75
  - deleting 76
  - modifying 76
- From field 81
- fulfillment 39

## H

- Hold Types
  - shipping
    - defining 73
    - purchase order 73
    - return order 73
    - sales order 73
    - transfer order 73
- Hub 40

## I

- index
  - catalog search 109
- inheritance
  - determining 9
- instruction types 46
  - creating 45
  - deleting 46
  - modifying 45

## L

- load types
  - creating 53
  - deleting 53
  - modifying 53
- LOADPRG purge rule 51
- loads 3
- Log File Name field 52

## M

- Maintained Externally field 78
- Mark For field 77
- marketplaces 2
- modification components 63
  - rules 61
  - types 61, 63
- modification reasons 34
  - creating 33
  - deleting 34
  - modifying 33
- Modification Rules
  - Shipping
    - Purchase Order 73
    - Return Order 73
    - Sales Order 73
    - Transfer Order 73
- multidivisional corporations 1

## N

- negotiation 39
- Not Maintained field 78

## O

- Order # field 77
- organization levels 9
  - rules 11
- organization rules 11
  - loading another organization's rules 14
  - overriding 12
- origins 3
- Override Freight Terms field 82
- Override Ship To field 82

## P

- pickup statuses 42
- pipeline determination 40
- pipelines 39
  - creating 41
  - process type 39
  - purchase order receipts
    - viewing action details 44
    - viewing condition details 43
    - viewing details 40
    - viewing status details 42
    - viewing transaction details 41
- Pricing organization 10, 11
- Priority field 82
- process model
  - shipping
    - purchase order 74
    - return order 74
    - sales order 74
    - transfer order 74
- Process Models
  - Shipping
    - defining 74
- process type details
  - shipping
    - purchase order 73
    - return order 73
    - sales order 73
    - transfer order 73
- Process Type Details
  - Shipping
    - defining 74
- process types 39
- Purge Code field 51
- purge rules 51
  - modifying 51
- purges 51

## R

- receipt 39
- Region Schema For Routing field 78
- repositories 39
- Retention Days field 51
- Rollback Segment field 51
- routing guide lines 80
  - deleting 84
  - modifying 79, 84
- routing guides
  - creating 78
  - deleting 84
  - fields 78

## S

- sellers 2
- services 39
- shipment 39
- shipment planning 77
- Shipments 2
- State field 81
- statuses 39, 42
  - creating 43
- stop types
  - modifying 59
- stops 3

Store# field 81

## T

- third-party logistics models 2
- Third-party logistics models 2
- To field 81
- transactions 39, 41
  - creating 42
- transportation
  - activity codes 31, 32
  - defining 31

## W

Write to Log File field 51

## Z

Zip Code field 81



---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing*

*IBM Corporation*

*North Castle Drive*

*Armonk, NY 10504-1785*

*U.S.A.*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing*

*Legal and Intellectual Property Law*

*IBM Japan Ltd.*

*1623-14, Shimotsuruma, Yamato-shi*

*Kanagawa 242-8502 Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation*

*J46A/G4*

*555 Bailey Avenue*

*San Jose, CA 95141-1003*

*U.S.A.*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2012. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2012.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

#### **Trademarks**

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)<sup>®</sup> are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center<sup>®</sup>, Connect:Direct<sup>®</sup>, Connect:Enterprise<sup>®</sup>, Gentran<sup>®</sup>, Gentran<sup>®</sup>:Basic<sup>®</sup>, Gentran:Control<sup>®</sup>, Gentran:Director<sup>®</sup>, Gentran:Plus<sup>®</sup>, Gentran:Realtime<sup>®</sup>, Gentran:Server<sup>®</sup>, Gentran:Viewpoint<sup>®</sup>, Sterling Commerce<sup>™</sup>, Sterling Information Broker<sup>®</sup>, and Sterling Integrator<sup>®</sup> are trademarks or registered trademarks of Sterling Commerce<sup>™</sup>, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.





Printed in USA