

Sterling Selling and Fulfillment Foundation



Global Inventory Visibility Configuration Guide

Release 92

Sterling Selling and Fulfillment Foundation



Global Inventory Visibility Configuration Guide

Release 92

Note

Before using this information and the product it supports, read the information in "Notices" on page 279.

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 to Global

| | |
|---|----------|
| Inventory Visibility | 1 |
| Business Models | 1 |
| Multidivisional Corporation | 1 |
| Third-Party Logistics | 2 |
| Marketplace | 2 |
| Global Inventory Visibility Configuration | 2 |
| Global Inventory Visibility: Inventory Rules | 3 |
| Global Inventory Visibility: Inventory Types and Considerations | 3 |
| Global Inventory Visibility: Distribution Rules | 3 |
| Global Inventory Visibility: Resource Capacity | 3 |
| Icons Used in This Guide | 3 |

Chapter 2. Navigating the Applications Manager

| | |
|---|----------|
| Manager | 5 |
| Starting the Applications Manager | 5 |
| The Applications Manager Layout | 5 |
| Application Rules Side Panel | 6 |
| Work Area | 12 |
| Actions Available in the Applications Manager | 16 |
| Using the Applications Manager's Lookup Functionality | 17 |
| Viewing the Document Types Associated with an Application | 17 |
| Viewing the User Logged into the Applications Manager | 18 |
| Using Lists and List Filtering | 19 |
| Date and Time Entry | 19 |
| Troubleshooting Errors | 20 |
| Using Special Characters | 20 |

Chapter 3. Configuring Inventory Rules

| | |
|--|----|
| 21 | |
| Defining ATP Rules | 22 |
| Creating an ATP Rule | 23 |
| Modifying an ATP Rule | 25 |
| Deleting an ATP Rule | 26 |
| Setting the Default ATP Rule and Default Lead Days | 26 |
| Defining Monitoring Rules | 26 |
| Event Based ATP Monitor Rules | 26 |
| Action Based ATP Monitor Rules | 29 |
| Onhand Inventory Monitor Rules | 31 |
| Defining Product Classes | 32 |
| Creating a Product Class | 32 |
| Modifying a Product Class | 33 |
| Deleting a Product Class | 33 |
| Defining Inventory Reasons | 33 |
| Creating an Inventory Reason | 33 |
| Modifying an Inventory Reason | 34 |
| Deleting an Inventory Reason | 34 |
| Defining IBA Rules | 34 |
| Setting Up IBA Rules | 34 |
| Creating Condition Details | 35 |
| Modifying Condition Details | 36 |

| | |
|---|----|
| Deleting Condition Details | 36 |
| Creating IBA Sequence Header Details | 36 |
| Modifying IBA Sequence Header Details | 37 |
| Deleting IBA Sequence Header Details | 37 |
| Defining Additional Inventory Rules | 37 |

Chapter 4. Configuring Inventory Types and Considerations

| | |
|--|----|
| 41 | |
| Defining Inventory Considerations | 41 |
| Defining Inventory Supply Types | 41 |
| Creating an Inventory Supply Type | 41 |
| Modifying an Inventory Supply Type | 42 |
| Deleting an Inventory Supply Type | 43 |
| Defining Inventory Demand Types | 43 |
| Creating an Inventory Demand Type | 43 |
| Modifying an Inventory Demand Type | 44 |
| Deleting an Inventory Demand Type | 44 |
| Inventory Availability Safety Factor | 45 |

Chapter 5. Configuring Cost Factor Groups

| | |
|---|----|
| 47 | |
| Defining Inventory Cost Factor Groups | 47 |
| Associating Cost Factors with a Cost Factor Group | 47 |

Chapter 6. Configuring Product Item Specific Distribution Groups

| | |
|--|----|
| 49 | |
| Creating a Distribution Group | 49 |
| Adding Nodes/External Organizations to a Distribution Group | 49 |
| Modifying a Distribution Group's Node/External Organization | 50 |
| Deleting a Distribution Group's Node/External Organization | 50 |
| Adding Advanced Distribution Details to a Distribution Group (For Backward Compatibility Only) | 50 |
| Deleting Advanced Distribution Details | 51 |
| Deleting a Distribution Group | 52 |

Chapter 7. Configuring Inventory Node Type Rules

| | |
|---|----|
| 53 | |
| Creating an Inventory Node Type Rule | 53 |
| Modifying an Inventory Node Type Rule | 53 |
| Deleting an Inventory Node Type Rule | 54 |

Chapter 8. Configuring Resource Capacity

| | |
|--|----|
| 55 | |
| Defining Capacity Rules | 55 |
| Defining Region Usage for Resource Pools | 56 |
| Defining Slot Groups | 56 |
| Creating a Slot Group | 57 |
| Modifying a Slot Group | 58 |
| Deleting a Slot Group | 59 |

| | |
|--|----|
| Defining Resource Pools | 59 |
| Creating and Modifying a Resource Pool | 59 |
| Deleting a Resource Pool | 66 |

Chapter 9. Configuring Value Added Services 67

| | |
|---|----|
| Value Added Services: Defining Activities | 67 |
| Creating an Activity Code | 67 |
| Modifying an Activity Code | 68 |
| Deleting an Activity Code | 68 |
| Defining Work Order Cancellation Reasons | 68 |
| Creating a Work Order Cancellation Reason | 68 |
| Creating a New Work Order Cancellation Reason from an Existing Work Order Cancellation Reason | 69 |
| Modifying a Work Order Cancellation Reason | 69 |
| Deleting a Work Order Cancellation Reason | 69 |
| Defining Allocation Considerations | 70 |
| Creating a Work Order Allocation Consideration | 70 |
| Modifying a Work Order Allocation Consideration | 71 |
| Deleting a Work Order Allocation Consideration | 71 |
| Defining Value-Added Services Modification Groups | 72 |
| Setting Up a Value-Added Services Modification Group | 72 |
| Value-Added Services: Modifying a Modification Group | 76 |
| Value-Added Services: Deleting a Modification Group | 76 |
| Defining Value Added Services Process Type Details | 76 |
| Viewing Value Added Services Process Type Details | 77 |
| Defining the Value Added Services Process Model | 78 |
| Value Added Services Process Model: Pipeline Determination | 78 |
| Value Added Services Process Model: Hub Rule | 78 |
| Value Added Services Process Model: Pipelines | 78 |
| Value Added Services Process Model: Transactions | 79 |
| Value Added Services Process Model: Statuses | 80 |
| Value Added Services Process Model: Conditions | 81 |
| Value Added Services Process Model: Actions | 82 |
| Value Added Services Process Model: Service Definitions | 82 |
| Defining Purge Criteria | 83 |
| Setting Up Purge Criteria | 83 |

Chapter 10. Configuring Count 85

| | |
|---|----|
| Defining Count Program | 85 |
| Creating a Count Program | 85 |
| Modifying a Count Program | 86 |
| Deleting a Count Program | 87 |
| Viewing Region Usage for Count | 87 |
| Defining Corporate Count Request Cancellation Reasons | 87 |
| Creating a Corporate Count Request Cancellation Reason | 87 |
| Creating a New Corporate Count Request Cancellation Reason from an Existing Corporate Count Request Cancellation Reason | 88 |

| | |
|---|----|
| Modifying a Corporate Count Request Cancellation Reason | 88 |
| Deleting a Corporate Count Request Cancellation Reason | 89 |
| Defining Corporate Count Request Purge Criteria | 89 |
| Setting Up Corporate Count Request Purge Criteria | 89 |

Chapter 11. Synchronizing with Node Inventory 91

| | |
|--|----|
| Loading the Inventory Picture from a Node | 91 |
| The LoadInventoryMismatch Service | 92 |
| Configuring the LoadInventoryMismatch Service | 93 |
| Synchronizing the Inventory Tables | 94 |
| The SyncLoadedInventory Service | 94 |
| Configuring the CollectInventoryMismatch Service | 95 |
| Executing the Synchronization Process | 96 |
| Purging the Temporary Table | 96 |

Chapter 12. Time-Triggered Transaction Reference 97

| | |
|---|-----|
| Time-Triggered Transaction Reference | 97 |
| Running Time-Triggered Transactions | 98 |
| Steps to Complete Before Scheduling Time-Triggered Transactions | 98 |
| Configuring Communication Between an Agent and a JMS Server | 98 |
| Prerequisites | 99 |
| Create an Initial Context Factory Code | 99 |
| Define the Transaction Information | 100 |
| Business Process Time-Triggered Transactions | 101 |
| Asynchronous Request Processor | 101 |
| Case Insensitive Data Loader | 102 |
| Change Load Status | 104 |
| Change Shipment Status | 105 |
| Close Delivery Plan | 106 |
| Close Load | 108 |
| Close Manifest | 109 |
| Close Order | 111 |
| Close Receipts | 112 |
| Close Shipment | 114 |
| Collect Shipment Statistics | 115 |
| Consolidate Additional Inventory | 117 |
| Consolidate To Shipment | 118 |
| Create Catalog Index | 120 |
| Create Chained Order | 124 |
| Create Derived Order | 125 |
| Create Order Invoice | 127 |
| Create Shipment Invoice | 128 |
| ESP Evaluator | 129 |
| Item Based Allocation | 131 |
| Mark Load as Trailer Loaded | 135 |
| Match Inventory | 136 |
| Payment Collection | 137 |
| Payment Execution | 139 |
| Post Inventory Match | 141 |
| Process Order Hold Type | 142 |
| Process Work Order Hold Type | 144 |
| Publish Negotiation Results | 145 |

| | |
|---|-----|
| Release | 147 |
| Route Shipment | 149 |
| Schedule | 150 |
| Send Invoice | 153 |
| Send Item Changes | 155 |
| Send Customer Changes. | 156 |
| Send Order | 158 |
| Send Release | 159 |
| Start Order Negotiation | 160 |
| Synchronize Colony Map | 161 |
| Update Best Match Region | 163 |
| PopulateOwnershipTransferSummary | 165 |
| Time-Triggered Purge Transactions | 166 |
| Purge Strategy | 166 |
| Configuring Purge Transaction Log Files | 166 |
| Available Purges | 167 |
| Task Queue Syncher Time-Triggered Transactions | 244 |
| Load Execution Task Queue Syncher | 244 |
| Order Delivery Task Queue Syncher. | 245 |

| | |
|--|-----|
| Order Fulfillment Task Queue Syncher | 246 |
| Order Negotiation Task Queue Syncher | 247 |
| Quote Fulfillment Task Queue Syncher | 248 |
| Monitors | 249 |
| Availability Monitor | 250 |
| Exception Monitor. | 251 |
| Inventory Monitor. | 253 |
| Negotiation Monitor | 254 |
| Enhanced Order Monitor | 255 |
| Enhanced Quote Monitor | 258 |
| Enhanced Return Monitor | 260 |
| Real-time Availability Monitor. | 263 |
| Shipment Monitor. | 270 |
| Work Order Monitor | 272 |

Index 275

Notices 279

Chapter 1. Introduction to Global Inventory Visibility

This book concentrates on the rules and setup configurations that make up the IBM® Sterling Global Inventory Visibility business application in the Applications Manager. This book 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 book to plan appropriate business practices as they pertain to Sterling Selling and Fulfillment Foundation. Programmers should refer to the *Sterling Selling and Fulfillment Foundation: Extending Transactions* 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.

This book assumes that you have read and are 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
- Sterling Global Inventory Visibility
- Catalog Management
- IBM 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 the entire Sterling Selling and Fulfillment Foundation 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.

Global Inventory Visibility Configuration

The Sterling Global Inventory Visibility application is a collection of common components used to define inventory and capacity availability throughout the system.

In Sterling Selling and Fulfillment Foundation, inventory is defined as physical product items against which supply and demand checks can be made. For example, refrigerators and dishwashers are considered to be inventory product items.

Capacity defines a physical location's resource availability to fulfill delivery and provided service items. For example, refrigerator delivery can be considered a delivery service item and dishwasher installation can be considered a provided service item.

In the Applications Manager, you can use the Global Inventory Visibility configuration grouping to establish inventory rules, inventory types, distribution rules, and resource capacity for your business applications.

Global Inventory Visibility: Inventory Rules

Inventory business rules are used to set up rules and common codes associated with inventory handling in Sterling Selling and Fulfillment Foundation, including:

- Available-to-promise (ATP) rules
- ATP monitoring rules
- Product classes
- Inventory reasons

For more information about Inventory Rules, see Chapter 3, “Configuring Inventory Rules,” on page 21.

Global Inventory Visibility: Inventory Types and Considerations

Inventory Types and Considerations is used to identify the supply and demand type associations used to determine inventory availability for a specific demand type. You can also create new supply and demand types for use in Sterling Selling and Fulfillment Foundation. For more information about Inventory Types and Considerations, see “Defining Inventory Considerations” on page 41.

Global Inventory Visibility: Distribution Rules

Distribution Rules is used to create a set of nodes/external organizations that can be used when determining item sourcing. You can define distribution rules that establish the ship node determination process within a distribution group. These rules determine the default node that an item should be sourced from within a group based on priority. You can create rules for individual items at a source node or for the entire source node. For more information about Distribution Rules, see Chapter 6, “Configuring Product Item Specific Distribution Groups,” on page 49.

Global Inventory Visibility: Resource Capacity

Resource Capacity is used to define components that determine delivery service item and provided service item availability. Resource capacity availability is used to determine appointments for a delivery service and/or provided service of a defined unit of measure for specific time slots and geographical regions. For more information about Resource Capacity, see Chapter 8, “Configuring Resource Capacity,” on page 55.

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 inventory and capacity availability. As shown, you may see this icon with different backgrounds, depending on the context within which you are working. |

Table 1. Icons Used in This Guide (continued)

| Icon | Name | Description |
|---|---------------------|---|
|  | Save | Click this icon to save entries you have made. The Save icon may be displayed in blue or gold. |
|  | Details | Click this icon to view or change details for components used to define inventory and capacity availability. As shown, you may see this icon with different backgrounds, depending on the context within which you are working. |
|  | Delete | Click this icon to delete components used to define inventory and capacity availability. As shown, you may see this icon with different backgrounds, depending on the context within which you are working. |
|  | Search | Click this icon to search for components used to define inventory and capacity availability. |
|  | Save As... | Click this icon to create new inventory and capacity availability data from existing data. |
|  | Zoom In | Click this icon to zoom in on your screen. |
|  | Add User | Click this icon to add a user. |
|  | Remove User | Click this icon to remove a user. |
|  | Transactions | Click this icon to view a list of transactions in the Transactions tab window. |
|  | Statuses | Click this icon to view a list of statuses in the Statuses tab window. |
|  | Conditions | Click this icon to view a list of conditions in the Conditions tab window. |
|  | Actions | Click this icon to view a list of actions in the Actions tab window. |
|  | Service Definitions | Click this icon to view a list of service definitions in the Service Definitions tab window. |

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.

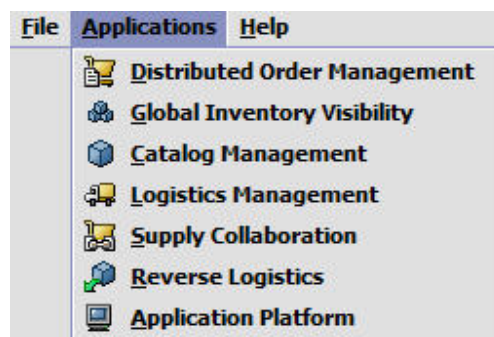


Figure 1. Applications Menu

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.

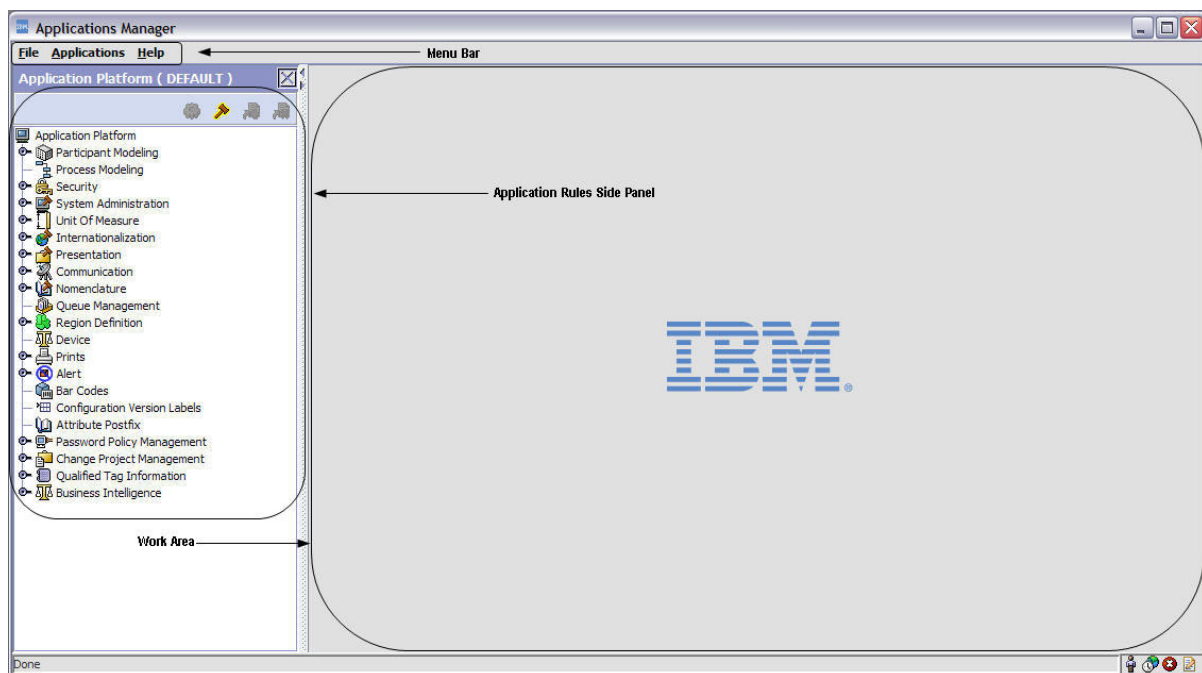


Figure 2. The Standard Applications Manager Interface

Application Rules Side Panel

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

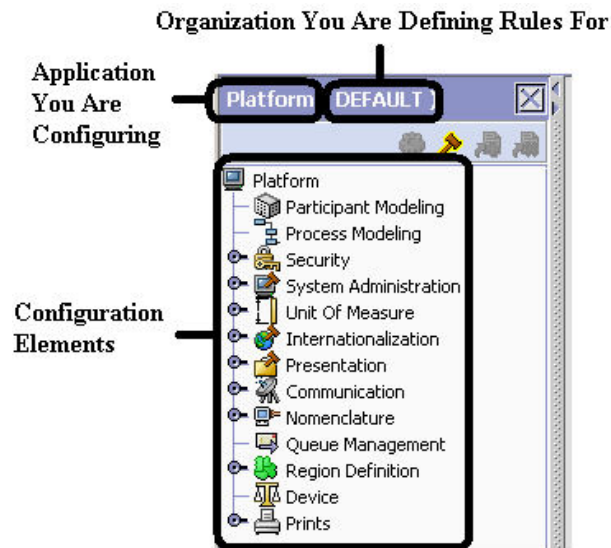


Figure 3. Example of Application Rules Side Panel

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 12.

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*.

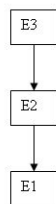
Enhanced Inheritance for Process Models

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.

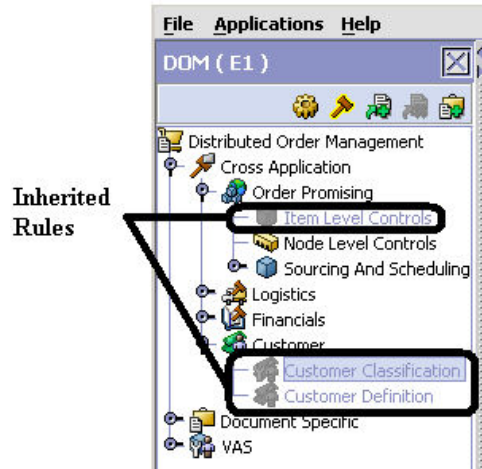


Figure 4. Inherited Rules in the Application Rules Side Panel

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

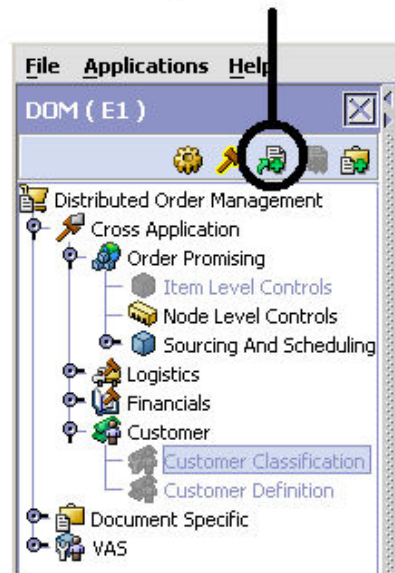


Figure 5. Override Configuration Icon

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.

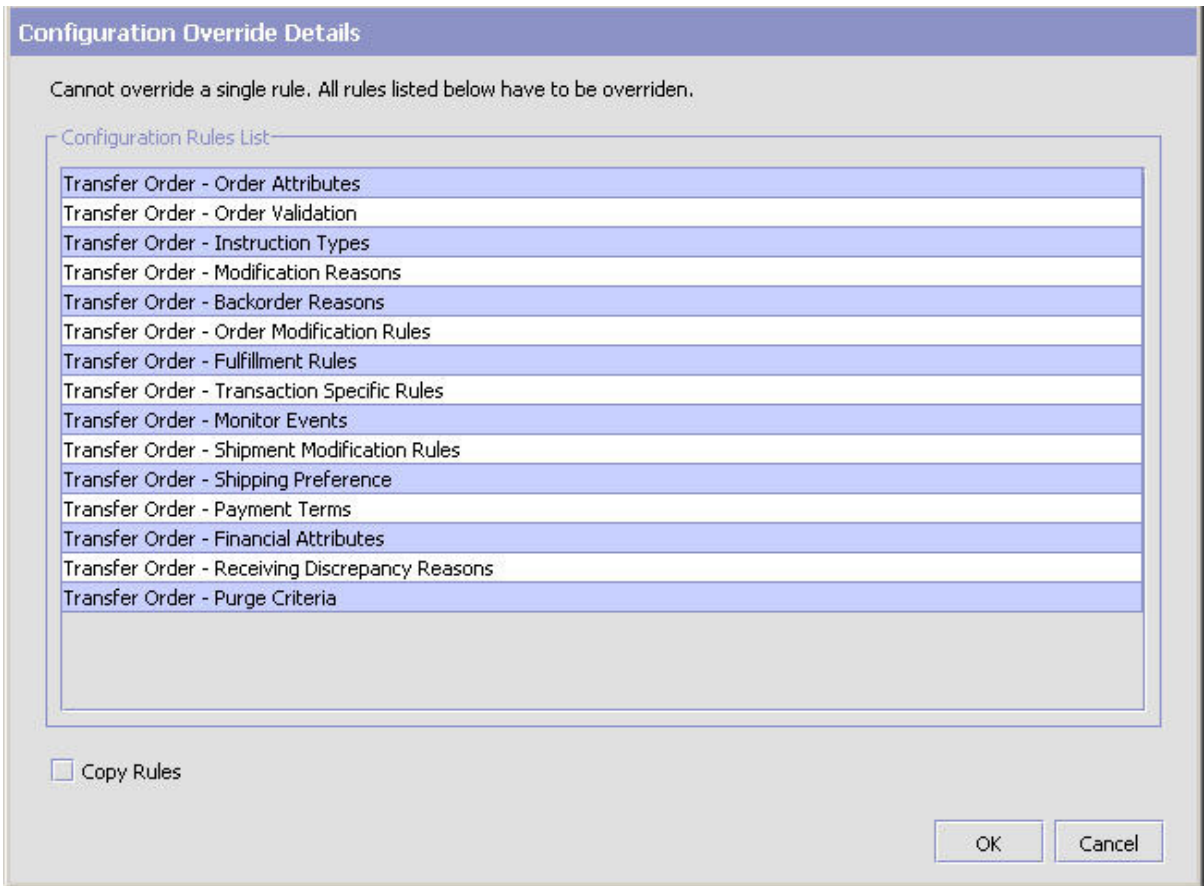


Figure 6. Example of Configuration Override Details Pop-Up Window

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.

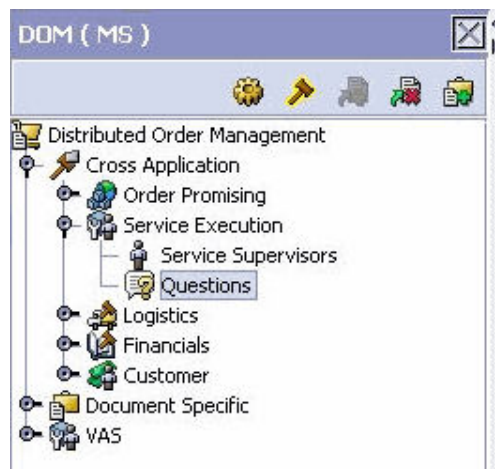


Figure 7. Give Back Configuration Ownership Icon

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 9 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.

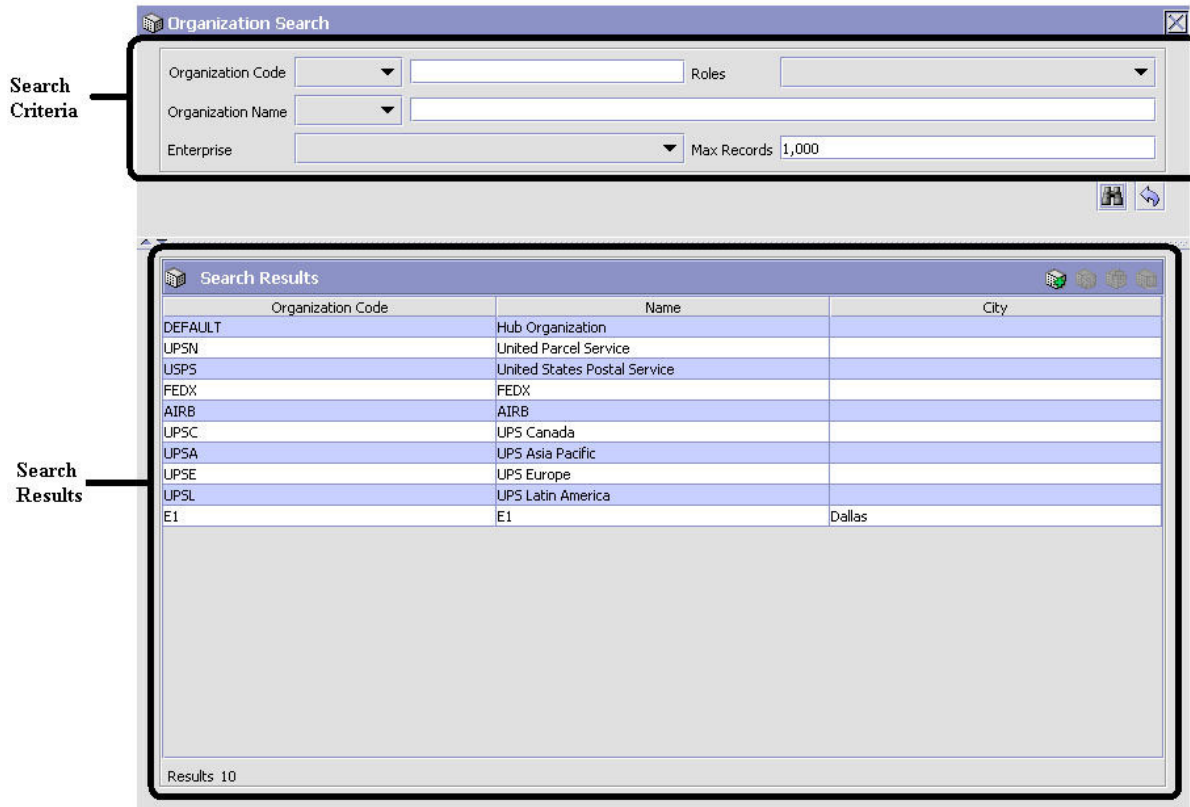


Figure 8. Search Window Example

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.

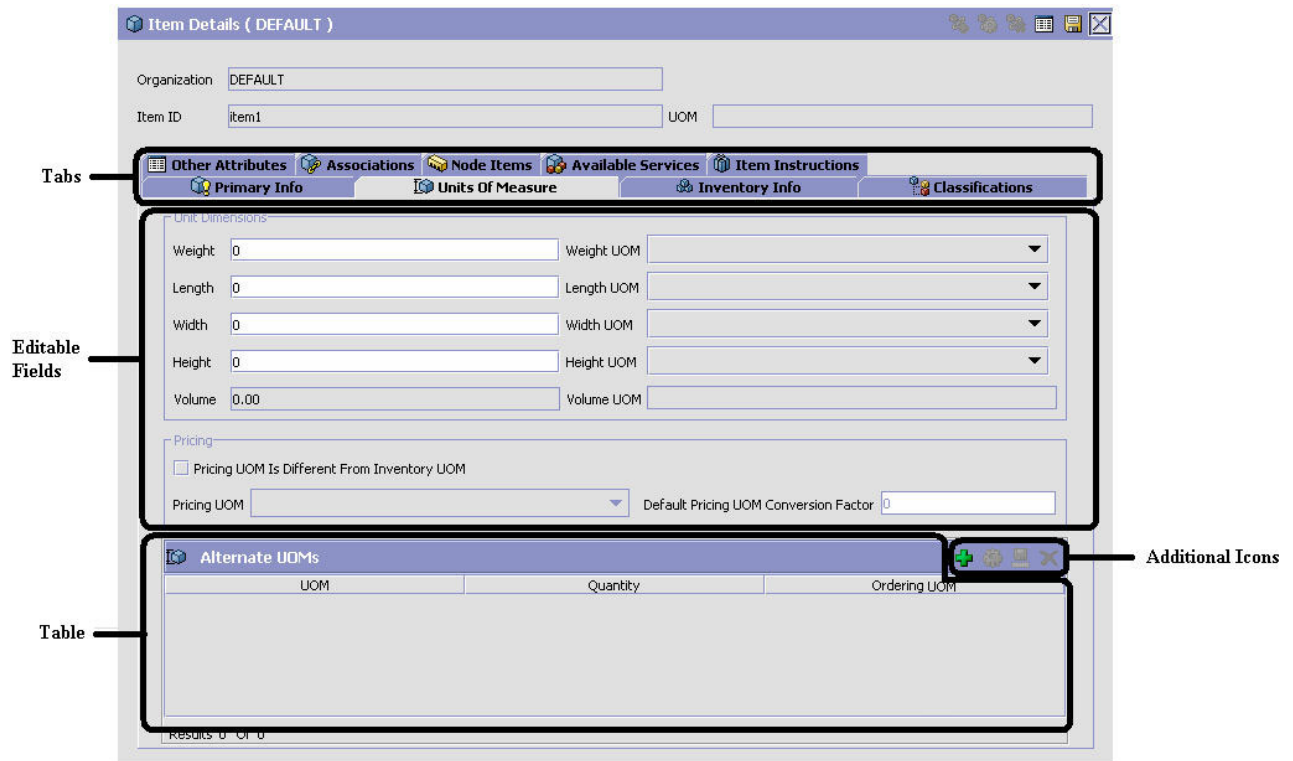


Figure 10. Details Window Example

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.

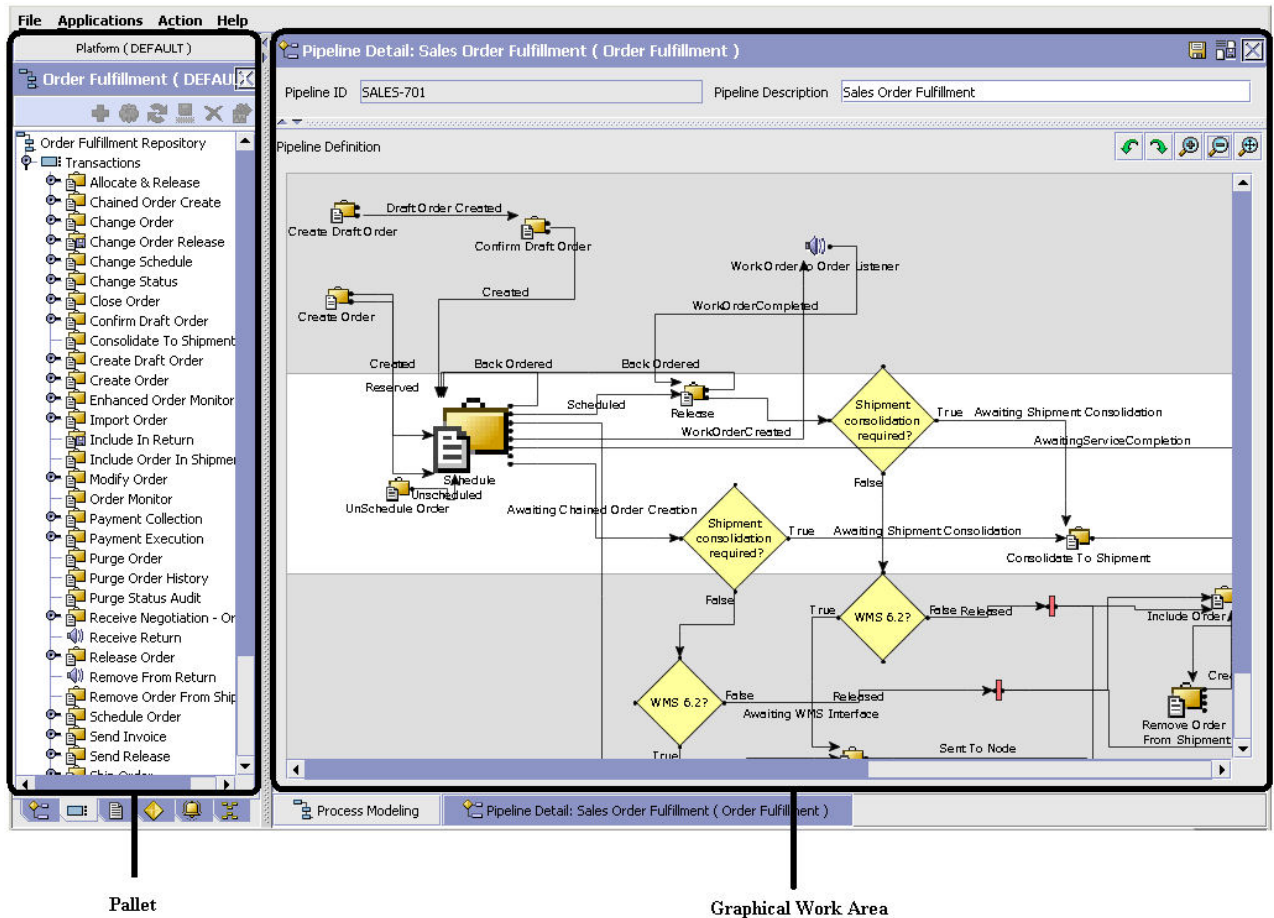


Figure 11. Drag and Drop Window Example

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.

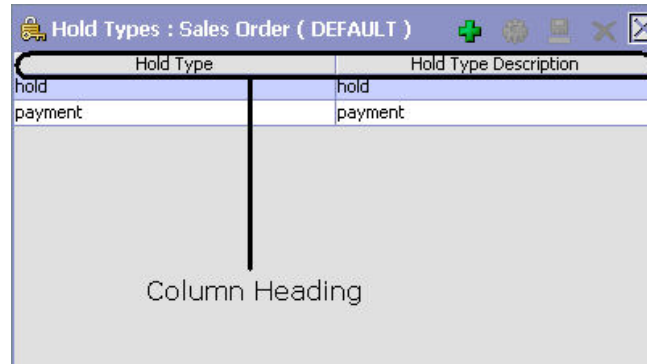


Figure 14. Column Headings in a List

Table 3. Table Filter Editor Window

| Field | Description |
|----------------------------------|---|
| Apply To Existing Records | Checking this box applies a new filter set of results that have been previously filtered instead of the whole set. |
| Max Records | Specify the maximum number of records that are to be returned from a filter. The default number is 100 |
| Dynamic Fields | 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 Is , Starts With , or Contains . |

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 Inventory Rules

Inventory business rules are used to set up rules and common codes used for product item availability calculations and inventory handling in Sterling Selling and Fulfillment Foundation.

You can access the Inventory Rules window by choosing the Inventory Rules branch from the tree in the application rules side panel. The following table describes the tabs and fields in the Inventory Rules window.

Table 4. Inventory Rules Screen, ATP Rules

| Fields | |
|----------------------|--|
| Default ATP Rule | Select the ATP rule you want to default to a catalog item if no ATP rule is specified |
| Lead Days | Enter the lead days you want to default to for an item that is not defined in the catalog. If the item exists in the catalog, the lead days specified for that item will be used. Lead days is the amount of time it takes to procure an item for shipping. |
| ATP Rules | |
| ATP Rule | The ATP rule identifier. |
| ATP Rule Description | The description for the ATP rule. |

Table 5. Inventory Rules Screen, Monitor Rules

| Fields | |
|---|---|
| Use Activity-Based Mode For Real-Time Availability Monitor | Select this checkbox if you want to use the real-time availability monitor in activity-based mode. |
| Distribution Group To Use For Node Level Monitoring. | Select a distribution group to use for node level monitoring from the dropdown, if applicable. You can also click the Configure Monitor Rule Distribution Group icon to select multiple distribution groups to use for node level monitoring, as well as multiple distribution groups to use for group-level monitoring. If you select the initial dropdown as well as choices on the Monitor Rule Distribution Groups screen, the initial dropdown choice will be respected. |
| Availability Monitor/Inventory Monitor Reglog Interval (Hours) | Enter the availability monitor/inventory monitor reglog interval, in hours. |
| Compute Availability Information with Ship Dates for Real-Time Availability Monitor | Select this checkbox if you want the real-time availability monitor to compute and publish a matrix of available quantities and corresponding maximum ship dates. |
| Maximum Ship Date Time (HH:MM) | Enter the time of day for the maximum ship date, in hours and minutes. |
| Number of Days To Offset the Maximum Ship Date | Enter the number of days by which the maximum ship date is offset. |

Table 5. Inventory Rules Screen, Monitor Rules (continued)

| | |
|---|--|
| Default Inventory Monitor Rule | Select the default inventory monitor rule to be used for any item that is not available on the system. Setting this rule ensures that activity will be recorded when the system is running in activity-based mode and that inventory is monitored across all nodes in the associated distribution group. The distribution group and other rule details are defined in the ATP Monitor Rules on this screen. |
| Default Node Level Inventory Monitor Rule | Select the default node level inventory monitor rule to be used for any item that is not available on the system. Setting this rule ensures that activity will be recorded when the system is running in activity-based mode and that inventory is monitored at the node level for all nodes defined in the distribution group. The distribution group and other rule details are defined in the ATP Monitor Rules on this screen. |
| ATP Monitor Rules | |
| ATP Monitor Rule | The ATP monitor rule identifier. |
| ATP Monitor Rule Name | The ATP monitor rule name. |
| Disabled | This field indicates whether the ATP monitor rule is disabled. |
| Onhand Inventory Monitor Rules | |
| Item ID | The item for this monitor rule. |
| Node | The node on this monitor rule. |

Table 6. Inventory Rules Screen, Product Classes

| | |
|-------------------|---|
| Fields | |
| Product Class | The product class. |
| Short Description | The short description for this product class. |

Table 7. Inventory Rules Screen, Inventory Reasons

| | |
|-------------------|--|
| Fields | |
| Inventory Reason | The inventory reason. |
| Short Description | The short description for this inventory reason. |

For more information about the Other Rules tab, see “Defining Additional Inventory Rules” on page 37.

Defining ATP Rules

Available-to-promise (ATP) rules enable you to determine the availability of a product item for current and future demand. This determination makes the most efficient use of inventory so that product items are not set aside for future orders when they could be used to fulfill more immediate demands. The availability of a product item is based on current and future supply, lead days, and ATP configuration. The **lead days** is the amount of time it takes a node to procure a product item for shipping. The **processing time** covers the time it takes for a product item to be received by a supplier and made ready for shipment (inbound processing) as well as shipping it from the warehouse (outbound processing).

ATP rules enable you to effectively manage orders for product items. Parameters can be set for the amount of time a product item is available for current and future

orders. With a First Expiration First Out (FEFO) inventory management system, perishable product items can be sold and shipped well before their expiration dates, ensuring first expired inventory is consumed first.

ATP rules enable you to fulfill current and future customer demand as well as more effectively manage warehouse inventory and processing time for product items.

Sterling Selling and Fulfillment Foundation provides a default ATP rule, called DEFAULT, that is used when no other rule has been defined.

Note: Due to the system-wide impact of creating ATP rules, IBM strongly recommends the following:

- If you are configuring ATP rules for the Hub organization, do not alter the factory default ATP rule. Use this rule as a guideline for setting up new ATP rules.
- If you are configuring ATP rules for an Enterprise organization, view the Hub's factory default ATP rule and use it as a guideline when creating new ATP rules.

Creating an ATP Rule

About this task

To create an ATP rule:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the ATP Rules tab.
3. Choose the **Create New** icon. The ATP Rule Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 8 for field value descriptions.
5. Choose the **Save** icon.

Table 8. ATP Rule Details Pop-Up Window

| Field | Description |
|----------------------|---|
| ATP Rule | Enter the ATP rule. |
| ATP Rule Name | Enter the name of the ATP rule. |
| ATP Rule Information | |
| Past Due Supply Days | <p>Enter the number of days after which supply that is not received cannot be considered in ATP calculations.</p> <p>For example, if you do not want to fulfill orders on supplies that are overdue by 2 days, enter 2 as the parameter for Past Due Supply Days.</p> <p>Note: IBM suggests using a value of '730' initially. Evaluate the system's behavior and adjust this value higher or lower as necessary.</p> |

Table 8. ATP Rule Details Pop-Up Window (continued)

| Field | Description |
|-----------------------------|--|
| Past Due Demand Days | <p>Enter the number of days after which demand for a product item is not considered in the ATP calculation.</p> <p>For example, if demand for a product item is delayed for longer than the requested ship date, you can limit the amount of time this demand can use up available inventory.</p> <p>Any demand that is past due for the number of days specified in this parameter is not included in ATP calculations for demand.</p> <p>Note: IBM suggests using a value of '730' initially. Evaluate the system's behavior and adjust this value higher or lower as necessary.</p> |
| Forward Consumption (Days) | <p>Enter the number of days future supply can be consumed to fulfill a current demand.</p> <p>Note: IBM suggests using a value of '730' initially. Evaluate the system's behavior and adjust this value higher or lower as necessary.</p> <p>Note: For tag controlled items, if demands are being matched against future inventory (purchase orders), during scheduling of a current order, the ForwardConsumptionDays is not considered. However, during an existing supply and demand reallocation the ForwardConsumptionDays is considered. This may cause a new order (refer to previous note) to be backordered because reallocation of other existing demand considers incoming supply only up to ForwardConsumptionDays. To avoid this, if possible, set the ForwardConsumptionDays in the ATP rule equal to the value defined in the "Maximum no. of days order can be shipped/delivered beyond its requested date" rule.</p> <p>For more information, see the note in the "Scheduling Rule Details" table for the "Maximum no. of days order can be shipped/delivered beyond its requested date" field in the <i>Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide</i>.</p> |
| Backward Consumption (Days) | <p>Enter the number of days to go back and check for available supply for a current demand.</p> <p>Demands are matched against the supply available on the same day. If there is not enough inventory available to fulfill the order that day, you can allow the system to consume inventory that was available several days back. You can determine how far back the system can check for available inventory before looking forward.</p> <p>Note: IBM suggests using a value of '730' initially. Evaluate the system's behavior and adjust this value higher or lower as necessary.</p> |

Table 8. ATP Rule Details Pop-Up Window (continued)

| Field | Description |
|---|---|
| Processing Time For Future Inventory (Days) | <p>Enter the amount of days it takes for inbound and outbound processing of a product item. This should also include special services such as gift wrapping, kit items, and special delivery.</p> <p>For example, if it takes 1 day to receive roses and store them in the warehouse and another 2 days to pick, pack, and ship them, then the processing time is 3 days.</p> <p>This parameter is used by the Schedule time-triggered transaction to determine if an order can be scheduled against expected supply.</p> |
| Advance Notification Time (Days) | <p>Enter the number of days in advance that a node needs to be notified to prepare an order for shipment.</p> <p>Release date is based on Advanced Notification Time. This is calculated as: <release date> = <requested ship date> - <advanced notification time>. An order is picked up for release if any line has a release date less than or equal to today. All other lines in the order with a ship date less than or equal to the lines satisfying the release date are also scheduled to ensure that they are shipped together.</p> <p>This parameter is used by the Release time-triggered transaction to determine when an order should be released to the ship node. Note: IBM recommends using a value of '0' for this field.</p> |
| Accumulation Time (Days) | <p>Enter the number of days for which a supply can be considered available for future demand.</p> <p>This parameter determines whether a currently available supply of a product item can be considered for future order fulfillment. This parameter must be set to less than or equal to the Backward Consumption Days.</p> <p>For example, if a purchase order becomes 'onhand' on 11/1 and this parameter is set to 10 days, the inventory from the purchase order is considered available until 11/11. Note: IBM suggests using a value of '730' initially. Evaluate the system's behavior and adjust this value higher or lower as necessary.</p> |

Modifying an ATP Rule

About this task

To modify an ATP rule:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the ATP Rules tab.
3. Select the applicable ATP rule and choose the **Details** icon. The ATP Rule Details pop-up window displays.
4. Modify information in the applicable fields. Refer to Table 10 on page 29 for field value descriptions.

5. Choose the **Save** icon.

Deleting an ATP Rule

About this task

To delete an ATP rule:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the ATP Rules tab.
3. Select the applicable ATP rule and choose the **Delete** icon.

Setting the Default ATP Rule and Default Lead Days

About this task

To set the default ATP rule and default lead days:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the ATP Rules tab.
3. From Default ATP Rule, select the ATP rule you want to default to a catalog item if no ATP rule is specified. For more information about setting an ATP rule for a catalog item, see the *Catalog Management: Configuration Guide*.
4. In Lead days, enter the lead days you want to default to a catalog item if no lead days is specified. **Lead days** is the amount of time it takes to procure an item for shipping. For more information about setting the lead days for a catalog item, see the *Catalog Management: Configuration Guide*.
5. Choose the **Save** icon.

Defining Monitoring Rules

ATP Monitoring Rules enable you to define a monitoring system for tracking the inventory availability of an item and raising specific actions or events when the inventory falls below a specified minimum level.

Onhand Inventory Monitor Rules define a monitoring system for tracking inventory changes for a given item and node combination. Each time the inventory availability of an item for a specific node falls below or above a specified level, an action is raised.

You can define three different monitoring rules in this screen:

- Event Based ATP Monitor Rules, which corresponds to the Real-time Availability Monitor
- Action Based ATP Monitor Rules, which corresponds to the Availability Monitor
- Onhand Inventory Monitor Rules, which corresponds to the Inventory Monitor and is action based

Event Based ATP Monitor Rules

The event based ATP monitor rule determines the parameters used to monitor the availability of inventory items. If the available quantity changes between the

configured levels, the Real Time Availability Monitor raises the REALTIME_AVAILABILITY_CHANGE event. The Real Time Availability Monitor is a time-triggered transaction that can be found and configured under the General process type.

For more information about configuring transactions, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*. For more information about the real-time availability monitor, see “Real-time Availability Monitor” on page 263.

Note: If several items utilize the same thresholds, IBM recommends using the same monitoring rule for all of those items.

Creating an Event Based ATP Monitoring Rule Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Monitor Rules tab.
3. If you want to be able to run the real-time availability monitor in activity-based mode, check Use Activity-Based Mode for Real-Time Availability Monitor. If this is checked, it is applicable for all monitoring rules, for all items.
4. If you want to specify a distribution group to use, select a distribution group from the Distribution Group to Use For Node Level Monitoring drop-down menu. If this is checked, it is applicable for all monitoring rules, for all items. If this option is left blank, the organization's default distribution group is used.
5. If you want the real-time availability monitor to compute and publish a matrix of available quantities and corresponding maximum ship dates, check Compute Availability Information with Ship Dates for Real-Time Availability Monitor. If this is checked, you can specify a time for the maximum ship date in the Maximum Ship Date Time field and a number of days to offset the maximum ship date in the Number of Days to Offset the Maximum Ship Date field.
6. Choose the **Create New** icon. The ATP Monitor Rule Details pop-up window displays.
7. Select the Event Based ATP Monitor Rule radio button.
8. Enter information in the applicable fields. Refer to Table 9 for field value descriptions.
9. Choose the **Save** icon.

Note: Once this rule is created, it is necessary to apply this rule to an item for it to take effect. For more information about defining a product's inventory information, see the *Catalog Management: Configuration Guide*.

Table 9. Event Based ATP Monitor Rule Details Pop-Up Window

| Field | Description |
|---------------------------------|--|
| ATP Monitor Rule | Enter the ATP monitor rule. |
| ATP Monitor Rule Name | Enter the name of the ATP monitor rule. |
| Lead Time Override (Days) | By default, inventory items are monitored from current date to current date plus the inventory item's lead days. If this field is set, the inventory items are monitored from current date to current plus the lead time override (Days). |
| Disabled | Check this if you want this monitoring rule to be disabled. |
| Event Triggered Based On | |

Table 9. Event Based ATP Monitor Rule Details Pop-Up Window (continued)

| Field | Description |
|--------------------------|--|
| High Quantity Boundary | Enter the available inventory level above which an inventory item is considered as being in high quantity. |
| Medium Quantity Boundary | Enter the available inventory level above which an inventory item is considered as being in medium quantity. |
| Low Quantity Boundary | Enter the available inventory level above which an inventory item is considered as being in low quantity. |

Modifying an Event Based ATP Monitoring Rule

About this task

To modify an ATP monitoring rule:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Monitor Rules tab.
3. If you want to be able to run the real-time availability monitor in activity-based mode, check Use Activity-Based Mode for Real-Time Availability Monitor. If this is checked, it is applicable for all monitoring rules, for all items.
4. If you want to be able to run the real-time availability monitor at the node level, check Use Real-Time Availability Monitor at Node Level. If this is checked, it is applicable for all monitoring rules, for all items.
5. If you want to specify a distribution group to use, select a distribution group from the Distribution Group to Use drop-down menu. If this is checked, it is applicable for all monitoring rules, for all items.
6. If you want the real-time availability monitor to compute and publish a matrix of available quantities and corresponding maximum ship dates, check Compute Availability Information with Ship Dates for Real-Time Availability Monitor. If this is checked, you can specify a time for the maximum ship date in the Maximum Ship Date Time field and a number of days to offset the maximum ship date in the Number of Days to Offset the Maximum Ship Date field.
7. Select the applicable ATP monitoring rule and choose the **Details** icon. The ATP Monitor Rule Details pop-up window displays.
8. Select the Event Based ATP Monitor Rule radio button.
9. Modify information in the applicable fields. Refer to Table 9 on page 27 for field value descriptions.
10. Click the **Save** icon.

Deleting an Event Based ATP Monitoring Rule

About this task

To delete an ATP monitoring rule:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Monitor Rules tab.

3. Select the Event Based ATP Monitor Rule radio button.
4. Select the applicable ATP monitoring rule and click the **Delete** icon.

Action Based ATP Monitor Rules

The availability of an item can be tracked on the current day, subsequent days within the ATP timeframe, and subsequent days outside the ATP timeframe. This enables you to more accurately order supplies to meet current and future demand.

For more information about configuring transactions, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*. For more information about the availability monitor, see “Availability Monitor” on page 250.

The action based ATP monitor:

- Checks the availability of all product items set up for monitoring.
- Divides the monitoring into lead and post-lead days.
- Establishes and builds an ATP table for availability within and beyond lead days based on planned supplies and actual demand. (Information about available supplies and demand is added until the end of a defined monitoring period.)
- Raises actions for any product items that fall below specified minimum inventory levels. For example, an e-mail can be sent to purchasing or warehouse managers concerning product item shortages so additional supplies can be ordered.

Creating an Action Based ATP Monitoring Rule About this task

To create an action based ATP monitoring rule:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Monitor Rules tab.
3. Click the **Create New** icon. The ATP Monitor Rule Details pop-up window displays.
4. Select the Action Based ATP Monitor Rule radio button.
5. Enter information in the applicable fields. Refer to Table 10 for field value descriptions.
6. Click the **Save** icon.

Table 10. Action Based ATP Monitor Rule Details Pop-Up Window

| Field | Description |
|----------------------------------|--|
| ATP Monitor Rule | Enter the ATP monitor rule. |
| ATP Monitor Rule Name | Enter the name of the ATP monitor rule. |
| Lead Time Override (Days) | By default, inventory items are monitored from current date to current date plus the inventory item's lead days. If this field is set, the inventory items are monitored from current date to current plus the lead time override (Days). |
| Disabled | Check this if you want this monitoring rule to be disabled. |
| Action Triggered Based On | |

Table 10. Action Based ATP Monitor Rule Details Pop-Up Window (continued)

| Field | Description |
|-----------------------------------|--|
| Within Lead Time | Calculated as the following: Today + Lead Days + Processing Time |
| High Minimum Available Quantity | Enter an amount for the minimum available inventory level within lead days. This defines the number of units which should be available every day within the lead days. If availability is less than this, an action is raised. |
| High Minimum Action | Select a predefined action that is raised if the available inventory level falls below the defined minimum for High. |
| Medium Minimum Available Quantity | Enter an amount (that is less than High) for the minimum available inventory level within lead days. If availability is less than this, a higher priority action is raised. |
| Medium Minimum Action | Select a predefined action that is raised if the available inventory level falls below the defined minimum for Medium. |
| Low Minimum Available Quantity | Enter an amount (that is less than Medium) for the minimum available inventory level within lead days. If availability is less than this, a higher priority action is raised. |
| Low Minimum Action | Select a predefined action that is raised if the available inventory level falls below the defined minimum for Low. |
| Outside Lead Time | Calculated from the end of lead days to (Today + Max Monitoring Days). |
| High Minimum Available Quantity | Enter an amount for the minimum available inventory level beyond lead days. This defines the number of units that should be available every day beyond the lead days. If an availability is less than this, an action is raised. |
| High Minimum Action | Select a predefined action that is raised if the available inventory level falls below the defined minimum for High. |
| Medium Minimum Available Quantity | Enter an amount (that is less than High) for the minimum available inventory level beyond lead days. If availability is less than this, an action is raised. |
| Medium Minimum Action | Select a predefined action that is raised if the available inventory level falls below the defined minimum for Medium. |
| Low Minimum Available Quantity | Enter an amount (that is less than Low) for the minimum available inventory level beyond lead days. If availability is less than this, an action is raised. |
| Low Minimum Action | Select a predefined action that is raised if the available inventory level falls below the defined minimum for Low. |

Modifying an ATP Monitoring Rule

About this task

To modify an ATP monitoring rule:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Monitor Rules tab.

3. Select the applicable ATP monitoring rule and click the **Details** icon. The ATP Monitor Rule Details pop-up window displays.
4. Modify information in the applicable fields. Refer to Table 10 on page 29 for field value descriptions.
5. Click the **Save** icon.

Deleting an ATP Monitoring Rule

About this task

To delete an ATP monitoring rule:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the ATP Monitor Rules tab.
3. Select the applicable ATP monitoring rule and click the **Delete** icon.

Onhand Inventory Monitor Rules

An onhand inventory monitor rule uses the Inventory Monitor time-triggered transaction, which can be found and configured under the General process type.

For more information about configuring transactions, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*. For more information about the Inventory Monitor, see “Inventory Monitor” on page 253.

Creating an Onhand Inventory Monitor Rule

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Monitor Rules tab.
3. In the Onhand Inventory Monitor Rules inner panel, click the **Create New** icon. The Onhand Inventory Monitor Rule Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 11 for field value descriptions.
5. Click the **Save** icon.

Table 11. Onhand Inventory Monitor Rule Details Fields

| Field | Description |
|--|---|
| Item ID | The ID of the inventory item that is being monitored. |
| Node | From the drop-down list, select the node that stores the inventory item. |
| Action Setup for Minimum Quantity | |
| High Quantity | The high quantity, below which the inventory monitor raises the associated action. |
| Medium Quantity | The medium quantity, below which the inventory monitor raises the associated action. |
| Low Quantity | The low quantity, below which the inventory monitor raises the associated action. |
| Action | Select a predefined action that is raised whenever the available quantity falls below the specified quantity. |
| Action Setup for Maximum Quantity | |

Table 11. Onhand Inventory Monitor Rule Details Fields (continued)

| Field | Description |
|-----------------|---|
| High Quantity | The high quantity above which the inventory monitor raises the associated action. |
| Medium Quantity | The medium quantity, above which the inventory monitor raises the associated action. |
| Low Quantity | The low quantity, above which the inventory raises the associated action. |
| Action | Select a predefined action that is raised whenever the available quantity rises below the specified quantity. |

Modifying an Onhand Inventory Monitor Rule Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Monitor Rules tab.
3. In the Onhand Inventory Monitor Rules inner panel, select the appropriate and click the **Details** icon. The Onhand Inventory Monitor Rule Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 11 on page 31 for field value descriptions.
5. Click the **Save** icon.

Deleting an Onhand Inventory Monitor Rule Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Monitor Rules tab.
3. In the Onhand Inventory Monitor Rules inner panel, select the appropriate rule and click the **Delete** icon.

Defining Product Classes

You can define common codes used when indicating a product class in the Inventory Console. The **product class** is a product item's classification, such as first quality, second quality, or finished good.

Creating a Product Class

About this task

To create a product class:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Product Classes tab.
3. Choose the **Create New** icon. The Product Class Details pop-up window displays.
4. In the Product Class field, enter the product class.
5. In the Short Description field, enter a brief description of the product class.

6. In the Long Description field, enter a more detailed description of the product class.
7. Choose the **Save** icon.

Modifying a Product Class

About this task

To modify a product class:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Product Classes tab.
3. Select the applicable product class and choose the **Details** icon. The Product Class Details pop-up window displays.
4. In Short Description, enter a brief description of the product class.
5. In Long Description, enter a more detailed description of the product class.
6. Choose the **Save** icon.

Deleting a Product Class

About this task

To delete a product class:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Product Classes tab.
3. Select the applicable product class and choose the **Delete** icon.

Defining Inventory Reasons

You can define common codes used to identify reasons for performing a modification in the Inventory Console.

Creating an Inventory Reason

About this task

To create an inventory reason:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Inventory Reasons tab.
3. Choose the **Create New** icon. The Inventory Reason Details pop-up window displays.
4. In the Inventory Reason field, enter the inventory reason.
5. In the Short Description field, enter a brief description of the inventory reason.
6. In the Long Description field, enter a more detailed description of the inventory reason.

7. Choose the **Save** icon.

Modifying an Inventory Reason

About this task

To modify an inventory reason:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Inventory Reasons tab.
3. Select the applicable inventory reason and choose the **Details** icon. The Inventory Reason Details pop-up window displays.
4. In Short Description, enter a brief description of the inventory reason.
5. In Long Description, enter a more detailed description of the inventory reason.
6. Choose the **Save** icon.

Deleting an Inventory Reason

About this task

To delete an inventory reason:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Inventory Reasons tab.
3. Select the applicable inventory reason and choose the **Delete** icon.

Defining IBA Rules

You can configure Item-Based Allocation rules for an inventory organization by enabling IBA and configuring the IBA demand type, conditions, and sequences.

Setting Up IBA Rules

About this task

To set up IBA rules:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the IBA Rules tab.
3. Enter the information in the applicable fields. Refer to Table 12 on page 35 for field value descriptions.
4. Choose the **Save** icon.

Table 12. IBA Tab

| Field | Description |
|---|--|
| Use Item Based Allocation | Check this box to enable item-based allocation. Item-based allocation is applicable for the items and nodes which have the 'Item Based Allocation Allowed' attribute enabled. For more information about item-based allocation, see the <i>Sterling Selling and Fulfillment Foundation: Product Concepts Guide</i> . |
| Demand Type To Look For Availability During Item Based Allocation | From the drop-down list, select the demand type to be used for obtaining availability during the item-based allocation process. |
| Supply Optimization | |
| Use the earliest available supply for all demands | Select this option if you want to use the First In First Out (FIFO) rule. The onhand inventory will be considered first for order promising and then the subsequent inventory will be considered. If the current demand is still not satisfied, the inventory arriving after the demand date will be considered. |
| Optimize supply selection to maximize availability | Select this option if you want to use the inventory that will be arriving on a date that is nearest to the demand date, that is, the inventory arriving on a date that is less than or equal to the demand date, will be considered first. If the current demand is still not satisfied, the inventory arriving after the demand date will be considered. |
| Order Solution Criteria | |
| Condition Name | The name of the condition for this priority. |
| Condition ID | The ID for this condition. |
| Order Selection Rules for Inventory Allocation | |
| Sequence No | The sequence number for this IBA priority. |
| Description | The description of this IBA sequence rule. |

Creating Condition Details

About this task

To create an IBA Condition detail:


Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the IBA Rules tab.
3. In the Order Selection Criteria panel, choose the **Create New** icon. The Condition Detail pop-up window displays.
4. Enter the information in the applicable fields. Refer to Table 13 for field value descriptions.
5. Choose the **Save** icon.

Table 13. Condition Details

| Field | Description |
|----------------|--|
| Condition Name | Enter the name of the condition for this priority. |

Table 13. Condition Details (continued)

| Field | Description |
|-----------------|---|
| Condition ID | Enter the ID for this condition. |
| Condition Value | <p>This field contains information you enter in the General Condition Builder. Click  to display the General Condition Builder.</p> <p>The <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i> contains information about Condition Builder attributes, and the <i>Catalog Management: Configuration Guide</i> contains information about using the Condition Builder.</p> |

Modifying Condition Details

About this task

To modify a Condition Detail:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the IBA Rules tab.
3. In the Order Selection Criteria panel, select the Condition Name and choose the **Details** icon.
4. Edit the detail and choose the **Save** icon.

Deleting Condition Details

About this task

To delete an IBA Condition detail:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the IBA Rules tab.
3. In the Order Selection Criteria panel, select the Condition Name and choose the **Delete** icon.

Creating IBA Sequence Header Details

About this task

To create an IBA Sequence Header detail:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the IBA Rules tab.
3. In the Order Selection Rules for Inventory Allocation panel, choose the **Create New** icon. The IBA Sequence Header Details pop-up window displays.
4. Enter the information in the applicable fields. Refer to Table 14 on page 37 for field value descriptions.

5. Choose the **Save** icon.

Results

Table 14. IBA Sequence Header Details

| Field | Description |
|--------------|---|
| Sequence No | Enter the sequence number for this IBA sequence. If you enter values that are several increments apart, you will be able to insert additional sequence numbers later. |
| Description | Enter the description of this IBA sequence rule. |
| Condition ID | Select from the available Condition IDs. |

Modifying IBA Sequence Header Details

About this task

To modify an IBA Sequence Header detail:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the IBA Rules tab.
3. In the Order Selection Rules for Inventory Allocation panel, select the Sequence Number and choose the **Details** icon.
4. Edit the Sequence Header and choose the **Save** icon.

Deleting IBA Sequence Header Details

About this task

To delete an IBA Sequence Header detail:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the IBA Rules tab.
3. In the Order Selection Rules for Inventory Allocation panel, select the Sequence Number and choose the **Delete** icon.

Defining Additional Inventory Rules

About this task

You can define additional rules that pertain to the Sterling Selling and Fulfillment Foundation inventory handling functionality.

To set up additional inventory rules:

Procedure

1. From the tree in the application rules side panel, choose Inventory Rules. The Inventory Rules window displays in the work area.
2. Choose the Other Rules tab.

3. Enter information in the applicable fields. Refer to Table 15 for field value descriptions.
4. Choose the **Save** icon.

Table 15. Other Rules Tab

| Field | Description |
|---|---|
| Inventory Costing Rules | |
| Inventory Costing Required | Select this field if inventory costing is necessary. When inventory costing is used, you must select a Costing Method. |
| Costing Method | |
| FIFO | Select this inventory costing method if you calculate inventory costs using a First In First Out (FIFO) algorithm. If you select this costing method, you must also specify a Sales Cost Level. |
| Average Cost | Select this inventory costing method if you calculate inventory costs using an average item cost algorithm. If you select this costing method, you must also specify a Standard Cost Computation Rule. |
| Other Rules | |
| Validate Item During Inventory Adjustment | Check this box if you want the item to be validated against the catalog during inventory adjustments. |
| Optimize supply on earliest ETA | Check this box when preference should be given to supply available earlier, for example ONHAND is given preference to PO supply. |
| Assume FEFO if A 'Ship-By Date' Is Not Provided | Check this box to ship all perishable items on a First Expired First Out (FEFO) basis, if the ship date is not provided for the items. If you uncheck this box, all perishable items for which the ship date is not provided are not shipped on the FEFO basis. |
| Create Demand Details | Check this box to enable recording of the inventory demand details whenever a demand is created in Sterling Selling and Fulfillment Foundation. You can view the demand details in the Inventory Console. Important: If you have implemented either the EXTERNAL_DEMAND_CHANGE event or the DEMAND_CHANGE event to be raised when inventory changes, then IBM recommends to check this box. Warning: If you check this box, it affects the overall performance of the entire system due to the large amount of data that can be potentially recorded. |
| Synchronize Expiration Date Across Inventory Reservations Based On The Reservation ID | Check this box to synchronize the expiration date across inventory reservations based on the reservation identifier. |
| Read Availability Across Items Sharing The Same | From the drop-down list, select the option used to identify items (for example, GTIN). When selected, the system will read the availability of all items that share the same item identifier for that option. |

Table 15. Other Rules Tab (continued)

| Field | Description |
|---|--|
| Default Reservation Expiration Time (in Hours) | Enter the number of hours after which you want the reservation to expire. This affects only those reservations for which the expiration date is not specified when creating the reservation. |
| Skip Inventory Audits For Adjustment Types | |
| Return | Select this field if you want inventory audit logging to be disabled for return adjustment type. |
| Receipt | Select this field if you want inventory audit logging to be disabled for receipt adjustment type. |
| Adjustment | Select this field if you want inventory audit logging to be disabled for adjustment type. |
| Shipment | Select this field if you want inventory audit logging to be disabled for shipment adjustment type. |
| Backordered Inventory Rules | |
| Put Inventory On Hold For Sourcing On Backorder From Node | Select this field if you want the system to put inventory on hold for sourcing when an item that is released to a node is backordered. This functionality is to prevent inventory from being scheduled to nodes that are providing an incorrect inventory picture. When a node is on hold for sourcing it is ignored when scheduling algorithms are run. |
| Inventory Will Be On Hold For A Minimum Of <number of hours> Hours | If you select the Put Inventory On Hold For Sourcing On Backorder From Node option, enter the time (in hours) that nodes are placed on hold for sourcing. Note: If the issue that caused the discrepancy in the inventory picture is resolved before this minimum time has passed, the node can be manually released from the sourcing hold through the Application Consoles. |
| After The Minimum Hold Time, The Node Will Be Eligible For Sourcing From The Next Planned Inventory Synchronization Time Daily Inventory Synchronization Time | Enter the time (in your organization's time format) at which daily inventory synchronization occurs. Once the node has been on hold for the minimum hold time, it is made available at the next Daily Inventory Synchronization Time. Note: If the issue that caused the discrepancy in the inventory picture is resolved before the Daily Inventory Synchronization Time has been reached, the node can manually be released from sourcing hold through the Application Consoles. |
| Use Future Supply For Nodes Where Inventory Is On Hold For Sourcing. | Select this option to allow the use of a future supply for nodes where inventory is on hold for sourcing. |

Chapter 4. Configuring Inventory Types and Considerations

Defining Inventory Considerations

About this task

You can identify the supply and demand type associations used to determine inventory availability for a specific demand type.

For example, you can configure a Scheduled demand type to check for availability against Onhand supply and Purchase Order supply. You can also configure an Allocated demand type to check for availability against only Onhand supply. With this configuration, scheduled inventory demands check for inventory that is both physically available and available in the future, while allocated inventory demands only check for inventory that is physically available.

Note: Unassigned demands are mapped to a supply using the demand owner's default Distribution Rule. If you do not configure the default Distribution Rule, the APIs may return results as AVAILABLE, causing orders to get accepted even though they cannot be actually sourced.

Note: When considering supply types and demand types used by particular nodes, note that the supply types at a ship node marked as INFINITE are considered to be ONHAND. Therefore, ONHAND supply types must be mapped to particular demand types, otherwise, orders for the node are backordered when availability calculations are made for the demand type.

To define inventory considerations:

Procedure

1. From the tree in the application rules side panel, choose Inventory Types and Considerations > Supply Types, Demand Types, and Considerations. The Inventory Types and Considerations window displays in the work area.
2. Choose the Inventory Considerations tab.
3. From the Demand Types columns, select the check boxes of the supply types you want to be considered when inventory availability is performed for that demand type.
4. Choose the **Save** icon.

Defining Inventory Supply Types

You can define parameters for the supply types used when maintaining inventory in Sterling Selling and Fulfillment Foundation.

Creating an Inventory Supply Type

About this task

To create an inventory supply type:

Procedure

1. From the tree in the application rules side panel, choose Inventory Types and Considerations > Supply Types, Demand Types, and Considerations. The Inventory Types and Considerations window displays in the work area.
2. Choose the Inventory Supply Types tab.
3. From the Inventory Supply Types table, choose the **Create New** icon. The Inventory Supply Type Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 16 for field value descriptions.
5. Choose the **Save** icon.

Table 16. Inventory Supply Type Details Pop-up Window

| Field | Description |
|--|---|
| Supply Type | Enter the name of the new supply type. Important: The supply type you enter here is the code that is passed and returned by the Sterling Selling and Fulfillment Foundation APIs. It is also the code that displays in the Inventory Console of the Sterling Selling and Fulfillment Foundation user interface. |
| Description | Enter a brief description of the new supply type. Important: The description you enter here is how this supply type is represented displays throughout the Applications Manager. |
| Costing Required | Select this field if you want costing data to be generated for inventory in this supply type. |
| Onhand Supply | Select this field if the supply type indicates that supply is physically available at a node. Note: Only supply types identified as Onhand are considered for inventory audits. |
| Retain Reference | Select this field to record references along with the supply record. |
| Trigger Item Based Allocation | Check this box to enable the triggering of item-based allocation when changes occur with this supply type. This is applicable for non on-hand supply types only. |
| Use Considered Demand Type for Item Based Allocation | Check this box to indicate that the demand with demand types, that consider this supply type, should be considered for item-based allocation. |
| Enable Audit | Check this box to enable inventory audit logging. |

Modifying an Inventory Supply Type

About this task

To modify an inventory supply type:

Procedure

1. From the tree in the application rules side panel, choose Inventory Types and Considerations > Supply Types, Demand Types, and Considerations. The Inventory Types and Considerations window displays in the work area.
2. Choose the Inventory Supply Types tab.

3. From the Inventory Supply Types table, locate the applicable inventory supply type and choose the **Details** icon. The Inventory Supply Type Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 16 on page 42 for field value descriptions.
5. Choose the **Save** icon.

Deleting an Inventory Supply Type

About this task

To delete an inventory supply type:

Procedure

1. From the tree in the application rules side panel, choose Inventory Types and Considerations > Supply Types, Demand Types, and Considerations. The Inventory Types and Considerations window displays in the work area.
2. Choose the Inventory Supply Types tab.
3. From the Inventory Supply Types table, locate the applicable inventory supply type and choose the **Delete** icon.

Defining Inventory Demand Types

You can define parameters for the demand types used when maintaining inventory in Sterling Selling and Fulfillment Foundation.

Note: If you create a custom demand type and want the availability picture displayed in the Inventory Console to be for the custom demand type, you need to extend the user interface and pass the demand type to the getATP API call.

Creating an Inventory Demand Type

About this task

To create an inventory demand type:

Procedure

1. From the tree in the application rules side panel, choose Inventory Types and Considerations > Supply Types, Demand Types, and Considerations. The Inventory Types and Considerations window displays in the work area.
2. Choose the Inventory Demand Types tab.
3. From the Inventory Demand Types table, choose the **Create New** icon. The Inventory Demand Type Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 17 on page 44 for field value descriptions.
5. Choose the **Save** icon.

Table 17. Inventory Demand Type Details Pop-up Window

| Field | Description |
|--|---|
| Demand Type | Enter the name of the demand type. Important: The demand type you enter here is the code that is passed and returned by the Sterling Selling and Fulfillment Foundation APIs. It is also the code that displays in the Inventory Console. |
| Description | Enter a brief description of the demand type. Important: The description you enter here is how this demand type is represented throughout the Applications Manager. |
| Demand Is Communicated To External Systems | Select this field to indicate that demands of this type are communicated to an external system for use in availability computations. |
| Committed Level | |
| Released | Select this option if the demand type represents a demand that has been scheduled and released. |
| Promised | Select this option if the demand type represents reservations created without orders. |
| Non-Committed | Select this option if the demand type represents demands that are not yet reserved or assigned to a specific node. |

Modifying an Inventory Demand Type

About this task

To modify an inventory demand type:

Procedure

1. From the tree in the application rules side panel, choose Inventory Types and Considerations > Supply Types, Demand Types, and Considerations. The Inventory Types and Considerations window displays in the work area.
2. Choose the Inventory Demand Types tab.
3. From the Inventory Demand Types table, locate the applicable inventory demand type and choose the **Details** icon. The Inventory Demand Type Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 17 for field value descriptions.
5. Choose the **Save** icon.

Deleting an Inventory Demand Type

About this task

To modify an inventory demand type:

Procedure

1. From the tree in the application rules side panel, choose Inventory Types and Considerations > Supply Types, Demand Types, and Considerations. The Inventory Types and Considerations window displays in the work area.
2. Choose the Inventory Demand Types tab.

- From the Inventory Demand Types table, locate the applicable inventory demand type and choose the **Delete** icon.

Inventory Availability Safety Factor

About this task

Sterling Selling and Fulfillment Foundation enables you to define an inventory availability safety factor, that indicates what percentage of current or future inventory should be excluded during order promising.

For example, you may consider that your Planned Purchase Order supply is less reliable than your In Transit supply for order promising. Therefore, you may want to exclude 60% of the Planned Purchase Order supply type when promising, as opposed to 10% of the In Transit supply type.

By default, fractional quantities are not truncated. That functionality can be turned off by modifying the `yfs.install.displaydoublequantity` property 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 behavior is as follows:

- If the property is set to Y, the fractional quantity is used after applying the percentage. This is the default behavior.
- If the property is set to N, the fractional quantity is truncated to the nearest lower integer after applying the percentage. For example, if a purchase order is placed with a quantity of 10, and the inventory availability safety factor percentage of the PO_PLACED supply type is set to 25%, a quantity of 7 would be available to fulfill future demand, as opposed to 7.5.

To modify, enable, or disable the inventory availability safety factor:

Procedure

- From the tree in the application rules side panel, choose Inventory Types in Considerations > Inventory Availability Safety Factor. The Inventory Availability Safety Factor window displays in the work area. Refer to Table 18 for field descriptions.

Table 18. Inventory Availability Safety Factor Window

| Field | Description |
|--------------------------|---|
| Supply Type | The name of the supply type. |
| Description | The description of the supply type. |
| Apply Safety Factor | Check this if you want to exclude the specified safety factor percentage during order promising for this supply type. Note: For safety factors to apply, this flag must also be checked for the node type and scheduling rule. For more information, refer to Chapter 7, “Configuring Inventory Node Type Rules,” on page 53, and the <i>Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide</i> . |
| Safety Factor Percentage | The percentage you want to exclude during order promising for this supply type. |

2. In the Channel Applications Manager, perform the following steps:
 - a. Click the **Expand** icon adjacent to **Configure Inventory Synchronization** to expand the branch.
 - b. Click the Advanced Configurations link.
 - c. Select Configure Availability Safety Factors.
3. To modify current inventory availability safety factors, select the Current Inventory Availability Safety Factor tab. To modify future inventory availability safety factors, select the Future Inventory Availability Safety Factor tab.
4. To modify the Safety Factor Percentage, double-click the desired percentage and enter a new value, between 0 and 100.

Note: The Safety Factor Percentage indicates the percentage to **exclude** during order promising.

5. To enable or disable the safety factor, check or uncheck the desired Apply Safety Factor check box.
6. Click the **Save** icon.

Results

A Safety Factor Percentage or Quantity specified for an individual item overrides values specified for a supply type. For more information about defining inventory information for product items, see the *Catalog Management: Configuration Guide*.

Chapter 5. Configuring Cost Factor Groups

When implementing inventory costing, cost factors can be applied to inventory to determine a derived cost. The cost factors can vary for different Enterprise or Seller organizations based on vendor preference, transaction type, and so forth. You define a cost factor group to represent a set of cost factors that are applied for a specific organization's needs.

You define inventory cost factor groups at the Enterprise level. In general, each Enterprise should specify the cost factor group to be used for the following purposes when determining the derived cost:

- Landed Cost Determination
- Standard Cost Determination
- Work in Process Handling

Enterprises should also specify the cost factor group to be used for the following purposes for each Seller organization:

- Landed Cost Determination
- Standard Cost Determination

Enterprises should also specify the cost factor group to be used for Work in Process Handling at the item level for each physical kit component item.

Defining Inventory Cost Factor Groups

About this task

To define inventory cost factor groups:

Procedure

1. From the tree in the application rules side panel, choose Cost Factor Groups. The Cost Factor Groups window displays in the work area.
2. Choose the **Create New** icon. The Cost Factor Group Detail window displays.
3. In the Group Name field, enter the name you want to use to reference this specific set of cost factors.
4. In the Group Description field, enter a brief description of this set of cost factors.
5. Choose the **Save** icon.
6. The newly created cost factor group name now displays in the Cost Factor Group list window.

Associating Cost Factors with a Cost Factor Group

About this task

To associate cost factors with a cost factor group:

Procedure

1. In the Cost Factor Group list window, double-click on the cost factor group for which you want to define cost factors.

The Cost Factor Group Detail window displays with the Create New icon enabled in the Cost Factors panel.

2. Choose the **Create New** icon. The Cost Factors Details window displays.
3. Enter information in the applicable fields. Refer to Table 19 for field value descriptions.
4. Choose the **Save** icon.

Table 19. Cost Factor Details Window

| Field | Description |
|---|---|
| Cost Factor Name | Enter a unique name for this cost factor. |
| Cost Factor Description | Enter a brief description of this cost factor. |
| Cost Factor Value | Specify the actual value used by this cost factor; percentage or amount. |
| Cost Factor Calculation Method | |
| Percentage | Select this option if this cost factor is calculated using a specific percentage. |
| Amount | Select this option if this cost factor is calculated using a specific amount. Note: For physical kits this applies to the parent item. |
| Cost Factor Application Method (for Amount cost factor calculation method only) | |
| Volume | Select this option if you want the amount cost factor calculation to be based on the inventory item's volume. An item's volume is calculated as height * width * length. You must then select the applicable unit of measure from the Cost Factor UOM drop-down list. Note: While configuring dimension UOMs, be sure to configure the equivalent Volume UOM. For more information about defining the equivalent volume UOM, see the <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i> . |
| Weight | Select this option if you want the amount cost factor calculation to be based on the inventory item's weight. You must then select the applicable unit of measure from the Cost Factor UOM drop-down list. |
| Quantity | Select this option if you want the amount cost factor calculation to be based on the quantity of the inventory item. You must then select the applicable unit of measure from the Cost Factor UOM drop-down list. Note: The non-applicable UOMs are ignored during cost factor calculation. |
| Cost Factor UOM | Select the unit of measure that is applicable to the Cost Factor Application Method that you selected. |

Chapter 6. Configuring Product Item Specific Distribution Groups

You can create a set of nodes/external organizations that can be used when determining product item sourcing. You can define distribution rules that establish the ship node determination process within the distribution group. The rules determine the default node that an item should be sourced from within a group based on priority. You can create rules for individual product items at a source node or for the entire source node.

Creating a Distribution Group

About this task

To create a distribution group:

Procedure

1. From the tree in the application rules side panel, choose Product Sourcing Distribution Group. The Product Sourcing Distribution Groups window displays in the work area.
2. Choose the **Create New** icon. The Distribution Group Detail window displays.
3. In Distribution Group, enter the name of the distribution group.
4. In Description, enter a brief description of the distribution rule.
5. Choose the **Save** icon.

Adding Nodes/External Organizations to a Distribution Group

About this task

To add a node/external organization to a distribution group:

Procedure

1. In the Distribution Group Details window, choose the **Create New** icon. The Distribution Details pop-up window displays.
2. Enter information into the applicable fields. Refer to Table 20 for field value descriptions.
3. Choose the **Save** icon.

Results

If you adding nodes or external organizations to a distribution group, do not use the advanced tab. Use sourcing rules instead. For more information about configuring sourcing rules, see the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

Table 20. Distribution Details Window

| Field | Description |
|--------|-------------|
| Source | |

Table 20. Distribution Details Window (continued)

| Field | Description |
|---------------------|---|
| Source Ship Node | Select Source Ship Node and select the applicable node if you want to add a node within your organization to the distribution group. |
| Source Organization | Select Source Organization and select the applicable organization if you want to add an external organization to the distribution group. |
| Priority | Enter the node/external organization's priority within the distribution group. Note: Priority is not unique to a distribution group, therefore more than one distribution group can have the same priority. |

Modifying a Distribution Group's Node/External Organization

About this task

To modify a distribution group's node/external organization:

Procedure

1. In the Distribution Rule Details window, choose the Distribution Detail tab.
2. Select the applicable distribution detail and choose the **Details** icon. The Distribution Details pop-up window displays.
3. Enter information into the applicable fields. Refer to Table 20 on page 49 for field value descriptions.
4. Choose the **Save** icon.

Deleting a Distribution Group's Node/External Organization

About this task

To delete a distribution group's node/external organization:

Procedure

1. In the Distribution Rule Details window, choose the Distribution Detail tab.
2. Select the applicable distribution detail and choose the **Delete** icon.

Adding Advanced Distribution Details to a Distribution Group (For Backward Compatibility Only)

About this task

You can add specific details, such as sourcing information, and assign them a date range through which they are effective.

Note: IBM strongly recommends the use of sourcing rules instead of advanced distribution groups. This feature is provided for backward compatibility purposes only.

Note: If setting up advanced distribution rules, do not use the base distribution rules under the distribution detail tab.

To add advanced distribution details to a distribution rule:

Procedure

1. In the Distribution Group Details window, choose the Advanced tab.
2. From the Distribution table, choose the **Create New** icon. The Distribution Details pop-up window displays.
3. Enter information in the applicable fields. Refer to Table 21 for field value descriptions.
4. Choose the **Save** icon.

Table 21. Advanced Distribution Details Window

| Field | Description |
|---------------------------------------|--|
| All Items | Select this option to apply the distribution rule to all of the items in the node you are setting the rule up for. |
| Apply To Specific Item At This Source | Select this option to apply the distribution rule to a specific item in the node or organization you are setting the rule up for. |
| Primary Info | |
| Item ID | If you selected Apply To Specific Item At This Source, enter the item ID for which you are creating the Distribution Rule. |
| Active | Check Active if the distribution rules are active. |
| Item Name at Node | If you selected Apply To Specific Item At This Source, enter the node's name for the item. The distribution record created for the inventory consolidator displays in the Inventory Console. |
| Priority | Enter a priority number for the node for this item and inventory scheduling, with 0 being the highest priority. |
| Effective Start Date | The date the distribution details take effect. |
| Effective End Date | The date after which the distribution details are no longer applied. |
| Source | |
| Source Ship Node | Choose Source Ship Node and select the applicable node if you are setting up the distribution details to be sourced from a particular ship node. |
| Source Organization | Choose Source Organization and select the applicable organization if you are setting up the distribution details to be sourced from a particular organization. |

Deleting Advanced Distribution Details

About this task

To delete a advanced distribution details:

Procedure

1. In the Distribution Group Details window, choose the Advanced tab.
2. From the Distribution table, select the applicable distribution details and choose the **Delete** icon.

Deleting a Distribution Group

About this task

To delete a distribution group:

Procedure

1. From the tree in the application rules side panel, choose Product Sourcing Distribution Group. The Product Sourcing Distribution Groups window displays in the work area.
2. Select the applicable distribution rule and choose the **Delete** icon.

Chapter 7. Configuring Inventory Node Type Rules

You can create inventory rules based on node types. These rules are applied to nodes belonging to the node type on the rule. For more information about creating node types, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

Creating an Inventory Node Type Rule

About this task

To create an inventory node type rule:

Procedure

1. From the tree in the application rules side panel, choose Inventory Node Type Rules. The Inventory Node Type Rules window displays in the work area.
2. Choose the **Create New** icon. The Inventory Node Type Rule Details window displays.
3. Enter information into the applicable fields. Refer to Table 22 for field value descriptions.
4. Click the **Save** icon.

Table 22. Inventory Node Type Rule Details Pop-up

| Field | Description |
|---|--|
| Node Type | Select the node type for which this rule should be used. For more information about node types, see the <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i> |
| Apply On Hand Safety Factor To On Hand Inventory Availability | Check this flag to apply the on hand safety factor to on hand inventory availability for nodes with this node type. Note: For safety factors to apply, this flag must also be checked for the supply type and scheduling rule. For more information about Safety Factors, see "Inventory Availability Safety Factor" on page 45. |
| Apply Future Safety Factor To Future Inventory Availability | Check this flag to apply the future safety factor to future inventory availability for nodes with this node type. Note: For safety factors to apply, this flag must also be checked for the supply type and scheduling rule. For more information about Safety Factors, see "Inventory Availability Safety Factor" on page 45. |

Modifying an Inventory Node Type Rule

About this task

To modify an inventory node type rule:

Procedure

1. From the tree in the application rules side panel, choose Inventory Node Type Rules. The Inventory Node Type Rules window displays in the work area.
2. Select the applicable inventory node type rule and choose the **Details** icon. The Service Slot Group Details pop-up window displays.

3. Enter information into the applicable fields. Refer to Table 22 on page 53 for field value descriptions.
4. Click the **Save** icon.

Deleting an Inventory Node Type Rule

About this task

To delete an inventory node type rule:

Procedure

1. From the tree in the application rules side panel, choose Inventory Node Type Rules. The Inventory Node Type Rules window displays in the work area.
2. Select the applicable inventory node type rule and choose the **Delete** icon.

Chapter 8. Configuring Resource Capacity

Resource capacity components are used to determine delivery service item and provided service item availability. Resource capacity availability is used to determine appointments for a delivery service and/or provided service of a defined unit of measure for specific time slots and geographical regions.

For more information about configuring delivery services and provided services, see the *Catalog Management: Configuration Guide*.

Defining Capacity Rules

About this task

You can define rules to set capacity reservation expiration limits and to allow resource capacity assignments to span across service slots.

To define capacity rules:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Capacity Rules. The Capacity Rules window displays.
2. Enter information in the applicable field. Refer to Table 23 for field value descriptions.

Table 23. Capacity Rules Window

| Field | Description |
|--|---|
| Capacity Reservation | |
| Default Capacity Reservation Expiration Time (in Hours) | Enter the number of hours after which a capacity reservation should expire. Once the expiration time has passed, the reservation is available for purging. |
| Capacity Across Slots | |
| Allow Spanning of Capacity Across Slots | Check this box to allow service resource capacity assignments to span across multiple service slots. Note: This option is only available to resource pools that maintain capacity at the service resource level and have a service slot group that is not hierarchical. |
| Span Across Non-Working Shifts For <n> Minutes | Enter the number of minutes for which a shift can span across its adjacent nonworking shift or shifts. |
| Ignore Prereserved Capacity | |
| Ignore prereserved capacity when recomputing appointment options | Check this box if you do not want to consider pre-reserved capacity for capacity calculations. |

3. Choose the **Save** icon.

Defining Region Usage for Resource Pools

About this task

A region schema is the complete hierarchical set of regions that define a given geography. A region is configured as a specific territory. For example, you can create a region for a complete state, city, or town. For more information about configuring region schemas, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

You can define the region schemas used for configuring delivery service resource pools and provided service resource pools. For more information about resource pools, see “Defining Resource Pools” on page 59.

To define region usage for resource pools:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Region Usage For Resource Pools. The Region Usage For Resource Pools pop-up window displays.
2. From Schema for Product being delivered, select the region schema you want to use when creating delivery service resource pools.
If you have preexisting resource pools and you have added additional regions to them, Schema for Product being delivered is not selectable.
3. From Schema for Provided Service, select the region schema you want to use when creating provided service resource pools.
If you have preexisting resource pools and you have added additional regions to them, Schema for Provided Service is not selectable.
4. Choose the **Save** icon.

Defining Slot Groups

A slot group is a collection of service slots. A service slot is defined as a period of time against which service item promising can be made.

You can define multiple slots and slot groups, enabling you to schedule appointments of different granularity for different resource pools. For example, you may want to associate a slot group comprised of service slots that promise four-hour windows with a delivery service resource pool, and a slot group comprised of service slots that promise two-hour windows with a provided service resource pool.

The slots within a slot group can be defined at different levels of hierarchy. The hierarchical slot group may be comprised of parent slots and subsequent child slots. For example, if a full day slot is considered as the parent slot, the morning and afternoon slots within the full day slot are considered child slots. The child slot itself can be branched to a number of successive child slots. For example, the morning slot can be divided into AM1 and AM2 and the afternoon slot in PM1 and PM2.

Within a slot group, slots can overlap when the capacity is maintained at the resource pool level. For example, if a parent slot extends from 08:00 to 18:00, the child slots can be between 08:00 to 13:00 and 12:00 to 18:00, and so forth.

The slot group associated with a resource pool maintains the capacity at the service resource level and does not allow overlapping of slots at the same level in the hierarchy. For example, a slot group that has a full day slot at the parent level and a morning and afternoon slots as its child slots.

Note: Slot groups are defined by the capacity organization and not by the organization that provides the capacity.

For more information about associating a slot group with a resource pool, see “Defining Resource Pools” on page 59.

Creating a Slot Group

About this task

To create a slot group:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Slot Group. The Service Slot Groups window displays in the work area.
2. Choose the **Create New** icon. The Service Slot Group Details pop-up window displays.
3. Enter information in the applicable fields. Refer to Table 24 for field value descriptions.
4. Choose the **Save** icon.

Table 24. Service Slot Group Details Pop-Up Window

| Field | Description |
|---------------|--|
| Slot Group ID | Enter the name of the slot group. |
| Description | Enter a brief description of the slot group. |
| Service Slots | A list of the slot group's service slots. |

Creating Hierarchical Service Slots

About this task

To create hierarchical service slots:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Slot Group. The Service Slot Groups window displays in the work area.
2. Locate the applicable slot group and choose the **Details** icon. The Service Slot Group Details pop-up window displays.
3. Select the service slot under which you need to create the child slots.
4. Choose the **Create New** icon. The Service Slot Details pop-up window displays.
5. Enter information in the applicable fields. Refer to Table 25 for field value descriptions.
6. Choose the **Save** icon.

Table 25. Creating Hierarchical Service Slots Screen

| Fields | Description |
|--------------------------|--|
| Service Slot Description | Enter a brief description of the service slot. |

Table 25. Creating Hierarchical Service Slots Screen (continued)

| Fields | Description |
|-------------------------|--|
| Start Time | Enter the starting time for the service slot. |
| End Time | Enter the ending time for the service slot. |
| Can Use For Appointment | <p>Check this box to use this slot for planning appointments.</p> <p>In some business cases, certain parent slots are used only for the purpose of defining maximum available capacity of their child slots and they are not actually used for appointment planning. These slots are configured to prevent over promising against their child slots. Therefore, you must be able to configure whether or not a slot can be used for planning appointments. If a slot is configured such that it cannot be used for appointment planning, it is shown as an available option when planning an appointment. Therefore, with this configuration, if a full day Slot is configured such that it cannot to be used for planning appointments, you can only see morning and afternoon slots as available when planning appointments.</p> |

Modifying a Slot Group

About this task

To modify a slot group:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Slot Group. The Service Slot Groups screen displays in the work area.
2. Locate the applicable slot group and choose the **Details** icon. The Service Slot Group Details pop-up window displays.
3. Enter information in the applicable fields. Refer to Table 24 on page 57 for field value descriptions.
4. Choose the **Save** icon.

Modifying Hierarchical Service Slots

About this task

To modify hierarchical service slots:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Slot Group. The Service Slot Groups screen displays in the work area.
2. Select the applicable slot group and choose the **Details** icon. The Service Slot Group Details pop-up window displays.
3. Select the appropriate slot and choose the **Details** icon. The Service Slot Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 24 on page 57 for field value descriptions.
5. Choose the **Save** icon.

Deleting a Slot Group

About this task

To delete a slot group:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Slot Group. The Service Slot Groups screen displays in the work area.
2. Locate the applicable slot group and choose the **Delete** icon.

Deleting the Hierarchical Service Slots

About this task

To delete a hierarchical service slot:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Slot Group. The Service Slot Groups window displays in the work area.
2. Locate the applicable slot group and choose the **Details** icon.
3. Select the appropriate slot and choose the **Delete** icon
You cannot delete a parent slot if child slots exist. Therefore, you must delete the child slots before deleting the parent slot.

Defining Resource Pools

Delivery and provided services are supplied by defined service nodes. Service nodes can provide multiple delivery and provided services. Resource pools provide the ability to define how much service capacity is available for these services by geographical area and time slots. A resource pool is an aggregate collection of resources needed to perform a delivery service or provided service. A resource pool is defined by the organization that is providing the capacity.

Note: For an enterprise user to access the resource pool or capacity of another enterprise, that user must have access to the other enterprise's capacity organization. In addition, Data Access Policies must be enabled.

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

A resource pool comprises the following components:

- Service slot group
- Resource calendar
- Delivery items or service items
- Regions served
- Standard capacity definition
- Team Members

Creating and Modifying a Resource Pool

About this task

To create a resource pool:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Resource Pools. The Resource Pool Search window displays in the work area.
2. Choose the **Create New** icon. The Create Resource Pool pop-up window displays.
3. Enter information in the applicable fields. Refer to Table 26 for field value descriptions.

Table 26. Create Resource Pool Pop-Up Window

| Field | Description |
|---------------------------------------|---|
| Resource Pool ID | Enter the name of the resource pool as you want it to appear. |
| Capacity Organization | Select the organization that maintains the capacity within this resource pool. |
| Resource Pool Description | Enter a brief description of the resource pool. |
| Service Slot Group | Select the service slot group you want to use to determine service promising. For more information about configuring service slot groups, see “Defining Slot Groups” on page 56. Important: When you change the service slot group, ensure that you reset the capacity of the original slot group in the Capacity Console. The capacity should be made unavailable. For more information about using the Capacity Console, see the <i>Sterling Selling and Fulfillment Foundation: Global Inventory Visibility User Guide</i> . |
| Region Schema | Select the region schema to associate with this resource pool. |
| Capacity Information Available | Select this to indicate that this resource pool can be considered as a source for capacity. If you do not select this, infinite capacity is considered. |
| Capacity Maintained At Resource Level | Select to indicate that capacity is maintained at the resource level for this resource pool. |
| Item Group | |
| Provided Service | Select this if you are configuring a resource pool for one or more provided services. |
| Delivery Service | Select this if you are configuring a resource pool for one or more delivery services. |
| Capacity UOM | Select the unit of measure of the capacity you want to use to for the resource pool. For more information about defining unit of measure for a capacity, see the <i>Catalog Management: Configuration Guide</i> . |
| Node | Select the node to associate with the resource pool. |

4. Choose the **Save** icon. The Resource Pool Details window displays.
5. Enter information in the applicable fields. Refer to Table 27 for field value descriptions.
6. Choose the **Save** icon.

Table 27. Resource Pool Details Window

| Field | Description |
|------------------|---|
| Resource Pool ID | The name of the resource pool as you want it to appear. |
| Node | The node you want to associate with the resource pool. |

Table 27. Resource Pool Details Window (continued)

| Field | Description |
|---------------------------------------|---|
| Capacity Organization | The organization that maintains the capacity within this resource pool. |
| Capacity UOM | The capacity unit of measure you want to use to for the resource pool. |
| Item Group | Indicates whether the resource pool is for delivery services or provided services. |
| Resource Pool Description | Enter a brief description of the resource pool. |
| Service Slot Group | Select the service slot group you want to use to determine service promising. For more information about configuring service slot groups, see “Defining Slot Groups” on page 56. |
| Supervisor ID | <p>Select the supervisor ID from the drop-down list.</p> <p>When determining the supervisor to use for a service work order, Sterling Selling and Fulfillment Foundation looks for the supervisor ID for a given node and seller organization combination, and then for the resource pool supervisor, and if no supervisor is found in either case, the default supervisor for the node.</p> <p>Click the Create New icon to add a new user that can be selected as a supervisor. The user is added in the context of the node that is being configured. For more information about configuring users, see the <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i>.</p> <p>For more information about configuring service supervisors, see the <i>Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide</i>.</p> |
| Capacity Information Available | Select Capacity Information Available to indicate that this resource pool can be considered as a source for capacity. |
| Capacity Maintained At Resource Level | <p>Select this field to indicate that capacity is maintained at the service resource level for this resource pool.</p> <p>Capacity can only be maintained at the service resource level if the resource pool measures capacity in a time-based unit of measure.</p> <p>Note: This feature has been designated for a future release.</p> |
| Calendar | |
| Use Nodes Calendar | Select Use Nodes Calendar, if you want to use the shipping calendar defined for the node in Participant Modeling to determine resource availability. For more information about configuring calendars and defining a node's calendar, see the <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i> . |
| Choose Calendar | Select Choose Calendar if you want to override the node's defined shipping calendar with different calendar to determine resource availability. For more information about defining calendars, see the <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i> . |
| Calendar | If you selected Choose Calendar, select the applicable calendar to use to determine resource availability. The calendars of the node as well as the calendars of the primary enterprise of the node display in the drop-down list. |
| Service Skills | |

Table 27. Resource Pool Details Window (continued)

| Field | Description |
|---|--|
| Service Skill ID | The Service Skill ID |
| Description | A description of the service skill |
| Provider Contact Address | |
| The contact address of the organization that provides this resource pool. | |
| Regions | |
| Service Specific Regions | Select Service Specific Regions to be able to define particular service regions from the Regions table. |
| Service All Regions | Select Service All Regions to indicate that this resource pool can service all regions configured in the region usage. |
| By Regions Only | Select By Regions Only if this resource pool services the specified regions for all time slots throughout the selected days. Note: If you select this radio button, you cannot select the By Regions And Slots radio button unless you delete all the region associations to this resource pool first. |
| By Regions And Slots | Select By Regions And Slots if this resource pool only services specified time slots throughout the day. Note: If you select this radio button, you cannot select the By Regions Only radio button unless you delete all the region associations to this resource pool first. |
| Regions | A list of regions serviced by the resource pool. For information about adding regions to a resource pool, see “Adding a Region Serviced by the Resource Pool” on page 63. For information about removing regions from a resource pool, see “Removing a Serviced Region from a Resource Pool” on page 64. |
| Resources | |
| Resource ID | The Resource ID. |
| Calendar ID | The Calendar ID of the calendar the resource is using. |
| Notes | Any additional notes on this resource. |
| Additional Capacity Limits | |
| Capacity is Limited By Weight | Check this box to limit the capacity by Weight. Select the appropriate UOM from the Weight UOM drop-down list. |
| Capacity is Limited By Volume | Check this box to limit the capacity by Volume. Select the appropriate UOM from the Volume UOM drop-down list. |

Note: You cannot modify the additional capacity limits configured for the same UOM types. However, modification is possible only if one of the capacity limits are not configured.

Note: For a Provided Service, only nodes belonging to the provider organization are displayed. For a Delivery Service, all nodes are displayed whether it belongs to the provider's node or not.

Adding a Service Skill to a Resource Pool

About this task

You can add multiple service skills to a resource pool. The resource pool can run services only for the specified service skills. For example, a resource pool consisting of a set of plumbers can provide multiple installation services, such as washing machine hookup and refrigerator hookup. The Service skills needed could be multiple installation skills like plumbing and electrical skills.

To add a service skill to a resource pool:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Resource Pools. The Resource Pool Search window displays in the work area.
2. In the Resource Pool Search window, perform the following steps:
 - a. Enter the applicable search criteria and choose the **Search** icon. A list of resource pools displays.
 - b. Choose the appropriate resource pool, and choose the **Details** icon. The Resource Pool Details window is displayed
3. In the Resource Pool Details window, choose the **Create New** icon from the Service skills table. The Service Skills List pop-up window displays.
4. Select the applicable service skills and choose the **Save** icon. The service skill is added to the Service Skills table.

Removing a Service Skill from a Resource Pool

About this task

To remove a service skill from a resource pool, in the Resource Pool Details window, select the applicable service skill from the Service Skills table and choose the **Delete** icon.

Adding a Region Serviced by the Resource Pool

About this task

You can associate regions that define the set of geographical areas a resource pool services. You can also specify the days of week and time of day that the resource pool services a given region as well as specify whether the region is the resource pool's primary region. If a region is not marked as a primary region, it is only considered if all other resource pools that may have been configured with the region marked as a primary region have no available capacity. For more information about configuring regions and region schemas, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

Note: If you are configuring a resource pool for delivery services, the regions you add must belong to the region schema you associated with delivery services for resource pool usage.

Note: If you are configuring a resource pool for provided services, the regions you add must belong to the region schema you associated with provided services for resource pool usage.

Note: For more information about associating region schemas for resource pool usage, see “Defining Region Usage for Resource Pools” on page 56.

To add a region to a resource pool:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Resource Pools. The Resource Pool Search window displays in the work area.
2. In the Resource Pool Search window, perform the following steps:
 - a. Enter the applicable search criteria and choose the **Search** icon. A list of resource pools displays.
 - b. Choose the appropriate resource pool, and choose the **Details** icon. The Resource Pool Details window is displayed
3. In the Resource Pool Details window, select the By Regions radio button if you would like to add a region for all time slots for a given day, or By Region And Slots if you want to pick the time of day to service a specific region. Click the **Create New** icon from the Regions table. The Region Search pop-up window displays.
4. Enter the applicable search criteria and click the **Search** icon. A list of regions displays.
5. Select the region you want to add to the resource pool and click the **Create New** icon. The region is added to the Regions table in the Resource Pool Details window.
 - If you want to select a child region or browse through the region hierarchy, select the applicable parent region and choose the **Zoom In** icon. In the Explore Region pop-up window select the applicable child region from the region tree and choose the **Create New** icon. You can also navigate within this window to browse through the complete region hierarchy until you find the region you want to select.
6. In the Regions table, select the Primary check box if the resource pool is a primary provider of the region and select the check boxes of the days of the week for each time slot, if applicable, when the resource pool can fulfill a service to the region.

Removing a Serviced Region from a Resource Pool

About this task

To remove a serviced region from a resource pool, in the Resource Pool Details window, select the applicable region from the regions table and choose the **Delete** icon.

Creating and Modifying a Service Resource

About this task

A *service resource* in Sterling Selling and Fulfillment Foundation is used to define one or more people that work as a team, and is represented as a single entity, that can perform provided or delivery services. Each resource has a calendar associated with it by either selecting one of the calendars defined by the node of the resource pool it belongs to, or using a node's shipping calendar. Services resources are associated with a single resource pool.

You can also associate team members with a service resource that can be selected to perform work order tasks in the Application Console.

To create a resource for a resource:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Resource Pools. The Resource Pool Search window displays in the work area.
2. In the Resource Pool Search window, perform the following steps:
 - a. Enter the applicable search criteria and choose the **Search** icon. A list of resource pools displays.
 - b. Choose the appropriate resource pool, and choose the **Details** icon. The Resource Pool Details window is displayed
3. In the Resource Pool Details window, choose the **Create New** icon from the Resources table. The Service Resources Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 28 for field value descriptions.

Table 28. Service Resource Details Window

| Field | Description |
|---|---|
| Resource ID | The name of the Resource as you want it to appear. |
| Calendar | |
| Use Node's Calendar | Select this option if the service resource being configured should use the shipping calendar of the node the resource pool is associated with. |
| Use Calendar | Select this option if the service resource being configured should use a calendar that is not the node's shipping calendar. |
| Calendar | Select a calendar for the service resource to use. The calendars of the node as well as the calendars of the primary enterprise of the node display in the drop-down list. |
| Contact Address | The service resource's contact address. Choose the Details icon to enter an address. Choose the Contact Info tab to view additional contact information. |
| Team Members | |
| Use this inner panel to add team members to your service resource. These team members are added to work order appointments when this resource is selected to run the appointment. <ul style="list-style-type: none"> • Click the Add User icon to add team members to the service resource. • Click the Remove User icon to remove the selected team members from the service resource. | |
| User ID | The identifier of the team member associated with this resource. |
| User Name | The name of the team member associated with this resource. |
| Notes | |
| Add any additional notes here. | |

5. Choose the **Save** icon.

Deleting a Service Resource

About this task

To remove a service resource from a resource pool, in the Resource Pool Details window, select the applicable service resource from the resources table and choose the **Delete** icon.

Deleting a Resource Pool

About this task

To delete a resource pool:

Procedure

1. From the tree in the application rules side panel, choose Resource Capacity > Resource Pools. The Resource Pool Search window displays in the work area.
2. Enter the applicable search criteria and choose the **Search** icon. A list of resource pools displays.
3. Select the applicable resource pool and choose the **Delete** icon.

Chapter 9. Configuring Value Added Services

Value Added Services (VAS) are performed to meet customer demands. Different types of VAS activities include:

- Monogramming on a T-shirt
- Building computer to buyer's specification
- Segregating individual boxes from larger boxes
- Stain guarding a sofa
- Installing a PC at home

As the above examples suggest, Value Added Services related activities apply both before and after the shipping process.

To support supplying Value Added Services, there may be a Value Added Service area in the warehouse, where Value Added Services are performed. These activities can include packaging, monogramming and ticketing. The retrieval and rules are defined further in this section.

Value Added Services: Defining Activities

Activities in the warehouse are grouped into Activity Groups. Examples of activity groups include Receiving, VAS, Picking and Packing.

Each activity group is associated with activity codes that define each activity that is performed. For example, receiving activity group has pre-receiving, quality control and receiving 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 29 for field value descriptions.
3. Choose the **Save** icon.

Table 29. 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 29 on page 67 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.

Defining Work Order Cancellation Reasons

A reason code is associated with cancellation of work order requests. When cancelling a work order, a Work Order Cancellation Reason must be supplied; therefore at least a default value for work order cancellations should be configured.

Note: Viewing of Work Order Cancellation Reason Code is available only for Enterprise and Node users.

Creating a Work Order Cancellation Reason

About this task

To create a work order cancellation reason:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > Work Order Cancellation Reasons. The Work Order Cancellation Reasons window displays.
2. In the Work Order Cancellation Reasons window, choose the **Create New** icon.
3. The Cancellation Reason Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 30 on page 69 for field value descriptions.
5. Choose the **Save** icon.

Table 30. Cancellation Reason Code Details Pop-up Window

| Field | Description |
|----------------------|--|
| Cancellation Reasons | Enter a code for the work order cancellation reason. |
| Short Description | Enter a short description for the work order cancellation reason code. |
| Long Description | Enter a long description for the work order cancellation reason code. |

Creating a New Work Order Cancellation Reason from an Existing Work Order Cancellation Reason

About this task

To create a new Work Order Cancellation Reason from an existing Work Order Cancellation Reason:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > Work Order Cancellation Reasons. The Work Order Cancellation Reasons window displays with the list of Work Order Cancellation Reason Codes.
2. Choose the Work Order Cancellation Reason to be copied.
3. Choose the **Save As...** icon. The Cancellation Reasons Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 30 for field value descriptions.
5. Choose the **Save** icon.

Modifying a Work Order Cancellation Reason

About this task

Once a Work Order Cancellation Reason has been created, it can be modified.

To modify a Work Order Cancellation Reason:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > Work Order Cancellation Reasons. The Work Order Cancellation Reasons window displays with the list of Work Order Cancellation Reason Codes.
2. Choose the Work Order Cancellation Reason to be modified.
3. Choose the **Details** icon. The Cancellation Reason Details pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 30 for field value descriptions.
5. Choose the **Save** icon.

Deleting a Work Order Cancellation Reason

About this task

To delete a Work Order Cancellation Reason Code:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > Work Order Cancellation Reasons. The Work Order Cancellation Reasons window displays with the list of Work Order Cancellation Reason Codes.
2. Choose the Work Order Cancellation Reason to be deleted.
3. Choose the **Delete** icon.

Defining Allocation Considerations

A work order for service items belonging to one of the following service item group codes would result in inventory transformations on confirmation:

- KIT - Kitting
- DKIT - De-kitting
- COMPL - Compliance
- INVC Inventory-Change

The Allocation Considerations configuration is used by Sterling Selling and Fulfillment Foundation to provide visibility into such inventory transformations. For instance:

- When a work order is created, demand is placed against the original inventory (the one being consumed), and supply is increased for the new inventory (the one being created).

Note: The supply being increased is not an onhand supply. It is an indicative supply that would be available in future.

Note: The demands being increased are not promised demands.

- When a work order is allocated, the demands placed are modified to indicate that the demands are promised. The supplies may also be modified to indicate their increased chance of arrival.

Note: These demands and supplies could be utilized to assess the availability of inventory.

- When a work order is confirmed, the supply for the original inventory is removed and supply for the new inventory is created.

Creating a Work Order Allocation Consideration

About this task

To create a work order allocation consideration:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > Allocation Considerations. The Work Order Allocation Considerations window displays.
2. In the Work Order Allocation Considerations window, choose the **Create New** icon. The Work Order Allocation Consideration pop-up window displays.
3. Enter information in the applicable fields. Refer to Table 31 on page 71 for field value descriptions.
4. Choose the **Save** icon.

Table 31. Work Order Allocation Consideration Pop-up Window

| Field | Description |
|---------------------------------|---|
| Service Item Group | Select the service item group for which the allocation considerations are being created. |
| Demand Type for Consumed | Associate the type of demand to identify inventory consumed to work orders for the service item group. |
| Supply Type for Created | Associate the type of supply to identify inventory created by the work order for the service item group. Typical values for supply type are 'ONHAND'. |
| Demand Type for Allocated | Associate the type of demand to identify inventory allocated to work orders for the service item group. |
| Supply Type for Allocated | Associate the type of supply to identify inventory allocated by the work order for the service item group. |
| Actual Supply Type for Consumed | Associate the actual type of demand to identify inventory allocated to work orders for the service item group. |
| Actual Supply Type for Created | Associate the actual type of supply to identify inventory created by the work order for the service item group. |

Modifying a Work Order Allocation Consideration

About this task

Once a work order allocation consideration has been created, it can be modified.

To modify a work order allocation consideration:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > Allocation Considerations. The Work Order Allocation Considerations window displays with the list of Work Order Allocation Considerations.
2. Choose the Work Order Allocation Consideration to be modified.
3. Choose the **Details** icon. The Work Order Allocation Consideration pop-up window displays.
4. Enter information in the applicable fields. Refer to Table 31 for field value descriptions.
5. Choose the **Save** icon.

Deleting a Work Order Allocation Consideration

About this task

To delete a work order allocation consideration:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > Allocation Considerations. The Work Order Allocation Considerations window displays with the list of Work Order Allocation Considerations.
2. Choose the Work Order Allocation Consideration to be deleted.
3. Choose the **Delete** icon.

Defining Value-Added Services Modification Groups

Most order document types flow through a pipeline without requiring any intervention by a customer service representative. However, there are times when modifications are required, such as modifying quantity or activities required. Sterling Selling and Fulfillment Foundation supports modifications through the Application Console and APIs. It is critical to decide which modifications are allowed for each modification type, modification level, and status combination.

Note: Contemplate business and system integration implications before allowing a modification that is disallowed as part of the system defaults.

You can configure rules and components used when determining what parts of value-added services can be modified as well as when in the value-added services lifecycle the modifications can be performed. The set of rules and components can be classified into modification groups. Each modification group can be associated with a user-defined condition. For more information about modification rules, see the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

Setting Up a Value-Added Services Modification Group

About this task

To set up a VAS modification group:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > VAS Modification Groups. The Status Modification Group List window displays.
2. From the Status Modification Group List, click the **Plus** icon. The Status Modification Group Details window displays.
3. Enter information in the applicable fields. For field value descriptions, see the following table.
4. Click the **Save** icon to save the modification group.

Table 32. Status Modification Group Details Window

| Field | Description |
|--|--|
| Status Modification Group ID | Enter a name for the status modification group. |
| Description | Enter a description for the status modification group. |
| Inherited From Status Modification Group | (Optional) Select a parent status modification group if you want to inherit modification rules from another group. |
| Is Override | (Optional) Check this option if you want this status modification group to take precedence. |

Table 32. Status Modification Group Details Window (continued)

| Field | Description |
|--------------------------------------|--|
| Condition | (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 Plus icon, which opens the Condition Detail window. Refer to “Adding Conditions to a Status Modification Group” on page 75 for information about configuring conditions for status modification groups. Note: If you do not specify a condition, this group's validation is enforced. |
| Subscribed Modification Types | |
| Available | 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 Right Arrow icon. |
| Subscribed | 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 Left Arrow icon. Click the Save icon to display the subscribed modification types in the Modification Rules tab. |
| Modification Rules | |
| Group By | Select a sort option for displaying modification types in the Primary Info tab. You can sort modification types by type, level, and status. |
| Primary Info | 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. Refer to Table 33 for field value descriptions. |

Results

The following table describes modifications types that are available for value-added services:

Table 33. Modification Rules Tab, Primary Information

| Field | Description |
|---------------------|-------------|
| Primary Info | |

Table 33. Modification Rules Tab, Primary Information (continued)

| Field | Description |
|----------------------------|---|
| Add Activity | Allow, disallow, or ignore the addition of an activity appropriately. |
| Add Instruction | Allow, disallow, or ignore the addition of an instruction appropriately. |
| Add Note | Allow, disallow, or ignore the addition or modification of notes associated with a work order. |
| Cancel Work Order | Allow, disallow, or ignore the cancellation of a work order appropriately. |
| Change Appointment | Allow, disallow, or ignore the addition, deletion, or modification of a work order appointment. |
| Change Appointment User | Allow, disallow, or ignore the addition, deletion, or modification of users assigned to complete an appointment in the work order. |
| Change Finish Date | Allow, disallow, or ignore the modification of a finish date by which the tasks should be finished. |
| Change Instruction | Allow, disallow, or ignore the modification of an instruction appropriately. |
| Change Other Attributes | Allow, disallow, or ignore the modification of other attributes appropriately. |
| Change Other Relationships | Allow, disallow, or ignore the modification of other relationships appropriately. |
| Change Product Line | Allow, disallow, or ignore the addition, deletion, or modification of the deliverable product line associated with a work order. |
| Change Service Line | Allow, disallow, or ignore the addition, deletion, or modification of the Service Line associated with a work order. |
| Change Ship To | Allow, disallow, or ignore the changes made to the service location address of the customer, where the service is performed if the item is a Provided Service Item. |
| Change Start Date | Allow, disallow, or ignore the modification of a start date appropriately. |
| Change Station Assignments | Allow, disallow, or ignore the modification of station assignments appropriately. |
| Decrease Priority | Allow, disallow, or ignore the decrease in priority appropriately. |
| Hold Type Modifications | Allow, disallow, or ignore the modification of the Hold Type with which a work order is associated. |
| Increase Priority | Allow, disallow, or ignore the increase in priority appropriately. |
| Remove Activity | Allow, disallow, or ignore the removal of an activity appropriately. |
| Resequence Activity | Allow, disallow, or ignore the resequencing of an activity appropriately. |
| Activity | Allow, disallow, or ignore the modification types at the activity level appropriately. |
| Work Order | Allow, disallow, or ignore the modification types at the work order level appropriately. |

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

| Option | Description |
|--|---|
| Class Name (if Dynamic is checked) | Enter the class name that implements the following Java interface: com.yantra.ycp.japi.YCPDynamicCondition Note: 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> . |
| Condition Properties (if Dynamic is checked) | 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> . |

Value-Added Services: Modifying a Modification Group

About this task

To modify a VAS modification group:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > VAS Modification Groups. The Status Modification Group List window displays.
2. From the Status Modification Group List, select the applicable modification group and click the **Edit** icon. The Status Modification Group Details window displays.
3. Enter information in the applicable fields. Refer to Table 32 on page 72 for field value descriptions.
4. Click the **Save** icon.

Value-Added Services: Deleting a Modification Group

About this task

To delete a VAS modification group:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > VAS Modification Groups. The Status Modification Group List window displays.
2. From the Status Modification Group List, select the applicable modification group and click the **Delete** icon.

Defining Value Added Services Process Type Details

Value Added Services Process Type Details define parameters and templates that distinguish a process type.

A **process type pipeline** is a series of transactions and statuses that guide document types, such as a Value Added Services execution, 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*.

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

Viewing Value Added Services Process Type Details

About this task

To view Value Added Services process type details:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > VAS Process Type Details. The Process Type Details: Work Order window displays.
2. Primary information of the Process Type displays in the applicable fields. Refer to Table 34 for field value descriptions.

Results

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

Table 34. Process Type Details: Work Order Window

| Field | Description |
|---------------------|--|
| Primary Info | |
| Process Type | This is automatically populated by the system as "WO_VAS". |
| Process Type Name | This indicates the name of the process type. |
| Description | This provides a brief description for the process type. |

Defining the Value Added Services Process Model

The Value Added Services process is modeled through a pipeline. This represents the process configuration that is unique to a warehouse. A warehouse may also specify unique processes for each participating enterprise.

For example, a warehouse that performs ticketing for outbound shipments after pick and pack into a carton is complete.

Value Added Services Process Model: 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.

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

Value Added Services Process Model: Hub Rule

When you expand the Pipeline Determination branch, the display 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 all user created determination rules (For example, My Rule) components display. Double-click on the applicable rule 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.

For more information about Pipeline Determination and Hub Rule, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

Value Added Services Process Model: Pipelines

About this task

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

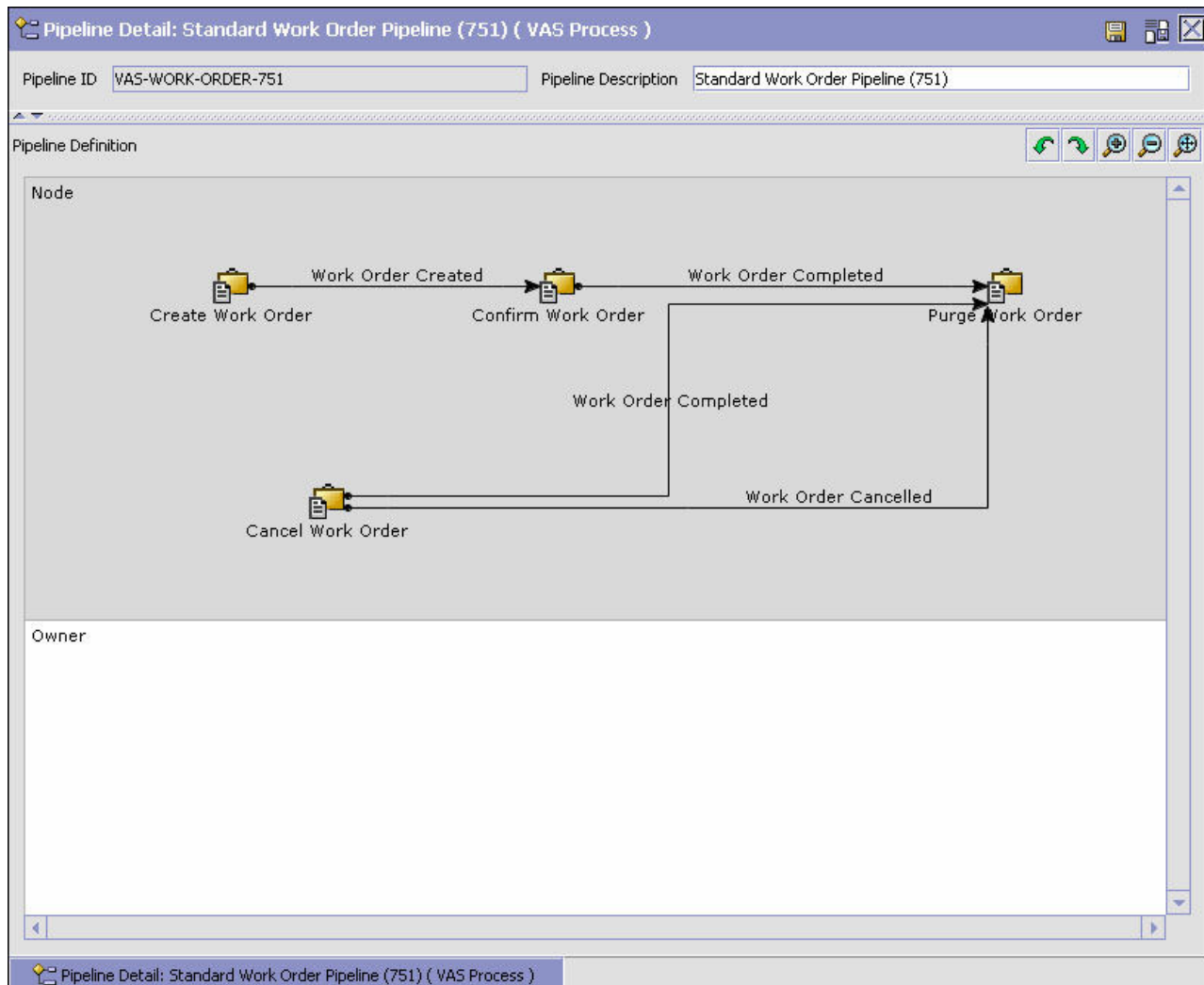
To view the Value Added Services pipeline details:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > VAS Process Model. The VAS Process window displays.
2. In the VAS Process window, choose VAS Process Repository > Pipelines > Standard Work Order Pipeline.
3. The Pipeline Detail: Standard Work Order Pipeline (VAS Process) window displays.

Results

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



Value Added Services Process Model: 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 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 Value Added Services pipeline:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > VAS Process Model. The VAS Process window displays.
2. In the VAS Process window, choose the **Transactions** icon.
3. The Transactions tab window displays. Refer to Table 35 for field value descriptions.

Results

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

Table 35. Work Order VAS Pipeline - Transactions Tab Window

| Field | Description |
|---------------------|---|
| Allocate Work Order | This transaction represents allocation of the work order for the Value Added Services process. |
| Cancel Work Order | This transaction represents the cancellation of a work order created for Value Added Services. |
| Confirm Work Order | This transaction represents that the work order needs to be confirmed for Value Added Services. |
| Create Work Order | This transaction represents creation of a work order for Value Added Services. |
| Purge Work Order | This transaction represents the purge of work orders created for Value Added Services. |
| Release Work Order | This transaction represents the release of work orders created for Value Added Services. |

Value Added Services Process Model: 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 Value Added Services pipeline:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > VAS Process Model. The VAS Process window displays.
2. In the VAS Process window, choose the **Statuses** icon.
3. The Statuses tab window displays. Refer to Table 36 on page 81 for field value descriptions.

Results

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

Table 36. Work Order VAS Pipeline - Statuses Tab Window

| Field | Description |
|------------------------------------|--|
| Work Order Created | This indicates that a work order is created. This corresponds to the first step of the 'Create Work Order' transaction. |
| Work Order With Components Created | This indicates that components items required have been added to the work order. This corresponds to completion of the 'Create Work Order' transaction. |
| Work Order Allocated | This indicates that allocation process is completed for the work order. This corresponds to 'Allocate Work Order' transaction. |
| Work Order Confirmed | This indicates all activities required for the work order is complete. This corresponds to the 'Confirm Work Order' transaction. This transaction creates inventory for the parent item on the work order. Putaway process for the finished inventory to storage or to packing or shipping zones can be initiated. |
| Work Order Canceled | This indicates cancellation of the Work Order for the shipment. This corresponds to the 'Cancel Work Order' transaction. |

Value Added Services Process Model: 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 Value Added Services pipeline:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > VAS Process Model. The VAS Process window displays.
2. In the VAS Process window, choose the **Conditions** icon.
3. The Conditions tab window displays. Refer to Table 37 on page 82 for field value descriptions.

Results

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

Table 37. Work Order VAS Pipeline - Conditions Tab Window

| Field | Description |
|---------------|---|
| HasComponents | Condition that evaluates if the 'HasComponents' field is 'Y' for a Work Order for Value Added Services. |

Value Added Services Process Model: 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 a Value Added Services pipeline:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > VAS Process Model. The VAS Process window displays.
2. In the VAS Process window, choose the **Actions** icon.
3. The Actions tab window displays. Refer to Table 38 for field value descriptions.

Table 38. Work Order VAS Pipeline - Action Tab Window

| Field | Description |
|-----------|---|
| Templates | Default templates are provided for: Publish Data – Send data to external queue or internal tables. Raising an Exception – Raise an alert using the Event Management from the published information. Send Email – Raise an email action utilizing a template to format from the published information. Send Email-HTML format – Raise an email action to create an HTML email format from the published information. |

Value Added Services Process Model: Service Definitions

About this task

Service definitions are a representation of the logic that regulates document workflow services. The Service Builder is a graphical interface that enables you to create a graphical representation of these *services*.

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

To view the service definition details of a Value Added Services pipeline:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > VAS Process Model. The Work Order VAS window displays.
2. In the Work Order VAS window, choose the **Service Definitions** icon.
3. The Service Definitions tab window displays.

Table 39. Work Order VAS Pipeline - Service Definition Tab Window

| Field | Description |
|---------------------|---|
| Service Definitions | Displays service definitions that are specific to the VAS pipeline, if any. |

Defining Purge Criteria

Transactional data collected by Sterling Selling and Fulfillment Foundation during the execution are periodically removed from the 'live' transactional tables. It is common to retain order related information for extended period of time. There are history tables provided for relevant transactional tables to move data from the day-to-day 'live' tables to a historical table.

Purges are 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.

Setting Up Purge Criteria

About this task

To set up purge criteria:

Procedure

1. From the tree in the application rules side panel, choose VAS > VAS Process > Purge Criteria. The Purge Criteria List window displays.
2. In the Purge Criteria List window, choose the **Details** icon. The Purge Criteria Details pop-up window displays.
3. Enter information in the applicable fields. Refer to Table 40 for field value descriptions.
4. Choose the **Save** icon.

Table 40. Purge Criteria Details Pop-up Window

| Field | Description |
|------------------|--|
| Purge Code | Identifies a purge program. This is a system defined code. |
| Description | Description of the 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. |

Table 40. Purge Criteria Details Pop-up Window (continued)

| Field | Description |
|-------------------|---|
| Retention Days | <p>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.</p> <p>The inventory purge does not take retention days into account when purging.</p> |
| Write To Log File | <p>Check this box if you want purged data written to a log. The log can be backed up and used as a journal at a later date.</p> |
| Log File Name | <p>Enter a log file name. This is applicable only if 'Write To Log File' is checked. This file consists records of the specific table that is purged.</p> <p>The log file is created in the directory specified in the <code>yfs.purge.path</code> property. If this is not passed, it defaults to the value specified in the <code>yfs.properties</code> file. If a variable is introduced, then the <code>yfs.purge.path</code> is ignored. To override this property, add an entry for it in the <code><INSTALL_DIR>/properties/customer_overrides.properties</code> file. For additional information about overriding properties using the <code>customer_overrides.properties</code> file, see the <i>Sterling Selling and Fulfillment Foundation: Properties Guide</i>.</p> <p>For more information about using variables for the log file directory, see 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 10. Configuring Count

Count requests are initiated through the console on an ad hoc basis. System events like exception being recorded during putaway, retrieval or pick, or location quantity dropping below minimum levels or to zero can be used to initiate a count request. A request also includes date and time parameters indicating the start and end time expected for the task.

For example, a user may request a count for a zone that has slow moving items to start the next day by assigning a low priority.

Count Requests may also be generated on a regular basis to consistently maintain inventory accuracy.

A count request is also created through the Event Management and inventory monitors, when inventory at a location reaches zero quantity or when minimum or maximum inventory levels are breached.

Defining Count Program

Sterling Selling and Fulfillment Foundation is equipped to automatically generate a count request using the Automatic Count Generation functionality for all nodes, a specific node, or all nodes in a region.

The Automatic Count Generation is set up through Count Programs, which define the valid date range of the program, the applicable zones in the warehouse, the products that are to be counted, and the periodicity of the count requests.

The Count Program is associated with a count calendar that would provide information about the list of working days, when the relevant nodes would perform count for this program.

Creating a Count Program

About this task

To create a count program:

Procedure

1. From the tree in the application rules side panel, choose Count > Count Program. The Count Program List window displays.
2. In the Count Program List window, choose the **Create New** icon. The Count Program Details window displays.
3. Enter a valid Program Name for the Program being created.
4. Choose the relevant count program span for the Program.
 - Choose 'All Nodes,' if the program is to span all the nodes.
 - Choose 'Specific Node' and the specific node from the drop-down list, if the program is to span a specific node.
 - Choose 'All Nodes in Region' and choose the **Search** icon to select the relevant region from the Regions pop-up window, if the program is to span

all nodes in a region. For more information about configuring region schemas, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

5. Choose the relevant Calendar to be used for the Program, from the drop-down list. The calendar of the node as well as the calendars of the primary enterprise of the node display in the drop-down list. For more information about creating a new calendar, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.
6. Choose the **Save** icon. The Count Program Conditions panel now displays in the Count Program Details window.
7. In the Count Program Conditions panel, choose the **Create New** icon. The Count Program Condition Details Pop-up displays.
8. Enter information in the applicable fields. Refer Table 41 for field value descriptions.
9. Choose the **Save** icon. The pop-up window is closed, and you are returned to the Count Program Details window.
10. After setting up all the relevant Count Program Conditions, choose the **Save** icon in the Count Program Details window.

Table 41. Count Program Condition Details Pop-up Window

| Field | Description |
|---|--|
| Description | Enter a description for the count program condition. |
| Count Program Condition | |
| Select locations to be counted in the following zones | Choose the Details icon. In the List of Values pop-up, choose the zones that are to be counted. |
| Select Product Classes whose inventory is to be counted | Choose the Details icon. In the List of Values pop-up, choose the product classes that are to be counted. |
| Item Classifications | Displays the item classifications that are available to be counted. |
| Count at Classification Level | Select if the counting is to be done at the item classification level. |
| Count inventory when unit value is between | Enter the minimum unit value to be counted. This defaults to 0 if no value is provided. |
| and | Enter the maximum unit value to be counted. This defaults to 99,999 if no value is provided. |
| Number of times to count | Enter the number of times to count. |

Modifying a Count Program

About this task

To modify a Count Program:

Procedure

1. From the tree in the application rules side panel, choose Count > Count Program. The Count Program List window displays with the list of existing programs.
2. Choose the Count Program that is to be modified. Choose the **Details** icon.
3. The Count Program Details window displays. Modify the entries as necessary.

4. Choose the **Save** icon.

Deleting a Count Program

About this task

To delete a count program:

Procedure

1. From the tree in the application rules side panel, choose **Count > Count Program**. The **Count Program List** window displays with the list of existing programs.
2. Choose the **Count Program** that is to be deleted.
3. Choose the **Delete** icon.

Viewing Region Usage for Count

About this task

A region schema is the complete hierarchical set of regions that define a given geography. A region is configured as a specific territory. For example, you can create a region for a complete state, city, or town. For more information about configuring region schemas, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

You can view the region schemas used for configuring count programs. For more information about count programs, see “Defining Count Program” on page 85.

To view region usage for count:

Procedure

1. From the tree in the application rules side panel, choose **Count > Region Usage For Count**. The **Region Usage For Count** pop-up window displays.
2. From **Region Schema for Count**, select the region schema you want to view the details for.
3. Choose the **View Details** icon. The **Region Schema Details** pop-up window displays.
4. Choose the **Save** icon.

Defining Corporate Count Request Cancellation Reasons

A reason code is associated with cancellation of corporate count requests.

Note: Viewing of Corporate Count Request Cancellation Reason Code is available only for Enterprise and Node users.

Creating a Corporate Count Request Cancellation Reason

About this task

To create a corporate count request cancellation reason:

Procedure

1. From the tree in the application rules side panel, choose Count > Corporate Count Request Cancellation Reasons. The Corporate Count Request Cancellation Reason Code window displays.
2. In the Corporate Count Request Cancellation Reason Code window, choose the **Create New** icon.
3. The Corporate Count Request Cancellation Reason Code Details pop-up window displays.
4. Enter information in the applicable fields. Refer Table 42 for field value descriptions.
5. Choose the **Save** icon.

Table 42. Corporate Count Request Cancellation Reason Code Details Pop-up Window

| Field | Description |
|--|---|
| Corporate Count Request Cancellation Reason Code | Enter a code for the corporate count request cancellation reason. |
| Short Description | Enter a short description for the corporate count request cancellation reason code. |
| Long Description | Enter a long description for the corporate count request cancellation reason code. |

Creating a New Corporate Count Request Cancellation Reason from an Existing Corporate Count Request Cancellation Reason

About this task

To create a new corporate count request cancellation reason from an existing corporate count request cancellation reason:

Procedure

1. From the tree in the application rules side panel, choose Count > Corporate Count Request Cancellation Reasons.
2. The Corporate Count Request Cancellation Reason Codes window displays with the list of Corporate Count Request Cancellation Reasons.
3. Choose the Corporate Count Request Cancellation Reason to be copied.
4. Choose the **Save As...** icon. The Corporate Count Request Cancellation Reason Code Details pop-up window displays.
5. Enter information in the applicable fields. Refer Table 42 for field value descriptions.
6. Choose the **Save** icon.

Modifying a Corporate Count Request Cancellation Reason

About this task

Once a Corporate Count Request Cancellation Reason has been created, it can be modified.

To modify a corporate count request cancellation reason:

Procedure

1. From the tree in the application rules side panel, choose Count > Corporate Count Request Cancellation Reasons.
2. The Corporate Count Request Cancellation Reason Codes window displays with the list of Corporate Count Request Cancellation Reasons.
3. Choose the Corporate Count Request Cancellation Reason to be modified.
4. Choose the **Details** icon. The Corporate Count Request Cancellation Reason Code Details pop-up window displays.
5. Enter information in the applicable fields. Refer Table 42 on page 88 for field value descriptions.
6. Choose the **Save** icon.

Deleting a Corporate Count Request Cancellation Reason

About this task

To delete a corporate count request cancellation reason code:

Procedure

1. From the tree in the application rules side panel, choose Count > Corporate Count Request Cancellation Reasons.
2. The Corporate Count Request Cancellation Reason Codes window displays with the list of Corporate Count Request Cancellation Reasons.
3. Choose the Corporate Count Request Cancellation Reason to be deleted.
4. Choose the **Delete** icon.

Defining Corporate Count Request Purge Criteria

Transactional data collected by Sterling Selling and Fulfillment Foundation during the execution are periodically removed from the 'live' transactional tables. It is common to retain order related information for extended period of time. There are history tables provided for relevant transactional tables to move data from the day-to-day 'live' tables to a historical table.

Purges are 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.

Setting Up Corporate Count Request Purge Criteria

About this task

To set up purge criteria:

Procedure

1. From the tree in the application rules side panel, choose Count > Corporate Count Request Purge Criteria. The Purge Criteria List window displays.
2. In the Purge Criteria List window, choose the **Details** icon. The Purge Criteria Details pop-up window displays.
3. Enter information in the applicable fields. Refer Table 43 on page 90 for field value descriptions.
4. Choose the **Save** icon.

Table 43. Purge Criteria Details Pop-up Window

| Field | Description |
|-------------------|---|
| Purge Code | Identifies a purge program. This is a system defined code. |
| Description | Description of the purge. |
| Rollback Segment | <p>Defines the rollback segment that should be explicitly used for the purge transaction qualified by the purge code.</p> <p>This is useful when there are huge logical data sets that have to be purged. This is optional and used for order related purges.</p> |
| 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 | Check this box if you want purged data written to a log. The log can be backed up and used as a journal at a later date. |
| Log File Name | <p>The log file is created in the directory specified in the <code>yfs.purge.path</code> property. If this is not passed, it defaults to the value specified in the <code>yfs.properties</code> file. If a variable is introduced, then the <code>yfs.purge.path</code> is ignored. To override this property, add an entry for it in the <code><INSTALL_DIR>/properties/customer_overrides.properties</code> file. For additional information about overriding properties using the <code>customer_overrides.properties</code> file, see the <i>Sterling Selling and Fulfillment Foundation: Properties Guide</i>.</p> <p>For more information about using variables for the log file directory, see the <i>Sterling Selling and Fulfillment Foundation: Configuration Guide</i>.</p> <p>For information about filename limitations related to internationalization, see the <i>Sterling Selling and Fulfillment Foundation: Localization Guide</i>.</p> |

Chapter 11. Synchronizing with Node Inventory

Sterling Selling and Fulfillment Foundation provides the functionality to reconcile its internal inventory picture with the actual inventory picture at the nodes.

Loading the Inventory Picture from a Node

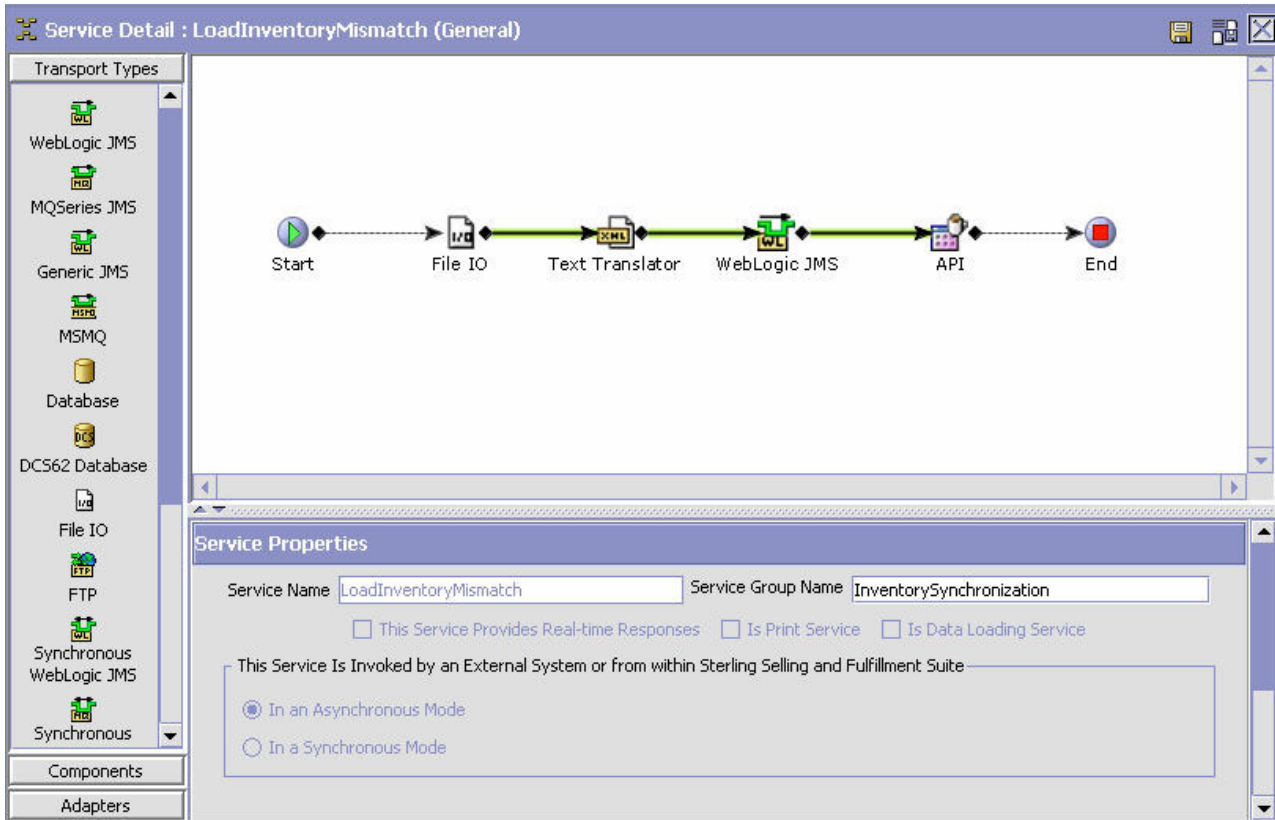
About this task

In order to reconcile internal and node inventory pictures, Sterling Selling and Fulfillment Foundation first needs to successfully load the node's inventory picture.

To achieve this, you need to use the LoadInventoryMismatch Service in the Service Definition Framework:

Procedure

1. From the Applications menu of the Applications Manager, click Application Platform.
2. From the tree in the application rules side panel, double-click Process Modeling.
3. Click the General tab. In the Process Types swimlane, right-click the General process type, and click Model Process. The Repository Details window and work area display for the General process type.
4. Click the Service Definitions tab.
5. Expand the InventorySynchronization branch.
6. Right-click LoadInventoryMismatch, and click details. The Service Detail window displays in the work area.



The LoadInventoryMismatch Service

The LoadInventoryMismatch service goes through several different steps.

Note: Although we have used WebLogic JMS as an example, Sterling Selling and Fulfillment Foundation supports the use of WebSphere® and JBoss MQ JMS.

From File IO to Text Translator

The LoadInventoryMismatch service first looks at a particular directory where the node uploads the inventory information in an XML file of the following format:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Inventory YantraMessageGroupID="OPTIONAL" ShipNode="REQUIRED" ApplyDifferences=""
  CompleteInventoryFlag="" ReasonCode="" ReasonText="">
  <Items>
    <Item InventoryOrganizationCode="REQUIRED" ItemID="REQUIRED" ProductClass=""
      UnitOfMeasure="" >
      <Supplies>
        <Supply ETA="" Quantity="" Segment="" SegmentType="" ShipByDate=""
          SupplyType="" SupplyReference="" SupplyReferenceType="" SupplyLineReference=""
          AvailabilityType="">
          <Tag BatchNo="" LotNumber="" LotAttribute1="" LotAttribute2=""
            LotAttribute3="" LotKeyReference="" ManufacturingDate="" RevisionNo="" />
        </Supply>
      </Supplies>
    </Item>
    <Item ...>
      ...
    </Item>
  </Items>
</Inventory>
<Item ...>
```

```
        ...  
    </Item>  
</Items>  
</Inventory>
```

That XML is then passed on to the Text Translator.

Note: There are as many messages sent to the JMS queues as there are <Items> nodes. In order to control the number of those messages, make use of the number of <Item> nodes within each <Items> node appropriately. IBM recommends including 100 <Item> nodes within each <Items> node.

From Text Translator to WebLogic JMS

The Text Translator parses the XML files, one at a time, and includes the attributes of the <Inventory> node into every one of the <Items> node.

If the YantraMessageGroupID attribute is not passed, Sterling Selling and Fulfillment Foundation generate one automatically. That attribute is unique for every file.

When a file has been successfully parsed, each <Items> node is added to the JMS Queue as a message. When the file has been completely parsed, the EOF (End Of File) node is added to the JMS Queue.

From WebLogic JMS to API

The JMS Queue then reads every message in the queue. The loadInventoryMismatch API is called for each message in the queue, with the appropriate XML as input.

When an EOF message is received by the Integration Server, the server first checks if there are any reprocessable messages for this service with the same YantraMessageGroupID. If there are any pending error messages to be reprocessed, then the EOF message is marked as a reprocessable error message and inserted into the YFS_REPROCESS_ERROR table. This message needs to be reprocessed along with other reprocessable errors for this service.

If there is no pending error messages to be processed, it calls the SyncLoadedInventory Service. For more information about the Sync Loaded Inventory service, see "The SyncLoadedInventory Service" on page 94.

The loadInventoryMismatch API inserts into the YFS_INVENTORY_SUPPLY_TEMP for each <Item> ... </Item> node.

Configuring the LoadInventoryMismatch Service

About this task

The LoadInventoryMismatch service is responsible for loading the inventory picture from the node, and eventually populating the YFS_INVENTORY_SUPPLY_TEMP table with the appropriate data.

The following steps are required to configure the service appropriately:

Procedure

1. In the LoadInventoryMismatch Service Detail window, click the green connector in between File IO and Text Translator. The File IO Receiver properties display under the graphic area.
 - Click the Runtime tab.
 - Ensure that the Create EOF Message field is checked.
 - For descriptions of the rest of the fields, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.
 - Click the File tab.
 - In the Incoming Directory field, enter the directory where the node uploads the inventory XMLs.
 - For descriptions of the rest of the fields, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.
2. In the LoadInventoryMismatch Service Detail window, click the green connector in between Text Translator and WebLogic JMS. The JMS Sender properties display under the graphic area.
 - Click the Runtime tab.
 - Ensure that the Queue Name entered is the name of an existing JMS Queue, and that it matches the Queue Name of the JMS Receiver properties.
 - For descriptions of the rest of the fields, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.
3. In the LoadInventoryMismatch Service Detail window, click the green connector in between WebLogic JMS and API. The JMS Receiver properties display under the graphic area.
 - Click the Runtime tab.
 - Ensure that the Queue Name entered is the name of an existing JMS Queue, and that it matches the Queue Name of the JMS Sender properties.
 - For descriptions of the rest of the fields, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.
 - Click the Exception tab.
 - Ensure that the Is Reprocessing field is checked.

Synchronizing the Inventory Tables

When a file containing the node's inventory picture has been successfully loaded into the YFS_INVENTORY_SUPPLY_TEMP table, the differences in the inventory pictures need to be reconciled. This is done through the SyncLoadedInventoryService.

The SyncLoadedInventory Service

The SyncLoadedInventory service calls the syncLoadedInventory API after the node's inventory picture has been successfully loaded into the YFS_INVENTORY_SUPPLY_TEMP table. The syncLoadedInventory API compares the YFS_INVENTORY_SUPPLY and YFS_INVENTORY_SUPPLY_TEMP tables, and whenever a difference is noticed, updates the YFS_INVENTORY_SUPPLY table, and raises the ON_INV_MISMATCH event.

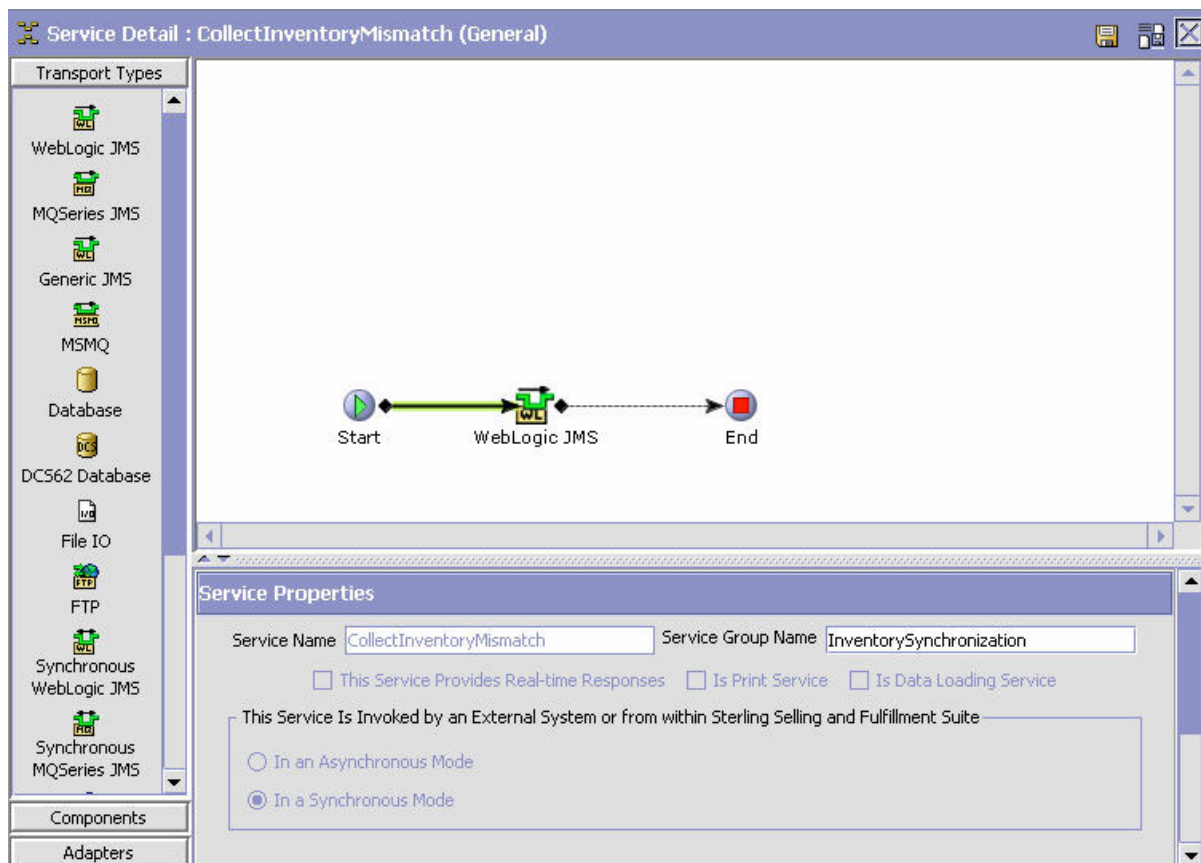
Configuring the CollectInventoryMismatch Service

About this task

The ON_INV_MISMATCH event invokes the CollectInventoryMismatch service, which places the published XML into a JMS queue.

Procedure

1. From the Applications menu of the Applications Manager, click Application Platform.
2. From the tree in the application rules side panel, double-click Process Modeling.
3. Click the General tab. In the Process Types swimlane, right-click the General process type, and click Model Process. The Repository Details window and work area display for the General process type.
4. Click the Service Definitions tab.
5. Expand the InventorySynchronization branch.
6. Right-click CollectInventoryMismatch, and click details. The Service Detail window displays in the work area.



7. In the CollectInventoryMismatch Service Detail window, click the green connector in between Start and WebLogic JMS. The JMS Sender properties display under the graphic area.
 - Click the Runtime tab.
 - Ensure that the Queue Name entered is the name of an existing JMS Queue.

- For descriptions of the rest of the fields, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

Executing the Synchronization Process

To start the process of synchronizing with node inventory, the pre configured LoadInvMismatchFileServer and LoadInvMismatchJMSServer servers in the LoadInventoryMismatch service need to be started using the provided agentserver.cmd or agentserver.sh in the <INSTALL_DIR>/bin directory:

- <INSTALL_DIR>/bin/agentserver.sh LoadInvMismatchFileServer
- <INSTALL_DIR>/bin/agentserver.sh LoadInvMismatchJMSServer

Once the two servers is running, files containing inventory information from nodes can be uploaded into the incoming directory specified for the File I/O component in the LoadInventoryMismatch service.

Purging the Temporary Table

Once the inventory picture of Sterling Selling and Fulfillment Foundation has been successfully updated, the contents of the YFS_INVENTORY_SUPPLY_TEMP need to be cleaned up. This can be achieved by running the Purge Inventory Supply Temp time-triggered purge transaction.

For more information about the Purge Inventory Supply Temp time-triggered purge transaction, see “Inventory Supply Temp Purge” on page 185.

Chapter 12. 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 100.

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 99.
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 99.
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 99.

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 44. 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 45. 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 46. 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 |

*These files are located in the following directory:
<INSTALL_DIR>/xapidocs/api_javadocs/XSD/HTML

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 47. 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 48. 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"> If left blank or the number specified is less than 10000, it defaults to 5000. If the number specified is greater than 10000, then that value is used. |
| 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 48. 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 49. 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 50. Change Load Status Parameters

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

Table 50. 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 51. 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 (<=) 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 52. 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 53. 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 54. 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 176).

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 55. 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 56. 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 57. 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 58. 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 190).

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 59. 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 60. 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 61. 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 62. 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 63. 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 64. 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 65. 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 66. 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 67. 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 68. Close Order Criteria Parameters

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

Table 68. 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 69. 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 70. 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 71. 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 71. 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 72. 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. Note: Both Node and AgentCriteriaGroup cannot be blank. You must specify a value for the Node parameter, the AgentCriteriaGroup 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 73. 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 74. 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 227).

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 75. 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 76. 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 76. 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 77. 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 (\leq) the current date value in the YFS_Task_Q table.

Events Raised

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

Table 78. 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 79. 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 79. Collect Shipment Statistics Attributes (continued)

| Attribute | Value |
|-------------------|-------|
| User Exits Called | None |

Criteria Parameters

The following are the criteria parameters for this transaction:

Table 80. 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 81. 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 82. 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 83. 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 84. 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 84. 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 85. 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 86. 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"> • It calls <code>beforeConsolidateToShipment</code> in <code>com.yantra.ydm.japi.ue</code>. • <code>YDMBeforeConsolidateToShipment</code> for each release before it begins processing. • After it finds the shipments, it calls <code>determineShipmentToConsolidateWith</code> in <code>com.yantra.ydm.japi.ue</code>. <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 87. 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 87. 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 88. 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 89. 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 90. 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 91. 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 91. Create Catalog Index Criteria Parameters (continued)

| Parameter | Description |
|---|---|
| Incremental Build | <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> |
| Category Domain | <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> |
| Auto Activate | <p>Y or N. Optional.</p> <p>Y to activate the index after building the index file.</p> <p>The default is N.</p> |
| Auto Insert Search Index Trigger | <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> |
| ColonyID | <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 92. 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 93. 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 94. 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 95. 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 96. 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 97. 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 | createDerivedOrder() |

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 98. Create Derived 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 99. Create Derived Order Statistics

| Statistic Name | Description |
|--------------------|-----------------------------------|
| NumOrdersProcessed | Number of orders processed. |
| NumOrdersCreated | 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 128.

Attributes

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

Table 100. 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 101. 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 102. 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 127.

Attributes

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

Table 103. 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 104. 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 105. 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 106. 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 107. 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 108. 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 109. 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 110. 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 111. 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 112. 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 113. 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 114. 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 (<=) 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 115. 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 116. 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 116. 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 117. 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 139.

Attributes

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

Table 118. 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 119. 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 120. 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 121. 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 122. 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 123. 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 124. 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 125. 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"> AUTHORIZATION - Validates the credit card account CHARGE - Applies the charge to the credit card |
| 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 126. Payment Execution Statistics

| Statistic Name | Description |
|-----------------------|--|
| NumAuthTransProcessed | Number of authorization transaction processed. |

Table 126. 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 127. 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 128. Post Inventory Match Attributes

| Attribute | Value |
|---------------------|----------------------|
| Base Transaction ID | POST_INVENTORY_MATCH |
| Base Document Type | General |

Table 128. 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 129. 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 130. 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 131. 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 132. 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 133. 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 134. 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 135. 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 136. 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 136. 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 137. 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 |
| * Note: 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 138. Publish Negotiation Results Attributes

| Attribute | Value |
|---------------------|-------------------------|
| Base Transaction ID | PUBLISH_ORD_NEGOTIATION |

Table 138. 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 139. 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 140. 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 141. 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 141. 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 |
| * Note: 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 142. 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 143. 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"> Y - Releases line quantities regardless of release date criteria N - Default value. Releases line quantities only after release date criteria have been met. |

Table 143. Release Criteria Parameters (continued)

| Parameter | Description |
|---------------------------------|--|
| CheckInventory | Optional. Determine whether inventory should be checked. Valid values are: <ul style="list-style-type: none"> • Y - Default value. Inventory needs to be checked. • N - Inventory does not need to be checked. |
| 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 144. 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 145. 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 146. 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 146. 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 147. 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 148. 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 149. 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 150. 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"> • 01 - Optimize on date (Default) • 02 - Optimize on ship node priority • 03 - Optimize on number of shipments |
| OrderFilter | Optional. Determines the types of orders to filter. Possible values are: <ul style="list-style-type: none"> • A - All orders (Default) • B - Backorders only • N - New orders only |
| ScheduleAndRelease | Optional. Notify the schedule process to release all releasable line quantities. Valid values are: <ul style="list-style-type: none"> • Y - Releases successfully scheduled line quantities. • N - Default value. Only schedules line quantities. 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"> • Y - Releases line quantities regardless of release date criteria. • N - Releases lines quantities only after release date criteria have been met. Default. |

Table 150. 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 151. 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"> • If the OrderFilter is “B” (Backorders Only) and there are no backordered or unscheduled lines. • If the OrderFilter is “N” (New orders Only) and there are some backordered or unscheduled lines. • 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. |
| 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"> • The result of the YFSCheckOrderBeforeProcessingUE user exit returns as false. • The Order has the HoldFlag attribute set to ‘Y’. • The Order has the SaleVoided attribute set to ‘Y’. • The Order does not have PaymentStatus as AUTHORIZED, INVOICED, PAID, nor NOT_APPLICABLE. |
| 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 151. Schedule Statistics (continued)

| Statistic Name | Description |
|-----------------------------------|---|
| NumOrdersScheduled | Number of orders that have at least one line that was scheduled. This includes scheduled lines in any status except BACKORDER. |
| NumOrdersProcOrdersCreated | Number of procurement orders created. |
| NumWorkOrdersCreated | Number of work orders created. |
| NumOrdersBackordered | Number of orders backordered. |
| NumOrderLinesScheduled | Number of order lines scheduled. |
| NumOrderLinesBackordered | Number of order lines backordered. |
| NumReleasesCreated | 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 152. 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 153. 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 154. 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 155. 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 156. 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 157. 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 158. 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 159. 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 160. 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 161. 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 162. 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 163. Send Order Criteria Parameters

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

Table 163. 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 164. 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 165. Send Release Criteria Parameters

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

Table 165. 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 166. 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 167. 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 168. Start Order Negotiation Attributes

| Attribute | Value |
|---------------------|-----------------------|
| Base Transaction ID | START_ORD_NEGOTIATION |
| Base Document Type | Order |
| Base Process Type | Order Fulfillment |

Table 168. 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 169. 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 170. 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 (\leq) 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 171. 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 172. 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 173. 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 174. 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 174. Update Best Match Region Criteria Parameters (continued)

| Parameter | Description |
|--------------------------------|---|
| TableType | <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> |
| ColonyID | <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> |
| UpdateOnly | <p>Mode in which to run. Valid values are:</p> <ul style="list-style-type: none"> • 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. • 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. |
| LastPersonInfoKey | <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> |
| LastRegionBest MatchKey | <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 175. 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 176. 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 177. 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 178. 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 178. Access Token Purge Criteria Parameters (continued)

| Parameter | Description |
|-----------|--|
| Live | Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 179. 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 180. 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 181. 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"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 182. 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 173.

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 201 for more information.

Criteria Parameters

The following are the criteria parameters for defining a draft order history transaction:

Table 183. 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">• Y - Default value. Removes qualifying records from the history tables that are listed in Tables Purged.• N - Test mode. Determines the rows that are removed without actually removing them. |
| 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 170.

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 201 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 184. Draft 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 | 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. |
| Live | Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged, to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| PurgeCode | Required. Set to DRAFTORDERPRG. 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_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 185. Delivery Plan Purge Attributes

| Attribute | Value |
|----------------------|-----------------|
| Base Transaction ID | DELIVERYPLANPRG |
| Base Document Type | Load |
| Base Process Type | Load Execution |
| Abstract Transaction | No |

Table 185. 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 186. 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"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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"> • Y - Default value. Records are deleted in batches. • N - Records are deleted one by one. |
| 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 187. 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 188. 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 189. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |

Table 189. Export Table 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"> • Y - Records are deleted in batches. • N - Default value. Records are deleted one by one. |
| 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_EXPORT table may exist in multiple schemas. Runs the agent for the colony. |

Statistics Tracked

The following statistics are tracked for this transaction:

Table 190. 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 191. 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 192. 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"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| BatchDelete | Required. The method by which all records are deleted from the table. Valid values are: <ul style="list-style-type: none"> • Y - Records are deleted in batches. • N - Default value. Records are deleted one by one. |
| 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 193. 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 194. 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 195. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Table Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 196. 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 197. 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 198. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 199. 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 200. 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 201. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 202. 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 203. 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 204. 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">• Y - Default value. Production mode. Deletes records from the regular tables.• N - Test mode. |
| 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 205. 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 190.

Attributes

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

Table 206. 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 207. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 208. 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 209. 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 210. 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 210. 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"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 211. 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 212. 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 213. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |

Table 213. 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 214. 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 215. 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 216. 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"> Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 217. 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 197.

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 218. 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 219. 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">• Y - Default value. Removes qualifying records from the history tables listed under Tables Purged.• N- Test mode. Determines the rows that are removed without actually removing them. |
| 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 220. 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 221. 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 222. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 222. 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 223. 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 201.

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 224. 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 225. 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"> Y - Default value. Removes qualifying records from the history tables listed under Tables Purged. N- Test mode. Determines the rows that are removed without actually removing them. |
| 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 226. 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 198. 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 227. 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 228. 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 228. Order Purge Criteria Parameters (continued)

| Parameter | Description |
|----------------------------|---|
| Live | Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| PurgeCode | 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"> • DRAFTORDERHISTPRG to purge draft order information from the order history tables. • DRAFTORDERNOLINEHISTPRG to purge draft orders without order lines from the order history tables. • DRAFTORDERNOLINEPRG to purge draft orders that have no order lines. • DRAFTORDERPRG to purge draft order information and archive it in the order history tables. PurgeCode cannot be set to the value ORDER_RELEASE_STATUS_PURGE. |
| AdditionalPurgeCode | 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 206. |
| 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 229. Order Purge Statistics

| Statistic Name | Description |
|---------------------------|----------------------------|
| NumOrdersProcessed | Number of order processed. |
| NumOrdersPurged | 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 201 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 230. Order Release Status 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. |
| Live | Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| PurgeCode | 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. |
| AdditionalPurgeCode | Required. To purge order release status records, set this parameter to ORDER_RELEASE_STATUS_PURGE. |
| 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

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 231. 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 232. 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 232. Order Status Audit Purge Criteria Parameters (continued)

| Parameter | Description |
|-----------------------|--|
| EnterpriseCode | Optional. Enterprise for which the Order Status 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"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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_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 233. Order Status Audit Purge Statistics

| Statistic Name | Description |
|------------------------------|---------------------------------|
| NumStatusAuditsPurged | 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 234. 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 235. 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"> • Y - Default value. Production mode. Deletes records from the regular tables. • N - Test mode. |
| 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 236. 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 237. 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 238. 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">• If left blank or the number specified is less than 10000, it defaults to 10000.• If the number specified is greater than 10000, then that value is used. |
| Live | Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |

Table 238. PersonInfo 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. |
| 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. |
| TableType | <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> |
| ColonyID | 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 239. PersonInfo Purge Statistics

| Statistic Name | Description |
|----------------------------|---------------------------------------|
| NumPersonInfoPurged | 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 240. 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 241. 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">• If left blank or the number specified is less than 10000, it defaults to 10000.• If the number specified is greater than 10000, then that value is used. |
| Live | Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 241. PersonInfo History Purge Criteria Parameters (continued)

| Parameter | Description |
|------------------|--|
| TableType | <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> |
| ColonyID | <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 242. PersonInfo History Purge Statistics

| Statistic Name | Description |
|---|---|
| NumPersonInfoHistoryRecords Purged | 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 243. 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 244. 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"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 245. 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 246. 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 247. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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. 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 249. 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 250. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 251. 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 221.

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 252. 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 253. 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">• Y - Default value. Removes qualifying records from the history tables listed under Tables Purged.• N- Test mode. Determines the rows that are removed without actually removing them. |
| 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 254. 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 219. 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 255. 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 256. 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 256. Receipt Purge Criteria Parameters (continued)

| Parameter | Description |
|-----------|--|
| Live | Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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. 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 258. 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 259. 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"> Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 260. 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 261. 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 262. 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 262. Reservation Purge Criteria Parameters (continued)

| Parameter | Description |
|------------------|--|
| Live | Optional. Mode in which to run. Valid values are: <ul style="list-style-type: none"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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_RESERVATION table may exist in multiple schemas. Runs the agent for the colony. |

Statistics Tracked

The following statistics are tracked for this transaction:

Table 263. 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 227.

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 114.

Attributes

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

Table 264. 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 265. 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">• Y - Default value. Removes qualifying records from the history tables listed under Tables Purged.• N- Test mode. Determines the rows that are removed without actually removing them. |
| 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 266. 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 225. 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 114.

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 267. 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 268. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 269. 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 270. 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 271. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 272. Shipment Statistics Purge Statistics

| Parameter | Description |
|------------------------------------|---------------------------------------|
| NumShipmentStatisticsPurged | 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 273. User Activity Purge Attributes

| Attribute | Value |
|----------------------------|-----------------|
| Base Transaction ID | USERACTIVITYPRG |
| 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 274. 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">• Y - Default value. Moves qualifying records from the regular tables listed under to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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">• Y - Default value. Records are deleted in batches.• N - Records are deleted one by one. |
| 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 275. 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 276. 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 277. 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">• Y - Default value. Moves qualifying records from the regular tables listed under to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 277. 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"> Y - Default value. Records are deleted in batches. N - Records are deleted one by one. |
| 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 278. 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 236.

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 279. 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 280. 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">• Y - Default value. Removes qualifying records from the history tables listed under Tables Purged.• N- Test mode. Determines the rows that are removed without actually removing them. |
| 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 281. 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 282. Work Order Purge Attributes

| Attribute | Value |
|---------------------|------------------|
| Base Transaction ID | WORK_ORDER_PURGE |
| Base Document Type | Work Order |
| Base Process Type | VAS |

Table 282. 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 283. 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"> • Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables. • N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 284. 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 285. 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 286. 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"> Y - Default value. Production mode. Deletes records from the regular tables. N - Test mode. |
| 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 286. 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 287. 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 288. 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 289. YFSInventoryOwnership Purge Criteria Parameters

| Parameter | Description |
|------------------------------------|---|
| Action | Required. Triggers the transaction. If left blank, this value 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), this value defaults to 5000. |
| EnterpriseCode | Optional. The inventory organization for which the YFSInventoryOwnership Audit Purge needs to 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">• Y - Default value. Production mode. Deletes records from the regular tables.• N - Test mode. |
| PurgeCode | Required. Cannot be modified. Used for internal calculations, such as determining retention days. Corresponds to the PurgeCode used in the Business Rules Purge Criteria. |
| Lead Days | Number of days before the present date, the agent will purge the records. |
| ColonyID | 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 290. 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 291. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 292. 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 293. 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 294. 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">• Y - Default value. Moves qualifying records from the regular tables listed under Tables Purged to the corresponding history tables.• N - Test mode. Determines the rows that are moved to history tables without actually moving them. |
| 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 295. 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 296. 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 297. 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 298. 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 299. 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 300. 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 301. 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 302. 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 303. 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 304. 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 305. 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 306. 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 307. 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 308. 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 309. 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 310. 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 311. 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 312. 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"> • 1 - current inventory • 0 - inventory within and outside of the ATP time frame. This is the default 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 owner inventory organization. Organization to process in this run. If not passed, all inventory organizations are processed. |

Table 312. 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 313. 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 314. 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 315. 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 316. 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 317. 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><AgentCriteriaAttribute> <OverriddenValue></pre> <p>For more information about passing these attributes, see the <i>Sterling Selling and Fulfillment Foundation: Installation Guide</i></p> |

Table 317. 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 318. 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 319. 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 320. 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 111.

The same relog interval is used for all document types.

Attributes

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

Table 321. 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 322. 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 323. 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 324. 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: <INSTALL_DIR>/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 111.

The same relog interval is used for all document types.

Attributes

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

Table 325. 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 326. 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 327. 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 (<=) 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 111.

The same relog interval is used for all document types.

Attributes

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

Table 328. 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 329. 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 329. 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 330. 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 331. 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 332. 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 333. 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 334. 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 335. Real-time Availability 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 | 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. |
| MonitorOption | 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. |
| ItemStatuses | 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. |
| FromAlertTimestamp | 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. |
| AllowedOverriddenCriteria | 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 335. Real-time Availability Monitor Criteria Parameters (continued)

| Parameter | Description |
|---|--|
| FromLastNumberOfHours | 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. |
| 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. |
| RaiseEventsOnAllAvailability Changes | 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. |
| ColonyID | 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 336. Events Raised by the Realtime Availability Monitor Transaction

| Transaction/Event | Key Data | Data Published* | Template Support? |
|-------------------------------------|----------|--|-------------------|
| REALTIME_AVAILABILITY_CHANGE | 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 337. 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 338. Shipment Monitor Criteria Parameters

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

Table 338. 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 339. 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 340. 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 341. 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 342. 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.

Index

A

Accumulation Time (Days) field 25
Action based ATP monitoring rules
 creating 29
actions 77
 value-added services
 viewing 82
activity codes
 transportation
 creating 67
 deleting 68
 modifying 68
Add Activity rule 74
Add Instruction rule 74
Add Note rule 74
Advance Notification Time (Days)
 field 25
Allocate Work Order transaction 80
Amount field 48
application rules side panel 6
Applications Manager
 actions 16
 document types 17
 entering dates/times 19
 lists 19
 lookup functionality 17
 special characters 20
 troubleshooting 20
 users 18
 layout 5
 starting 5
 work area 12
Assume FEFO if no Ship By Date
 Provided field 38
ATP Monitor Rule field 27, 29
ATP Monitor Rule Name field 27, 29
ATP monitoring rules
 deleting 28, 31
 modifying 28, 30
ATP Rule field 23
ATP Rule Name field 23
ATP rules 22
 creating 23
 default ATP rule 26
 defining 22
 definition 22
 deleting 26
 modifying 25
ATP Rules
 default lead time 26
Availability 55
available-to-promise rules. See ATP
 rules 22

B

Backward Consumption (Days) field 24
building
 catalog index 121
business analysts 1

business models 1
buyers 2

C

Calendar field 61
Cancel Work Order transaction 74, 80
Cancellation Reason field 69
capacity
 resource reservation 55
 resources spanning slots 55
Capacity Information Available field 60,
 61
Capacity Organization field 60, 61
capacity rules
 default capacity reservation expiration
 time
 allow capacity to span service
 slots 55
 defining 55
 defining 55
capacity to span service slots 55
Capacity UOM field 60, 61
catalog
 index building 121
Change Appointment rule 74
Change Appointment User rule 74
Change Finish Date rule 74
Change Instruction rule 74
Change Other Attributes 74
Change Other Relationships 74
Change Product Line rule 74
Change Service Line rule 74
Change Ship To rule 74
Change Start Date rule 74
Change Station Assignments rule 74
Choose Calendar field 61
CollectInventoryMismatch service
 configuring 95
conditions 77
 creating 82
 value-added services
 viewing 81
configuration screens
 accessing 7
Confirm Work Order transaction 80
corporate count request cancellation
 reason codes
 creating 87
 creating new from existing 88
 defining 87
 deleting 89
 modifying 88
Cost Factor Description field 48
cost factor groups
 associating cost factors 47
 defining 47
Cost Factor Name field 48
Cost Factor UOM field 48
Cost Factor Value field 48
Costing Required field 42

count programs
 creating 85
 deleting 87
 modifying 86
count request cancellation reason
 codes 88
counts
 defining 85
Create Demand Details field 38
Create Work Order transaction 80

D

Decrease Priority rule 74
default capacity reservation expiration
 time
 defining 55
Delivery Service field 60
Demand Is Communicated To External
 Systems flag 44
Demand Type field 44
demand types
 creating 43
 deleting 44
 modifying 44
Description field 77
Disabled field 27, 29
distribution groups
 adding external organizations 49
 adding nodes 49
 creating 49
 deleting 52
 deleting advanced distribution
 details 51
 deleting external organizations 50
 deleting nodes 50
 modifying external organizations 50
 modifying nodes 50

E

Enterprise administrators 1
enterprises 2

F

FEFO (First Expiration First Out) 22
first expiration first out. See FEFO 22
Forward Consumption (Days) field 24

H

HasComponents condition 82
Hold Type Modifications rule 74
Hub administrators 1

I

- inbound processing 22
- Increase Priority rule 74
- index
 - catalog search 121
- inheritance
 - determining 7
- inventory availability 26
- inventory availability safety factor 45
 - modifying 45
- inventory considerations
 - configuring 41
 - defining 41
- inventory demand types 43
- inventory handling 21
- inventory pictures
 - loading 91
- inventory reasons 22, 33
 - creating 33
 - deleting 34
 - modifying 34
- inventory rules 3
 - configuring 21
- inventory supply types 41
- inventory tables
 - synchronizing 94
- inventory types
 - configuring 41
- item availability 21
- Item Group field 61

L

- Lead Time Override field 27, 29
- LoadInventoryMismatch service 92, 93
 - configuring 93
- Log File Name field 84, 90

M

- marketplaces 1, 2
- Maximum no. of days order can be shipped/delivered beyond its requested date field 24
- modification rules 72
 - value-added services
 - setting up 72
- monitor rules 22, 26
 - event based 26
 - creating 27
- multidivisional corporations 1

N

- Node field 60
- node inventory
 - synchronizing 91
- node synchronization
 - synchronization process
 - executing 96
 - temporary table
 - purging 96
- nodes
 - inventory pictures
 - loading 91

- Non-Committed field 44

O

- Onhand Supply field 42
- Optimize Supply on earliest ETA 38
- organization levels 7
 - rules 9
- organization rules 9
 - loading another organization's
 - rules 12
 - overriding 10
- Outbound processing 22

P

- Past Due Demand Days field 24
- Past Due Supply Days field 23
- Percentage field 48
- pipelines 72, 77
 - creating 79
 - value-added services
 - viewing 78
- Pricing organization 8, 9
- Priority field 50
- process type details
 - value-added services
 - viewing 77
- Process Type field 77
- Process Type Name field 77
- Processing Time (Days) field 25
- product classes 22, 32
 - creating 32
 - deleting 33
 - modifying 33
- Promised field 44
- Provided Service field 60
- Purge Code field 83, 90
- purge criteria
 - corporate count request s 89
 - value-added services
 - setting up 83
- Purge Work Order transaction 80

Q

- Quantity field 48

R

- reason codes 68, 87
- region schemas
 - defining for resource pools 56, 87
- regions 56, 87
 - adding to a resource 63
 - removing from a resource pool 64
- Release Work Order transaction 80
- Released field 44
- Remove Activity rule 74
- Resequence Activity rule 74
- resource capacity 55
 - across slots 55
 - reservation 55
- Resource Pool Description field 60, 61
- Resource Pool ID field 60

- resource pools 59
 - adding regions to 63
 - creating 59
 - deleting 66
 - modifying 59
 - removing a region 64
- service resources
 - creating 64
 - deleting 66
 - modifying 64
- service skills
 - adding 63
 - removing 63
- Retain Reference field 42
- Retention Days field 84, 90
- Rollback Segment field 83, 90

S

- sellers 2
- service definitions
 - value-added services
 - viewing 83
- service nodes 59
- service resources
 - creating 64
 - definition 64
 - deleting 66
 - modifying 64
 - team members 65
- service skills
 - adding 63
 - removing 63
- Service Slot Group field 60, 61
- service slots 56
- services 77
- ship node determination 3
- Slot Group ID field 57
- slot groups 56
 - creating 57
 - deleting 59
 - modifying 58
- Source Organization field 50
- Source Ship Node field 50
- statuses 77
 - creating 81
 - value-added services
 - viewing 80
- Supervisor ID field 61
- Supply Type field 42
- supply types 41
 - creating 41
 - deleting 43
 - modifying 42
- synchronization process
 - executing 96
- Synchronize Expiration Date Across Inventory Reservations Based On Reservation ID field 38
- SyncLoadedInventory service 94
- system integration 72

T

- team members 65
- third-party logistics companies 1

- third-party logistics models 2
- Third-party logistics models 2
- transactions 77
 - creating 80
 - value-added services
 - viewing 79
- transportation
 - activity codes 67, 68

U

- Use Item Based Allocation field 35
- Use Node's Calendar flag 65
- Use Nodes Calendar field 61

V

- Validate Item During Inventory
 - Adjustment flag 38
- value-added services
 - configuring 67
 - defining 67
 - modification rules 72
- Volume field 48

W

- Weight field 48
- Work Order Allocated status 81
- work order allocation considerations
 - creating 70
 - deleting 71
 - modifying 71
- Work Order Canceled status 81
- work order cancellation reason codes
 - creating 68
 - creating new from existing 69
 - defining 68
 - deleting 69
 - modifying 69
- Work Order Confirmed status 81
- Work Order Created status 81
- Work Order With Components Created
 - status 81
- Write To Log File field 84, 90

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)[®] 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[®], Connect:Direct[®], Connect:Enterprise[®], Gentran[®], Gentran[®]:Basic[®], Gentran:Control[®], Gentran:Director[®], Gentran:Plus[®], Gentran:Realtime[®], Gentran:Server[®], Gentran:Viewpoint[®], Sterling Commerce[™], Sterling Information Broker[®], and Sterling Integrator[®] are trademarks or registered trademarks of Sterling Commerce[™], Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.



Printed in USA