

Sterling Store Inventory Management



# Implementation Guide

*Release 9.2*



Sterling Store Inventory Management



# Implementation Guide

*Release 9.2*

**Note**

Before using this information and the product it supports, read the information in "Notices" on page 61.

**Copyright**

This edition applies to the 9.2 Version of IBM Sterling Store Inventory Management and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2006, 2012.**

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

---

# Contents

<b>Chapter 1. Introduction to the IBM Sterling Store Inventory Management Implementation</b>	<b>1</b>
--	----------

<b>Chapter 2. Installing IBM Sterling Store Inventory Management Reference Implementation</b>	<b>3</b>
---	----------

Install IBM Sterling Store Inventory Management using Master Data	3
Install IBM Sterling Store Inventory Management using Configuration Data	4
Install IBM Sterling Store Inventory Management using Transaction Data	4

<b>Chapter 3. Managing Users</b>	<b>5</b>
----------------------------------	----------

Defining Access Control	5
Access Control Solution	5
Access Control End-User Impact	5
Access Control Implementation	6
Access Control Reference Implementation	6

<b>Chapter 4. Inventory Adjustments</b>	<b>7</b>
---	----------

Inventory Adjustments	7
Inventory Adjustment Reasons	7
Inventory Adjustment Solution	7
Inventory Adjustment End-User Impact	8
Configuring Inventory Adjustment Rules	8
Inventory Adjustment Reference Implementation	8

<b>Chapter 5. Inventory Searches</b>	<b>11</b>
--------------------------------------	-----------

Inventory Searches	11
Inventory Searches Solution	11
Inventory Searches End-User Impact	11
Inventory Searches Implementation	11
Inventory Searches Reference Implementation	12

<b>Chapter 6. Inventory Moves</b>	<b>13</b>
-----------------------------------	-----------

Inventory Moves	13
Inventory Moves Solution	13
Inventory Moves End-User Impact	13
Inventory Moves Implementation	14
Inventory Moves Reference Implementation	14

<b>Chapter 7. Inventory Audits</b>	<b>15</b>
------------------------------------	-----------

Inventory Audit	15
Inventory Audit Solution	15
Inventory Audit End-User Impact	15
Inventory Audit Implementation	15
Inventory Audit Reference Implementation	15

<b>Chapter 8. Counts</b>	<b>17</b>
--------------------------	-----------

Counts	17
Cycle Count	17

Cycle Count Process	17
Cycle Count Solution	18
Cycle Count End-User Impact	20
Cycle Count Implementation	20
Cycle Count Reference Implementation	21
Physical Counts	23
Physical Count Process	23
Physical Count Solution	23
Physical Count End-User Impact	26
Physical Count Implementation	26
Physical Count Reference Implementation	26

<b>Chapter 9. Receiving</b>	<b>29</b>
-----------------------------	-----------

Receiving	29
Receiving Processes	29
Receiving Process Solution	30
Receiving Process End-User Impact	32
Receiving Process Implementation	32
Receiving Process Reference Implementation	33

<b>Chapter 10. Managing Locations</b>	<b>35</b>
---------------------------------------	-----------

Managing Store Locations	35
Store Location Solution	35
Store Location End-User Impact	35
Store Location Implementation	35
Store Location Reference Implementation	35

<b>Chapter 11. Shipping</b>	<b>37</b>
-----------------------------	-----------

Shipping	37
Printing Pick Tickets	37
Pick Ticket Solution	37
Pick Ticket End-User Impact	37
Pick Ticket Implementation	38
Pick Ticket Reference Implementation	38
Searching for Shipments	39
Shipments Search Solution	39
Shipments Search End-User Impact	40
Shipments Search Implementation	41
Shipments Search Reference Implementation	41
Recording Back Room Picks	41
Back Room Picks End-User Impact	41
Back Room Pick Implementation	41
Back Room Picks Reference Implementation	42
Recording Customer Picks	42
Customer Picks Solution	42
Customer Picks End-User Impact	43
Customer Picks Implementation	43
Customer Picks Reference Implementation	44
Creating Loads	44
Load Solution	44
Load End-User Impact	45
Load Implementation	45
Load Reference Implementation	45
Adding to a Manifest	46
Adding Manifest Solution	46

Adding Manifest End-User Impact . . . . .	46
Adding Manifest Implementation . . . . .	46
Adding Manifest Reference Implementation . . . . .	47
Closing a Manifest . . . . .	47
Closing Manifest Solution . . . . .	47
Closing Manifest End-User Impact. . . . .	47
Closing Manifest Implementation . . . . .	47
Closing Manifest Reference Implementation . . . . .	47

**Chapter 12. Alert Console . . . . . 49**

Alert Console. . . . .	49
Alert Console Solutions . . . . .	49
Alert Console End-User Impact. . . . .	49
Alert Console Implementation . . . . .	49
Configuring Alerts . . . . .	49
Configuring Queues . . . . .	50

Triggering the Agent . . . . .	50
Alert Console Reference Implementation . . . . .	51

**Chapter 13. Reference Implementation 53**

Reference Implementation . . . . .	53
Reference Implementation Setup . . . . .	53
Store Layout Configuration . . . . .	53
Inventory Rules Configuration . . . . .	54
Inbound Rules Configuration . . . . .	56
Outbound Rules Configuration . . . . .	59

**Notices . . . . . 61**

**Index . . . . . 65**

---

## Chapter 1. Introduction to the IBM Sterling Store Inventory Management Implementation

The IBM® Sterling Store Inventory Management presents solutions to the challenges found in typical store management business scenarios.

An in-store pick-up system allows customers to place orders on the web site and pick up orders at their nearest store. The added value for customers stems from savings on shipping costs, immediate gratification, and the opportunity to view and buy additional items once they are inside the store. Retailers expect to increase the frequency of customer visits to the store by creating this stronger integration between channels, thereby, boosting revenues and profits. This also serves as a differentiating factor by providing additional service and increasing their customers' convenience.

One of the keys to success with in-store pickup is effective coordination between a retailer's online and offline operations. Given the competitive pressures and customer benefits associated with this service, in-store pick-up will soon no longer be an order winner for retailers, but rather an order qualifier, a requirement for any successful multi-channel retailer.

The inventory management for in-store pickup operations includes the use of mobile devices for managing certain operations in the store.





---

## Chapter 2. Installing IBM Sterling Store Inventory Management Reference Implementation

The Sterling Store Inventory Management provides a pre-configured reference implementation. The reference implementation provides standard configurations and processes followed in a typical store. Additionally, it may also include certain transactional data necessary to demonstrate its ability.

Installing the Sterling Store Inventory Management reference implementation is optional. Also, there are other options when installing the reference implementation:

- “Install IBM Sterling Store Inventory Management using Master Data”
- “Install IBM Sterling Store Inventory Management using Configuration Data” on page 4
- “Install IBM Sterling Store Inventory Management using Transaction Data” on page 4

For more information about installing the reference implementation, refer to the *Sterling Selling and Fulfillment Suite: Applications Reference Implementation Guide*.

---

### Install IBM Sterling Store Inventory Management using Master Data

#### About this task

The default installation of the IBM Sterling Store Inventory Management reference implementation is done with the configuration data and master data.

To use your own master data, do not choose the default installation. To load the master data, you can use the Rapid Deployment Tools (RDT). For more information about RDT, see the *Sterling Selling and Fulfillment Foundation: Integration Guide*.

You can use RDT for creating locations and items.

#### Procedure

1. Before you run RDT, create External/RDTConfigData/LoadStore folder under <INSTALL\_DIR> directory.
2. Under <INSTALL\_DIR>/External/RDTConfigData/LoadStore create Input, Error, Working, and Complete directories.
3. Create a csv file and include all stores that you want to configure in this file. The csv file should have all columns specified in the RDTStoreConfigData.xls.
4. In the RDTStoreConfigData.csv file, delete the first row (header) and ensure that the file has only 38 columns.
5. Run the integration server using the <INSTALL\_DIR>/bin/startIntegrationServer script with the server name SOP\_Store\_Loader as the parameter.

The server retrieves all files located in the Input directory, copies them into the Working directory, and loads the store. Any errors that occur are copied to the Error directory. The loaded records are copied to the Complete directory.

Once the stores are loaded you can configure each store as a seller using the Sterling Store Inventory Management Configurator. If a store is modelled as a seller you can define the inventory and catalog organization code for the store. However, for the store inventory operations, the store uses inventory and catalog organization code of the enterprise with which you logged into the application.

After configuring the stores as sellers, you can onboard the store one by one using the Sterling Store Inventory Management Configurator. For more information about onboarding a store, see "Onboarding Store".

---

## **Install IBM Sterling Store Inventory Management using Configuration Data**

### **About this task**

If you choose to install the reference implementation with your own configuration data, it is assumed that you understand the various concepts and configurations involved. For more information about the various concepts and configurations, see the *Sterling Store Inventory Management: Concepts Guide* and the *Sterling Store Inventory Management: Implementation Guide*.

---

## **Install IBM Sterling Store Inventory Management using Transaction Data**

### **About this task**

You can use the transaction data provided with the IBM Sterling Store Inventory Management to understand and modify its processes. However, if you use your own transaction data, ensure that the data is complete and in sync with the configuration and master data provided by the *IBM Sterling Store Inventory Management*.

---

## Chapter 3. Managing Users

---

### Defining Access Control

User Security enables you to ensure that users access only the required information for carrying out their tasks. This requires you to set up user groups and associate users to the user groups. The IBM Sterling Store Inventory Management Configurator allows you to create users and user groups and assign permissions.

---

### Access Control Solution

A user has the privilege to change the password or change the user group of another user.

#### Changing a Password

A user can change password in the Change Password screen. The `modifyUserHierarchy` API is invoked to change the password. The following table describes the various validations that the system performs before changing the password.

*Table 1. Change Password Validations*

Validation	Result
If the old password and new password match	An error message displays.
If the new password and confirmed password do not match	An error message displays.
If the old password is incorrect	An error message displays.
If the old password field is blank	An error message displays.

#### Managing Users

A store manager can assign a user to a user group in the Manage Users screen. The `getUserList` API provides a list of all users, and the `getUserGroupList` API provides a list of all user groups to which the user can be assigned. After selecting the user and the appropriate user group to which the user needs to be assigned, click the Confirm button. The `modifyUserHierarchy` API is invoked, which modifies the user group.

If the store manager wants to transfer a store associate from one store to another store, the `getOrganizationHierarchy` API validates the store that you have entered before transferring the store associate. However, the store manager cannot choose to move their own user identifier to another store. Only another user who has permission to transfer the store manager can perform this operation.

---

### Access Control End-User Impact

None.

---

## **Access Control Implementation**

None.

---

## **Access Control Reference Implementation**

None.

---

## Chapter 4. Inventory Adjustments

---

### Inventory Adjustments

Inventory adjustments are made when item quantities at a location do not match with the system quantity. Before performing inventory adjustments, the IBM Sterling Store Inventory Management validates the item and location details.

---

### Inventory Adjustment Reasons

Inventory adjustments are associated with a reason code and reason text. These codes are used as tracking mechanisms for the exceptions that require an adjustment. The inventory adjustment reason enables tracking and reporting of adjustments at a reason code level.

#### Default Inventory Adjustment Reason Codes

The adjustment reason codes configured as part of the IBM Sterling Store Inventory Management are:

- Re-Classify — This reason code is used to adjust inventory that is incorrectly classified. This reason code increases or decreases the inventory quantity.
- Scrap — This reason code is used to scrap inventory. This reason code always decreases the inventory quantity.
- Found-New — This reason code is used when a new item is found in the location. This reason code always increases the inventory quantity.
- Missing — This reason code is used when inventory is missing for an item. This reason code always decreases the inventory quantity.

---

### Inventory Adjustment Solution

When adjusting inventory, the system validates the item and item UOM by calling the `getItemDetails` and `getItemUOMList` APIs. After entering the location, the item quantity in that location is validated by calling the `getNodeInventory` API.

Confirmation of the inventory adjustment is done by calling the `adjustLocationInventory` API.

The configuration rules that need to be set up for inventory adjustments are:

- Inventory adjustment reason codes.
- Inventory transition rules for a store.

The following table lists all validations done during an inventory adjustment.

*Table 2. Inventory Adjustment Validations*

Validation	Result
If the user enters a wrong item ID	An error message displays.
If the user enters a wrong location	An error message displays.

Table 2. Inventory Adjustment Validations (continued)

Validation	Result
If the user does not enter the item quantity correctly	An error message displays.
If the quantity entered for negative adjustment is lesser than zero	An error message displays.
If the item entered is tag-controlled	An error message displays.

---

## Inventory Adjustment End-User Impact

The IBM Sterling Store Inventory Management supports only one product class and inventory UOM.

---

## Configuring Inventory Adjustment Rules

An inventory adjustment is associated with a reason. The Inventory Adjustment Reason allows tracking and reporting of all adjustments at a reason code level. Inventory adjustment reason codes provide granularity for reporting and tracking of a store performance.

For more information about configuring store inventory adjustment reasons, see the *Sterling Store Inventory Management: Configuration Guide*.

---

## Inventory Adjustment Reference Implementation

This section explains the reference implementation provided by the Sterling Store Inventory Management for Inventory Adjustments.

### Inventory Adjustment Reference Implementation Reasons

The Inventory Adjustment Reasons configured in the Sterling Store Inventory Management are described in the following table.

Table 3. Inventory Adjustment Reasons

Reason Code	Description	Operation
Re-Classify	Used by a store associate to adjust inventory that was incorrectly classified	Increases and decreases inventory
Found-New	Used by a store associate when inventory is found in the location, but not recorded in the system	Increases inventory
Missing	Indicates that the inventory is missing	Decreases inventory
Scrap	Indicates that the inventory has been damaged and is scrapped	Decreases inventory

## Inventory Transitions

The following table lists the default configuration for inventory status transitions provided in the Sterling Store Inventory Management.

*Table 4. Inventory Status Transitions*

<b>Zone</b>	<b>Transition</b>
For all zones	Blank to Normal

For more information about defining inventory transition rules, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.





---

## Chapter 5. Inventory Searches

---

### Inventory Searches

A store associate may want to search for an item based on certain search criteria. Along with the list of items that match the search criteria, the store associate may also want to view item details such as availability, price, promotions, and specifications.

---

### Inventory Searches Solution

The IBM Sterling Store Inventory Management provides search screens for searching items. Additionally, a mechanism to view small and large icons that represent each item in your catalog is also provided.

This section explains the complete Item Inquiry process.

#### Searching for Items

The `getOrganizationList` API is used to retrieve a list of organization codes. The `getItemAttributeGroupsForCategory`, `getClassificationPurposeList`, and `getCategoryList` APIs are called to populate the item-specific attributes. The `getClassificationPurposeList` and `getCategoryList` APIs are called to populate type of item. Depending on the type of item selected, the other search criteria display.

#### Retrieving Item Details

The `getItemListForOrdering` API retrieves a list of items based on the search criteria. This list also provides additional information such as pricing, availability, and so forth.

---

### Inventory Searches End-User Impact

You can configure item classifications and classification hierarchies that can be used as search criteria. A classification hierarchy contains all of the valid values a given classification can accept in a hierarchical format. You can define specification groups for a classification value. Based on the selected item type, the specification attributes display.

For more information about defining classifications, see the *Sterling Selling and Fulfillment Foundation: Catalog Management Configuration Guide*.

---

### Inventory Searches Implementation

In the IBM Sterling Store Inventory Management, whenever an item is selected, item images display along with the item details. The items are associated with large or small images. These images reside on different servers and can be retrieved when needed. To configure the server from which you fetch images, you must define entries such as name, protocol, base URL, and so forth in the `locations.ycfg` file stored in the `com.yantra.yfc.rcp_1.0.0` plug-in folder. You can create more than one URL to display different types of images.

For more information about configuring connection settings for fetching images from the server, see the *Sterling Selling and Fulfillment Foundation: Installation Guide*.

**Note:** You must create the IMAGE URL to display images from the server. You can configure the IMAGE URL to get images in gif and jpeg formats.

**Note:** The screens within the Sterling Store Inventory Management application have been designed for small item images of 32 X 32 pixels in size and large items of 180 X 180 pixels in size. It is recommended that your images conform to these sizes for ideal display within the screens.

---

## Inventory Searches Reference Implementation

When checking for the availability of an item in a store, by default, the system displays all items available for the next 5 days from the current date.

---

## Chapter 6. Inventory Moves

---

### Inventory Moves

Inventory move operations are performed in a store when receiving items or during space consolidation. The IBM Sterling Store Inventory Management application does not allow a store associate to move more than the existing quantity of items from one location to another location. Before moving inventory, the Sterling Store Inventory Management validates the item and location details.

---

### Inventory Moves Solution

Before moving inventory, the system validates item and item UOM by calling the `getItemDetails` and `getItemUOMList` APIs.

After you enter the location, the system validates the quantity of inventory in that location by calling the `getNodeInventory` API.

The inventory move is confirmed by calling the `moveLocationInventory` API.

The following table lists all validations performed during inventory moves.

*Table 5. Inventory Move Validations*

Validation	Result
In the From Location and To Location fields, if the user enters invalid locations	An error message displays.
In the Item ID field, if the user enters an invalid item identifier	An error message displays.
In the Move Qty field, if the user enters a quantity that is more than the current quantity that displays in the Current Qty field	An error message displays.
In the Item ID field, if the user does not enter the item that needs to be moved	An error message displays.

Inventory moves are applicable only to stores that track multiple locations or limited locations.

---

### Inventory Moves End-User Impact

The IBM Sterling Store Inventory Management supports only one product class and inventory UOM.

---

## Inventory Moves Implementation

The `getNodeInventory` API returns the available quantity, Item ID, UOM, and product class.

Any additional configuration is not required for the Inventory Move task.

---

## Inventory Moves Reference Implementation

This section explains the reference implementation for Inventory Moves, provided as a part of the IBM Sterling Store Inventory Management.

The following table lists the configuration for Inventory Moves.

*Table 6. Inventory Moves*

Reason Code	Description
STORE_MOVE	Defines the adjustment reason code for inventory update caused due to inventory move operation.

---

## Chapter 7. Inventory Audits

---

### Inventory Audit

The IBM Sterling Store Inventory Management provides the capability for a store associate to search and view all inventory transactions performed in a store. In addition, comprehensive visibility to inventory audits such as modification reason, modification type, and adjusted quantity are also provided.

---

### Inventory Audit Solution

A store associate can view details for a particular modification type, date range, reason, location, and modifications performed by any particular user. A user can also view the item details for a particular item. The option to search for inventory by location is applicable only for those stores that have one or more locations.

Audits are written in IBM Sterling Multi-Channel Fulfillment Solution for each transaction along with appropriate reason codes. Based on the selected modification type, the Inventory Audit Details panel displays the audit information. For example, if the modification type is RECEIPT, details such as the BOL number, item description, receipt number, order number, shipment number, carrier and carrier service code displays. The default transaction type includes RETURN, RECEIPT, SHIPMENT, and ADJUSTMENT.

---

### Inventory Audit End-User Impact

None.

---

### Inventory Audit Implementation

Additional configuration is not required for Inventory Audit.

---

### Inventory Audit Reference Implementation

This section explains the reference implementation for Inventory Audits, provided as part of the IBM Sterling Store Inventory Management.

The following tables lists all configurations provided for Inventory Audits.

*Table 7. System Adjustments*

Reason Code	Description
STORE_MOVE	The reason code for inventory updates caused due to an inventory move operation.
STORE-PHYSICAL-COUNT	The reason code for inventory updates caused due to a physical count operation.
STORE-CYCLE-COUNT	The reason code for inventory updates caused due to a cycle count operation.
BLIND	The reason code for blind receiving.
SHIP	The reason code used when confirming a shipment.

*Table 7. System Adjustments (continued)*

<b>Reason Code</b>	<b>Description</b>
RECEIPT	The reason code used while receiving inventory.

*Table 8. Manual Adjustments*

<b>Reason Code</b>	<b>Description</b>
RE-CLASSIFY	Used by a store associate to adjust inventory that was incorrectly classified.
FOUND-NEW	Used by a store associate when inventory is found in the location, but not recorded in the system.
MISSING	Indicates that the inventory is missing.
SCRAP	Indicates if the inventory has been damaged and is scrapped.

---

## Chapter 8. Counts

---

### Counts

A count system allows you to execute counts in a planned or ad hoc manner. Counts are typically done to eliminate mismatches between the system and the actual inventory. The types of count processes that may be followed are:

- Cycle Count
- Physical Counts

Cycle count ensures that a count is performed periodically in the system. Count tasks may be either generated for a cycle count program or an ad hoc request.

Physical Counts are performed to synchronize the physical and system inventory quantity. During the physical count process, no other activity is performed in the store. This process is usually followed as a year-end activity.

This chapter explains the count processes and solution offered by the IBM Sterling Store Inventory Management.

---

### Cycle Count

Cycle count is an inventory management procedure where a small subset of inventory is counted on a given day. Cycle Count is performed periodically to maintain inventory accuracy.

Count requests are generated on a periodic basis. Count tasks are generated for count requests that are created, and later executed through count sheets.

The procedures available for count include the ability to generate tasks based on the items.

- Count tasks are generated either ad hoc or through the system.
- The store associate performs the count tasks.
- If variances are found as a result of the count, the store associate can:
  - Accept the variance
  - Rectify the count entry

### Cycle Count Process

A count program is set up to facilitate counting.

- The count program provides a list of items on a daily basis.
- The store associate may select all or a subset of items from the list that needs to be counted for that day.
- The store associate can also add items to the count sheet.
- Count sheets are printed for the selected list of items.
- Store associates count inventory in the locations.
- The store associate enters the count sheet number and records the count in the system.
- If the count does not match, the store associate recounts the inventory in that location.

- Variances, if any, are inspected, corrected, and accepted.

## Cycle Count Solution

Cycle Count in the IBM Sterling Store Inventory Management involves the following processes:

### Printing Count Worksheets

A store associate can print count worksheets for a system-generated count request or reprint count sheets. You can also create count requests and then print count sheets for these requests.

To print a count sheet, the `manageCountSheet` API is called. This API is used:

- To create a new batch or update an existing batch with new count requests.
- To create count requests, tasks, and batch, if batch and count request do not exist.
- If batch exists, but there is no count request. The API creates count requests and tasks, and updates the batch.

*Table 9. Printing Count Sheet Validation*

Validation	Result
If you enter the incorrect count sheet number to reprint	When you click the Print button, an error message displays.
If the number of items to count is greater than the number of items pending count	When you click the Print button, an error message displays.
If you do not select any items from the list	When you click the Print button, an error message displays.

For more information about cycle count worksheets, see the *Sterling Store Inventory Management: User Guide*.

### Requesting a Count by Item

The Sterling Store Inventory Management enables you to generate ad hoc count requests. When you enter the identifier of the item, the system checks to verify whether a count request is already open for that item. When you leave the Item ID field, the `getCountRequest` API is invoked. A template to return the total number of records is set. If the total number of records is greater than zero, an error message displays.

*Table 10. Requesting Count By Item Validations*

Validation	Result
If you try to print a count sheet without selecting an item	An error message displays.
If you enter an incorrect item ID	When you press the tab key, an error message displays.
If the item that you entered is already put on count	When you press the tab key, an error message displays.



## Recording the Count

The user needs to update the count details from the count worksheet in the system. After entering the count sheet number, the count sheet with all item details displays. The `getNodeInventory` API is invoked to get the item details. The `recordCountSheet` API is called to record the count results for existing tasks. It accepts the count result by accepting the variance. If new items are found at a location, the API creates count tasks and updates the batch.

Table 11. Recording Count Validation

Validation	Result
If numbers are not entered as the count quantity	An error message displays.
If you enter an incorrect item ID	When you press the tab key, an error message displays.
If the record is confirmed without entering the count quantity for some counts	A warning message displays. The count is assumed to be zero.

## Cancelling a Count

There may be many reasons for which a store associate wants to cancel all pending counts. In such situations, upon invoking the `getCountRequestList` API, a list of all pending counts is provided. The `cancelCountRequest` API is called to cancel all counts. When the user clicks the Cancel button, permissions are set at the action level.

## Viewing Count Archives

You can view count archive information present in one or multiple count sheets that are closed or cancelled. This may be useful when you want to analyze counts performed on a particular item or counts that are cancelled. When invoked, the `getCountRequestList` API provides a list of all counts that are closed or cancelled. A purge criteria can be set to view count archives. The user can view count sheets only up to the duration set in the purge criteria. For example, if the users set the purge criteria for 6 months, they can view the count archive only for the last 6 months.

## Count Summary

A store associate can view a list of count details for all count tasks performed in a day.

- To get manual counts, invoke the `getCountRequestList` API by specifying the current date and `countProgramName=""`.
- To get system assigned counts, invoke the `getCountRequestList` API by specifying the current date and `countProgramName!=""`.
- To get pending counts, invoke the `getCountRequestList` API by specifying the date range until the previous day and status as "Open".
- To get the number of counts completed and count variance for the day, invoke the `getCountResultList` API by specifying the current date.

## Variance

A store associate may want to view variances for the current day and request a recount. A recount involves putting items to count manually. The `getCountRequestList` API is called to get all items having a variance.

For more information about cycle count variances, see the *Sterling Store Inventory Management: User Guide*.

The following configuration rules are set up for cycle counts:

- Adjustment Reasons  
The adjustment reason code configured as part of the Sterling Store Inventory Management for cycle count is STORE-CYCLE-COUNT. This reason code is used to update inventory changes due to the cycle count.
- Cancellation Reasons  
Cancellation reason code is used when a store associate needs to cancel all pending counts. The count cancellation reason provided by the Sterling Store Inventory Management is STORE-INCOMPLETE-COUNTS.
- Task types  
The COUNT task type is provided for Cycle Count.
- Count Request types  
The DEFAULT count request type is provided for Cycle Count.

## Cycle Count End-User Impact

None.

## Cycle Count Implementation

When invoked, the `getCountRequestList` API provides a list of count requests that are generated. The `getTaskList` API provides the task list for all count requests. Count tasks are created for an item and location. The `createBatch` API creates the count sheets for the given tasks.

## Print Cycle Count Worksheet

The service component `PrepareXMLforAPI` prepares XML for invoking multi API `GetBatchAndPrinter`. This multi API calls the `getBatchDetails` and `getPrinter` APIs. The service `PrepareXMLForPrinting` prepares the input XML for printing a count sheet. The `PrintCycleCountWorksheet` service prints the count sheets. This is illustrated by the following Figure.

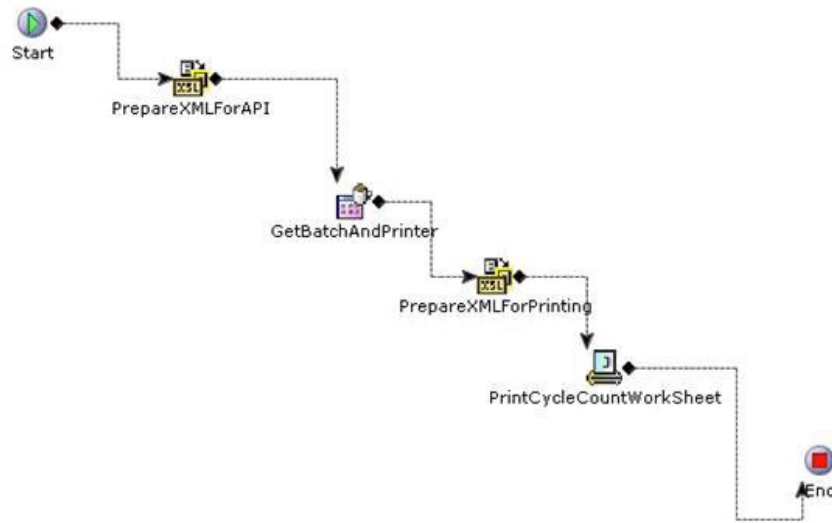


Figure 1. Printing Cycle Count worksheet

To print a cycle count worksheet or variance report:

1. Configure a printer for the store.
2. Add the StorePrinterAlias parameter to the device attributes of the printer and assign the printer name.
3. Associate the printer to print the cycle count worksheet and variance report.
4. Select “Use IBM Sterling Multi-Channel Fulfillment Solution Resource Bundle” for the jasper print component of the service.
5. Specify the number of copies to print for the jasper print component.

## Configuring Agents

The agents that need to be configured for count tasks are:

- EXECUTE\_COUNT\_PROGRAM  
This agent creates count requests based on the count program defined.
- CREATE\_COUNT\_TASKS\_79  
This agent creates new count tasks from the count requests that have been generated.

## Cycle Count Reference Implementation

This section explains the reference implementation provided by the IBM Sterling Store Inventory Management for Cycle Counts.

### Count Program

A default count program is provided. Based on the Count Program Condition settings, count tasks are generated.

The Count Program Condition provided by the Sterling Store Inventory Management is Count items in a store 4 times a Year.

## Adjustment Reason Code

The adjustment reason code provided for cycle count is:

*Table 12. Adjustment Reason Code*

Reason Code	Description
STORE-CYCLE-COUNT	Defines the adjustment reason code for inventory change due to cycle count.

## Cancellation Reason Code

The Count Cancellation reason code provided for cycle count is:

*Table 13. Cancellation Reason Code*

Reason Code	Description
STORE-INCOMPLETE-COUNTS	Indicates the cancellation reason for cycle counts. The user cannot change this value. The system inserts this value as part of the store configurator modification or store synchronization process.

## Count Task Type

The Count Task Type provided for cycle counts is:

*Table 14. Count Task Type*

Reason Code	Description
COUNT	Indicates the task type for cycle counts. The user cannot change this value. The system inserts this value as part of the store configurator modification or store synchronization process.

## Count Request Type

The Count Request Type defined for cycle count is:

*Table 15. Count Request Type*

Reason Code	Description
DEFAULT	Indicates the count request type for cycle counts. The user cannot change this value. The system inserts this value as part of the store configurator modification or store synchronization process.

## Pipeline Configuration

To support the Cycle Count process, a pipeline has been defined. The following figure illustrates the Cycle Count Pipeline.

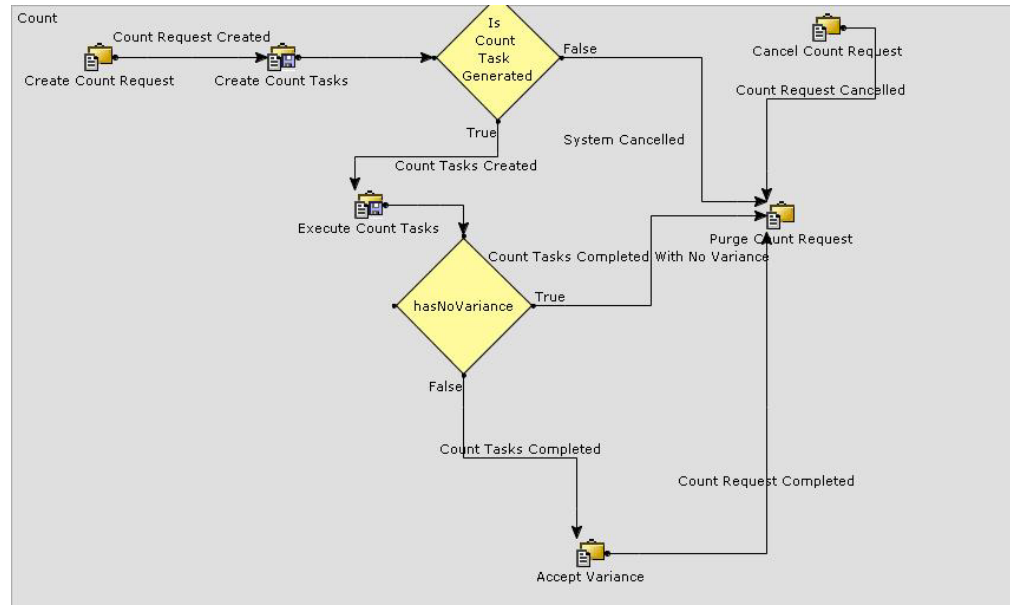


Figure 2. Cycle Count Pipeline

## Physical Counts

The Physical Count process is performed in a store to synchronize the physical and system inventory quantity. During this process, no inventory transaction is performed in the store, except for recording the count quantity. Typically, the physical count process is performed once a year, and on all items in the store.

### Physical Count Process

The Physical Count process is:

1. The store manager starts the physical count.
2. A count request is generated.
3. A store associate performs the count.
4. A store associate records the count details such as count sheet number, item, location, and item quantity in the system.
5. The system creates a count task for each item and location, and adds it to a batch. Each batch represents a count sheet.
6. The system records the count and calculates variance.
7. If variances are found as a result of count, a store associate can:
  - Accept the variance and rectify the count entry error for selected count sheets.
  - Initiate a recount on selected count sheets.
8. After accepting the variances, the store manager can print one or more physical count reports.
9. The store manager ends the physical count process.

### Physical Count Solution

A Physical Count is a process where a store stops all inventory transactions, and physically counts its entire inventory.

## Starting Physical Counts

Before starting the physical count process in a store, you must ensure no inventory transaction is performed, except for recording count. Starting a Physical Count process is permission controlled. The `startPhysicalCountForStore` API is called, which invokes `createCountRequest`, `releaseCountRequest`, and `startPhysicalCount` APIs.

The validations performed before the start of a physical count are:

Table 16. Start Physical Count Validations

Validation	Result
If the physical count process has already started	The Start Physical Count button is disabled.
On clicking the Start Physical Count button	A warning message displays.

## Recording Physical Counts

A store associate records the count in a worksheet that is printed using an external system. After finishing the count, you need to record the count details in the system. The `recordPhysicalCountSheet` API is called to record the physical count details.

**Note:** In Release 7.1 HF13 and higher, the `yfs.ui.countRecords` property is provided in the `yfs.properties` file in order to generate the desired number of lines in the count sheets when performing a physical count.

## Viewing the Physical Count Status

The logic used to display the missing count sheets is:

- The system finds the lowest count sheet number recorded.
- The system finds the highest count sheet number recorded.
- All count sheets missing within this range display. The individual count sheet numbers display only if the range of missing count sheets is greater than 5.

The `getPhysicalCountStatus` API is called to check if a physical count is running. If the physical count has started, the `getCountRequestList` API is used to get count requests. For each count request, the `getBatchList` API is called to identify the count sheets.

For more information about physical count variance report, see the *Sterling Store Inventory Management: User Guide*.

## Searching Count Sheets

The `getBatchList` API is called to retrieve all count sheets or the specified count sheet.

## Viewing Variance

You can view variances during a physical count, when accepting the variance or during an audit inquiry.

The `getCountResultListForStore` API is invoked to get all items with a variance. To display the data at the item location level, the `getCountRequestList` API is called. This API returns the count result at item, location level.

The IBM Sterling Store Inventory Management provides pagination for all Variance screens and the Count Sheet Search screen.

## Rectifying Counts

The `getItemUOMList`, `getCountList`, and `getTaskList` APIs are called to display the count sheet.

Upon confirming, the `recordPhysicalCountSheet` API is invoked.

Upon clicking Void, the `recordPhysicalCountSheet` API is invoked and the value of void is set to "Y".

## Cancelling Physical Counts

To cancel a physical count, a multi API is invoked. The multi API comprises of `cancelPhysicalCount` and `endPhysicalCount` APIs.

## Accepting Variances

To accept variances, the `acceptVarianceForPhysicalCount` API is called. This API gets all missing items and creates count tasks for each missing item. The results of these count tasks are set to zero and all location level tasks are cancelled. Now, the `acceptVariance` API is invoked.

## Ending Physical Counts

After accepting a variance, you can end the physical count by clicking the End button. The `endPhysicalCount` API is invoked to end the physical count.

## Audit Inquiry

An Audit Inquiry displays the audit for the last completed physical count. The `getCountRequestList` API is invoked to get the current count request.

*Table 17. Audit Inquiry Validation*

Validation	Result
If no physical count is performed	An error message displays.

The configuration rules set up for physical count are:

- **Adjustment Reasons**

The adjustment reason code configured as part of the IBM Sterling Store Inventory Management for physical count is `STORE-PHYSICAL-COUNT`. This reason code is used to update inventory change due to physical count.

- **Cancellation Reasons**

The cancellation reason code is used when a store associate wants to cancel a physical count that was started accidentally. The count cancellation reason provided by the IBM Sterling Store Inventory Management is STORE-INVALID-COUNT.

- **Task types**

The task type provided by the IBM Sterling Store Inventory Management for physical count is PCCOUNT.

- **Count Request types**

The count request type for physical counts is PHYSICAL-COUNT.

## Physical Count End-User Impact

None.

## Physical Count Implementation

The SOP\_PhysicalCount\_Varianc\_1.0 service invokes the multi API. This multi API comprises of getPrinter, getMissingItems, and getCountResultListforStore APIs. This service prepares the input XML for jasper print and prints the variance report as illustrated in the following figure.

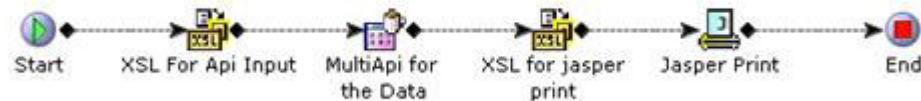


Figure 3. SOP\_PhysicalCount\_Varianc\_1.0

To print a variance report:

- Configure a printer for the store.
- Add the StorePrinterAlias parameter to the device attributes of the printer, and assign the printer name.
- Associate the printer to print the variance report.
- Select “Use IBM Sterling Multi-Channel Fulfillment Solution Resource Bundle” for the jasper print component of the service.
- Specify the number of copies to print for the jasper print component.

## Physical Count Reference Implementation

This section explains the reference implementation provided by the IBM Sterling Store Inventory Management for physical counts.

### Adjustment Reason Code

The Sterling Store Inventory Management adjustment reason code for physical count is:



Table 18. Adjustment Reason Code

Reason Code	Description
STORE-PHYSICAL-COUNT	Defines the adjustment reason code for inventory change due to a physical count.

## Cancellation Reason Code

The Sterling Store Inventory Management count cancellation reason code for physical counts is:

Table 19. Cancellation Reason Code

Reason Code	Description
STORE-INVALID-COUNTS	Indicates the cancellation reason for physical count. The user cannot change this value. The system inserts this value as part of the store configurator modification or store synchronization process.

## Count Task Type

The Sterling Store Inventory Management count task type for physical count is:

Table 20. Count Task Type

Reason Code	Description
PCCOUNT	Indicates the task type for physical count. The user cannot change this value. The system inserts this value as part of the store configurator modification or store synchronization process.

## Count Request Type

The count request type defined for physical counts is:

Table 21. Count Request Type

Reason Code	Description
PHYSICAL-COUNT	Indicates the count request type for physical count. The user cannot change this value. The system inserts this value as part of the store configurator modification or store synchronization process.

## Pipeline Configuration

The following figure illustrates the Physical Count Pipeline.

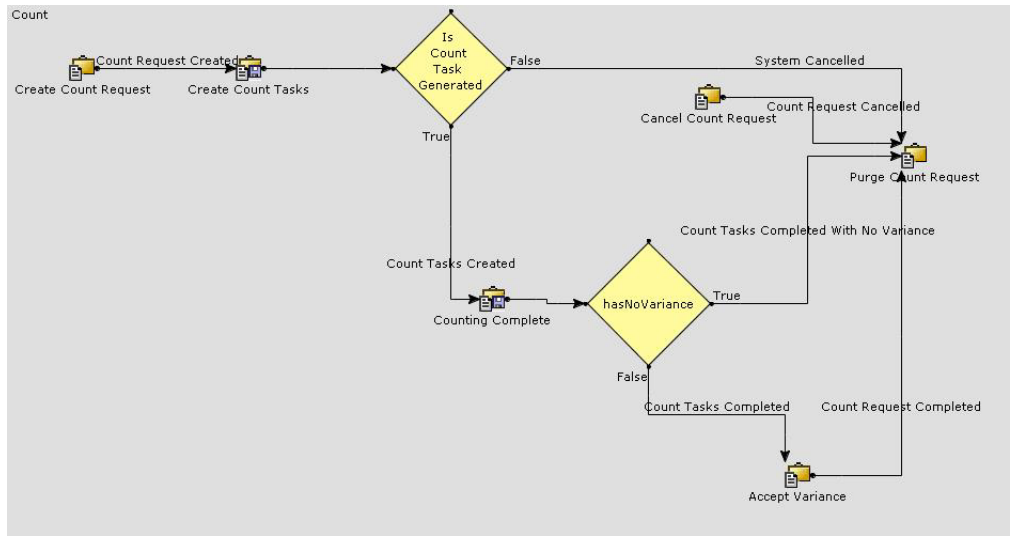


Figure 4. Physical Count Pipeline

---

## Chapter 9. Receiving

---

### Receiving

The IBM Sterling Store Inventory Management enables you to control the receiving process with the information that is available in a store. The Sterling Store Inventory Management supports receiving inventory through paper-based or mobile devices.

This chapter explains the receiving process and solution offered by the Sterling Store Inventory Management.

---

### Receiving Processes

The Receiving processes followed in a store are:

- Receipts made for an order or a shipment:  
Inventory is received from other stores, distribution centers (DC), or vendors.
- Receipts made for Blind Receiving:  
Inventory is received when no BOL or order number exists for the inventory in the system.

#### Print Receiving Worksheets

The Print Receiving worksheet process is:

- The receiving worksheet is printed based on the inbound order or shipment.
- The store associate records receipt information in the receiving worksheet.
- The store associate records the receipt details in the console.

#### Record Receiving

The Record Receiving process is:

- The store associate can record receiving worksheet details based on the BOL number or order number.
- The store associate can record additional items that are listed in the order, and may be present in some other shipment of the same BOL.
- The IBM Sterling Store Inventory Management provides the ability to select all records if no discrepancy exists while receiving against an order or shipment.

#### Blind Receiving

The Blind Receiving process is:

- The store associate receives inventory with no BOL or order number.
- The inventory details are recorded on a blank worksheet. The blank worksheet is printed using an external system.
- The store associate records details from the worksheet in the system and clicks the Confirm button.

---

## Receiving Process Solution

This section explains the various configurations that may be completed during the Receiving process.

The configuration rules that need to be set up for receiving are:

- Receiving preferences
- Receiving Discrepancies provided by the IBM Sterling Store Inventory Management
- The disposition codes configured by the Sterling Store Inventory Management

### Print a Receiving Worksheet for Expected Inbound Shipments

The user can print a receiving worksheet for a shipment based on the BOL number. The `getShipmentList` API is invoked to get a list of shipments associated with the BOL. For more information about receiving worksheets for a shipment, see the *Sterling Store Inventory Management: User Guide*.

Table 22. Validations Performed

Validations	Result
If you enter an incorrect BOL number	Upon clicking the Print button, an error message displays.
If the BOL# that you entered does not have allowed shipments for which inventory can be received	Upon clicking the Print button, a warning message displays.
If the BOL# does not exist in the IBM Sterling Multi-Channel Fulfillment solution	Upon clicking the Print button, error message displays.

### Select Shipment for the BOL or Order Number Entered

The `getShipmentList` API provides a list of shipments for the entered BOL or order number. The print service is called to print the receiving worksheet for the selected shipment.

### Print a Receiving Worksheet for Expected Inbound Orders

The user can print a receiving worksheet for a known order. The `getShipmentList` API is invoked to get a list of shipments associated with the order number. For a list of validations performed, see Table 1. For more information about receiving worksheet for an order, see the *Sterling Store Inventory Management: User Guide*.

### Search for the Expected Shipments

The `getOrganizationList` API provides a list of DCs. Based on the search criteria, the `getShipmentList` API is invoked.

Table 23. Validations Performed

Error Case	What Happens?
If you enter an incorrect store or vendor	An error message displays.
If you enter an incorrect item ID	An error message displays.
If you enter the From date greater than the To date	An error message displays.

## Search for the Expected Order

Based on the search criteria, the user can search receipts for an order. If shipments are not present for the order, a worksheet is printed based on the order details. The getOrderList API is used to list the order. For a list of validations performed, see Table 2.

## Recording Receipt of a Shipment

The user can record the quantity received from the receiving worksheet. The getShipmentList API is called to check if the BOL number exists for the shipment. If multiple shipments exist for the BOL, in the Shipment Selection screen, select the required shipment. The getShipmentDetails API displays the Record screen. Once the user records the receiving details, on clicking the Confirm button the recordReceiptInOneStep API is invoked for recording the details. This API records the discrepancy.

**Note:** The summation of ordered quantity - (minus) the summation of received quantity must be zero to result in a receipt with no discrepancy.

When the user checks the Receipt Complete flag, the discrepancy is calculated and the shipment is received. After recording the receiving worksheet for a shipment, a discrepancy report is printed. For more information about receipt discrepancy reports, see the *Sterling Store Inventory Management: User Guide*.

Table 24. Validations Performed

Error Case	What Happens?
If you enter an incorrect BOL #	Upon clicking the Proceed button, an error message displays.
If you enter a BOL # that does not have allowed shipments for which inventory can be receipt	Upon clicking the Proceed button, an error message displays.
If you enter an invalid item ID	An error message displays.

## Recording Receipt of an Order

The user can record inventory received against an order. The getShipmentList API is called to check if any shipment exists for the order. If shipments do not exist for the order, the getOrderDetails API is called to allow the user to record against the order. The Sterling Store Inventory Management creates a dummy shipment for an order without shipments. Once the user records the receiving details, upon clicking the Confirm button the recordReceiptInOneStep API is invoked for recording the

details. This API records the discrepancy. When the user checks the Receipt Complete flag, the discrepancy is calculated and order is received.

**Note:** The summation of ordered quantity - (minus) the summation of received quantity must be zero to result in a receipt with no discrepancy.

For a list of validations performed, see Table 3. After recording the receiving worksheet for an order, a discrepancy report is printed. For more information about receipt discrepancy reports, see the *Sterling Store Inventory Management: User Guide*.

## Recording Blind Receipts

The user can record blind receipt. Once the user records the receiving details, upon clicking the Confirm button the recordReceiptInOneStep API is invoked for recording the details.

Table 25. Validations Performed

Error Case	What Happens?
If you enter an invalid store code	An error message displays.
If you enter an invalid item ID	An error message displays.

## Searching for Inbound Shipments

You can view shipments that are already received and that can be received. The getOrganizationList API provides a list of all DCs. After you enter the search criteria, the getShipmentList API is invoked. The getShipmentDetails API displays details for the selected shipment. For a list of validations performed, see Table 2.

---

## Receiving Process End-User Impact

None.

---

## Receiving Process Implementation

You can print a receiving worksheet, receiving discrepancy report, or a customer label for an order or shipment.

To print a worksheet, discrepancy report, or customer label:

1. Configure a printer for the store.
2. Add the StorePrinterAlias parameter to the device attributes of the printer, and assign the printer name.
3. Associate the printer to print the receiving worksheet, receiving discrepancy report, or a customer label.
4. Select "Use IBM Sterling Multi-Channel Fulfillment Solution Resource Bundle" for the jasper print component of the service.
5. Specify the number of copies to print for the jasper print component.

---

## Receiving Process Reference Implementation

This section explains the reference implementation provided by the IBM Sterling Store Inventory Management for Receiving.

### Receiving Pipeline

The following figure illustrates the Receiving Pipeline.

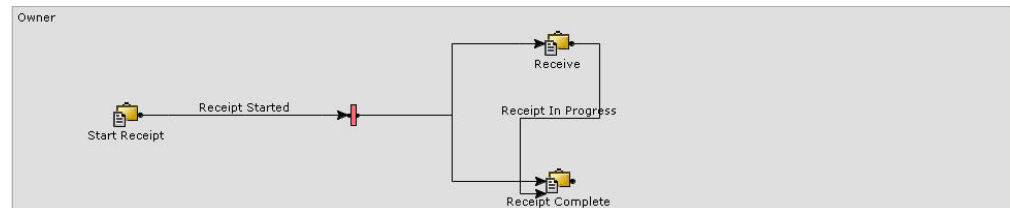


Figure 5. Receiving Pipeline

### Receiving Discrepancies

The discrepancy codes defined in the Sterling Store Inventory Management are:

- OVER
- SHORT
- DAMAGE

### Node Disposition

The disposition codes configured in the Sterling Store Inventory Management is QC-PASSED.





---

## Chapter 10. Managing Locations

---

### Managing Store Locations

A store manager can manage back room locations in a store. This is applicable only to those stores that can track inventory in multiple locations within the store.

---

### Store Location Solution

In a store modelled with multiple locations, in addition to the two locations set as part of the configuration, a store associate can create additional locations. All locations created in the store are regular locations. The store associate can also delete existing locations.

---

### Store Location End-User Impact

Before deleting a location or remodeling a store, the store associate must ensure that inventory does not exist in the location. For example, consider a model store with multiple locations. Before this store is modelled as a store with one location, the administrator needs to move inventory from locations that are not defined in the new store.

---

### Store Location Implementation

The getLocationList API is used to call the existing locations in a store. The deleteLocation API is used to delete a location. The createLocation API is used to create a location.

---

### Store Location Reference Implementation

This section explains the reference implementation for managing locations, provided as part of the IBM Sterling Store Inventory Management.

#### Store With Multiple Locations

The configuration provided by the Sterling Store Inventory Management is:

*Table 26. Store with Multiple Locations*

Store ID	Store Type
Matrix_MODEL_MULTIPLE_LOCATION	Model Store
Matrix_S4	Follower store



---

## Chapter 11. Shipping

---

### Shipping

Shipping includes the processes from the point when shipments are created up to the point when shipments are picked up by a customer or loaded into trailers.

Sometimes the price of the items may have changed since the customer purchased the items. The store associate must be able to apply the latest price on the items.

Shipping involves:

- Printing Pick Tickets
- Searching for Shipments
- Recording Back Room Picks
- Recording Customer Picks
- Creating Loads
- Adding to a Manifest
- Closing a Manifest

---

### Printing Pick Tickets

Pick ticket is a sheet containing the list of items that need to be pulled from inventory to fill an order.

#### Pick Ticket Solution

Pick tickets can be printed for a particular shipment. If pick tickets are printed for all pending shipments or orders, the `getShipmentList` API is called. This API checks to see if the status is less than the shipment shipped and pick ticket is not printed for the shipment.

#### Pick Ticket End-User Impact

The search criteria entered by the user to search for a shipment is case-sensitive.

You may want to create store-specific pipelines and not use the reference implementation provided by the IBM Sterling Store Inventory Management.

To create a pipeline for outbound execution, follow these steps:

1. Log in to the IBM Sterling Multi-Channel Fulfillment solution as the Enterprise user.
2. From the Sterling Multi-Channel Fulfillment Solution Consoles menu bar, select Configuration > Launch Sterling Store Inventory Management Configurator. The Sterling Store Inventory Management Configurator displays.
3. From the Configuration Setup screen, select Configure Outbound Process Modeling > Configure Shipping Process.
  - a. In the Store condition group, enable the ON\_SUCCESS event of the CREATE SHIPMENT transaction.
  - b. Call the Store Change Shipment Status action.

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

## Pick Ticket Implementation

This section explains the various implementations required for shipping.

### Process Modeling

The ON\_SUCCESS event of CREATE SHIPMENT transaction calls, the SOP\_ChangeStatus\_Backroom\_1.0 service. This service calls the getRuleDetails API to provide rule details for the back room pick. If the flag is set to "Y", the CHECK\_BACKROOM\_PICK\_REQD.0001.ex (BaseTransactionKey="Change\_Shipment\_Status") transaction changes the status of the shipment to Ready For Backroom Pick. Otherwise, the shipment is in the Ready For Customer status.

For the BACKROOM\_PICK (BaseTransactionKey=" Change\_Shipment\_Status") transaction, if the Pickup Status is Ready For Backroom Pick, the drop status is Ready For Customer.

## Pick Ticket Reference Implementation

This section explains the reference implementation for shipping, provided as part of the IBM Sterling Store Inventory Management.

### Pipeline Configuration

The following figure illustrates the in shipping pipeline.

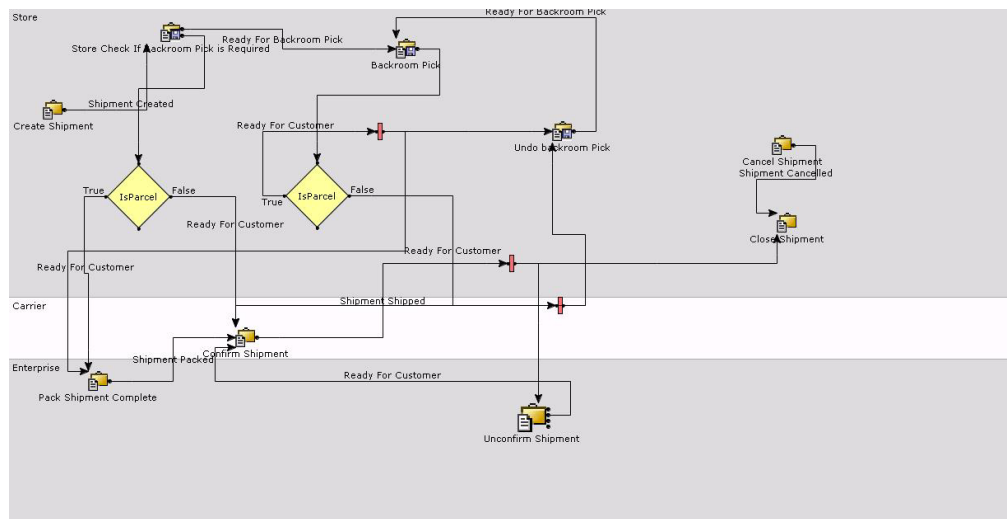


Figure 6. Shipping Pipeline

The Check if backroom pick is required transaction is introduced between the Shipment Created and Ready For Backroom Pick statuses. If a back room pick is not needed, the flow goes to Ready For Customer Pick status.

The Backroom Pick transaction is introduced between the Ready For Customer Pick and Ready For Backroom Pick statuses.

---

## Searching for Shipments

The store associate can search for shipments based on certain search criteria. Along with the list of shipments that match the search criteria, the store associate can view other details related to the shipments.

### Shipments Search Solution

The IBM Sterling Store Inventory Management provides screens for searching shipments. The screen also provides related tasks pertaining to the shipment console.

#### Solution for Searching for Shipments

The `getShipmentList` API is used to retrieve a list of all the shipments.

#### Shipment Details

The store associate can view the shipment details such as items in the shipment, quantity of items, unit price of the item, and line price for the selected shipment. The `getShipmentDetails` API retrieves the shipment details.

#### Undo Back Room Pick

Sometimes, after performing the back room pick, the customer may decide not to pick up the order. The store associate needs to undo the back room pick process so that the items can be moved back. The `getCommonCodeList` API provides the reason codes to undo Back Room pick. Once the store associate clicks the Undo button, the `undoPick` API is called to undo the back room pick.

*Table 27. Validations Performed*

Error Case	What Happens?
If the shipment status is not valid for undo back room pick	An error message displays.

#### Viewing Shipment Audits

You may want to track the changes done to a shipment. The Shipment Audit screen provides details of the changes done to a shipment. The `getAuditList` API provides a list of reason codes.

*Table 28. Validations Performed*

Error Case	What Happens?
If the user does not select a shipment and clicks View Shipment Audit	An error message displays.

#### Changing a Shipment Container

You can add, modify, or delete containers of a shipment.

- To add containers to a shipment:
  - The `addToContainer` API is called for stores integrated with IBM Sterling Multi-Channel Fulfillment solution.

- The changeShipment API is called for stores that are not integrated with Sterling Multi-Channel Fulfillment Solution.
- To modify containers of the shipment:
  - The changeShipmentContainer API is called for stores integrated with Sterling Multi-Channel Fulfillment solution.
  - The changeShipment API is called for stores that are not integrated with Sterling Multi-Channel Fulfillment solution.
- To delete a container from the shipment:
  - The changeShipmentContainer API is called for stores integrated with Sterling Multi-Channel Fulfillment solution.
  - The changeShipment API is called for stores that are not integrated with Sterling Multi-Channel Fulfillment solution.
- If the container is added to a manifest, the removeContainerFromManifest API is called.

Table 29. Validations Performed

Error Case	What Happens?
If the user clicks on modify container information without selecting the shipment	An error message displays.

## Removing Shipment From Manifest

Sometimes, a store associate may want to remove shipments from the manifest. This may be due to containers being damaged in the shipment. In such situations, the removeContainerFromManifest API is called to remove the shipment.

Table 30. Validations Performed

Error Case	What Happens?
If the user clicks on remove shipment from manifest without selecting the shipment	An error message displays.
If shipment is not added to the manifest	An error message displays.

## Undo BOL

The store associate may want to remove shipments from a load due to many reasons like non-availability of carriers, items being damaged, or change in the ship date. In such situations, the manageLoad API is called to undo BOL.

Table 31. Validations Performed

Error Case	What Happens?
If user clicks on Undo BOL without selecting a shipment	An error message displays.

## Shipments Search End-User Impact

None.

## Shipments Search Implementation

None.

## Shipments Search Reference Implementation

None.

---

## Recording Back Room Picks

The movement of items from back room to the customer service area needs to be tracked in a store with multiple locations. Based on the store configuration, the IBM Sterling Store Inventory Management supports recording of back room picks for a store with or without locations.

### Back Room Picks Solution

During the recording of the a back room pick, the changeShipment API is called to record the picked quantity. The changeShipmentStatus API is called when the shipment status needs to be changed from “Ready for Backroom Pick” to “Ready for Customer”. The moveLocationInventory API is called to move inventory from the back room pick location to customer pick location. During back room pick, if no items are picked for any of the shipments, the shipment is cancelled. The changeShipment API is invoked and the value of the Cancelled Shipment flag is set to “Y”.

If there is a shortage due to inadequate inventory, the remaining items are backordered. Backordering enables you to track the shortage of inventory at a particular location. When inventory shortage is selected, the changeShipmentStatus and moveLocationInventory APIs are called. The changeShipment API is also called in which the value of the BackOrder flag is set to “Y”. If you select Will Pick Later, the shipment status does not change. The changeShipment and moveLocationInventory APIs are called. If you select Shortage resolution varies by line, the changeShipmentStatus and moveLocationInventory APIs are called. The changeShipment API is also called in which the value of the BackOrder flag is set to “N”.

### Barcode Configuration

Barcode configuration is done to get the default quantity for items that you scanned during the back room pick operation.

## Back Room Picks End-User Impact

None.

## Back Room Pick Implementation

This section explains the various implementations required for recording back room pick.

### Statuses

Ready For Customer (1100.70.06.30) and Ready For Backroom Pick (1100.70.06.10) are the statuses provided by the IBM Sterling Store Inventory Management. It is recommends you to not delete these statuses.

## Barcode Implementation

This section explains how to set the Barcode configuration.

To set the Barcode configurations:

1. Log in to the IBM Sterling Multi-Channel Fulfillment solution as the Enterprise user.
2. From the Sterling Multi-Channel Fulfillment Solution Consoles menu bar, select Configuration > Launch Sterling Store Inventory Management Configurator. The Sterling Store Inventory Management Configurator displays.
3. From the Configuration Setup screen, select Store Management > Store Settings > Tasks common to all stores.
4. In Tasks common to all stores, select Onboard a store, to configure a new store. For more information about configuring new stores, *Sterling Store Inventory Management: Configuration Guide*.

## Back Room Picks Reference Implementation

This section explains the reference implementation provided as part of the IBM Sterling Store Inventory Management for recording back room pick operations.

### Barcode Configuration

The barcode configuration provided by the Sterling Store Inventory Management are:

Table 32. Barcode Configuration

Reason Code	Description
SCAN_MODE	This is applicable for a store with no location, one location, and three locations.
ENTRY_MODE	This is applicable to all stores.

---

## Recording Customer Picks

It is assumed that a store has only one customer pick location. When a customer walks into a store with proof of purchase, a store associate needs to verify the identity of the customer.

### Customer Picks Solution

In a customer pick scenario, when a customer walks into the store to pick up orders, the store associate must verify the customer identity. There are various criteria based on which the verification is done. You can search for an order using the credit card number, order number, customer name, or phone number. Depending on the search criteria, appropriate APIs are called. If you do not enter the credit card number in the Search screen, the `getshipmentList` API is called. If the credit card number is entered, the `getCreditCardOrderUE` user exit is called for validation. If the credit card validation fails, an error message displays.

During verification of customer details, if you type the credit card number, the last four digits of the credit card number is verified using the `getShipmentDetails` API. For orders not existing in the IBM Sterling Multi-Channel Fulfillment Solution, the credit card validation is done through the `getCreditCardOrderUE` user exit.



A store associate may apply the latest price to the items picked up by the customer. The `getCurrentPriceListUE` user exit is called to get the latest price. The shipment details are passed to this user exit.

Store associates can scan the items to be picked by enabling the scan mode. Scan mode cannot be enabled for stores with multiple locations or when location control is enabled. When scanning items, the `translateBarcode` API is called. This converts the barcode on an item to item ID. Depending on the information, either the Shortage Resolution screen or Confirm screen displays.

A customer may pick up part of an order or the entire order and pick the remaining items at a later time. In that case, the shipment status will not change. The `changeShipment` and `moveLocationInventory` APIs are called. If there is a shortage of ordered items in the store, the `changeShipmentStatus` and `moveLocationInventory` APIs are called. Along with this, the `changeShipment` API is called and the value of the BackOrder flag is set to "Y". If you select Shortage resolution varies by line, the `changeShipmentStatus` and `moveLocationInventory` APIs are called. The `changeShipment` API is also called in which the value of the BackOrder flag is set to "N".

You can confirm an order by calling the `ConfirmCustomerPick` custom API. If the price is overridden, the `changeOrderPriceUE` user exit calls the `changeOrder` API.

## Customer Picks End-User Impact

None.

## Customer Picks Implementation

This section explains the various implementations required for recording customer pick.

### Credit Card Implementation

To validate the credit card number, you must implement the `getCreditCardOrderUE` user exit. This user exit's default implementation gets the order for the entered credit card number. Then the list of orders are looped through and a query is formed to get the shipment list for each order from the `getShipmentList` API.

### Extended Price Implementation

The `getCurrentPriceListUE` user exit is called to get the latest prices of an item. If the returned value of this user exit is more than the price of items when ordered, the original price is maintained. If the returned value is less than the original price, the current price displays. If this user exit is not implemented the ordered price will be the same as the current price.

### Barcode Implementation

#### About this task

This section explains how to set the barcode configuration.

To set the barcode configurations:

#### Procedure

1. Log in to the IBM Sterling Multi-Channel Fulfillment Solution as the Enterprise user.

2. From the Sterling Multi-Channel Fulfillment Solution Consoles menu bar, select Configuration > Launch IBM Sterling Store Inventory Management Configurator. The Sterling Store Inventory Management Configurator displays.
3. From the Configuration Setup screen, select Store Management > Store Settings > Tasks common to all stores.
4. In Tasks common to all stores, select Onboard a store, to configure a new store. For more information on how to configure model store, see the *Sterling Store Inventory Management: Configuration Guide*.

## Customer Picks Reference Implementation

This section explains the reference implementation provided as part of the IBM Sterling Store Inventory Management for recording customer pick.

### Cancellation Reason Codes

The following table lists the cancellation reason codes provided by the Sterling Store Inventory Management when the customer wants to cancel an order.

Table 33. Cancellation Reason Codes

Reason Codes	Description
CN1	Change Of Mind
CN2	Cheaper Price Found
CN3	Late/Failed Delivery
CN4	Unacceptable Delivery Time Promised
CN5	Product No Longer Available
CN6	Fraudulent Order
CN7	Address Undeliverable
CN8	Duplicate Order
CN9	Other

---

## Creating Loads

A customer may want the order to be delivered to a particular address. In such situations, the IBM Sterling Store Inventory Management allows a store associate to consolidate shipments that have common customers or destinations. The store associate can also determine the load's weight and volume.

### Load Solution

Once an order is confirmed for a particular customer, a shipment is created for that order. The shipment can be delivered to the customer through a load or manifest.

You can create a new load or merge shipments to an existing load. Once the shipment number or order number is entered, the `getShipmentList` API is called. This API groups all the shipments added as part of load.

You can consolidate shipments based on the `BillToCustomerId`, `BillToAddressKey`, and `ToAddressKey` attributes. In the Shipment Consolidation screen, all shipments with the common `BillToCustomerId`, `BillToAddressKey`, and `ToAddressKey` attributes display. When you click the Next button, the create load screen appears. If any loads exist with the same `BillToCustomerId`, `BillToAddressKey`, and

ToAddressKey then the load details display in the load consolidation screen. You can either create a new load for the consolidated shipments or include shipments in any of the displayed load.

When a new load is created, the manageLoads API is called with create option. If shipments need to be merged with an existing load, the manageLoads API is called with merge option.

You can determine the number of pallets or cartons for each load. You also need to determine the SCAC and Service for the load from the drop-down list. Only the Truck-Load or Less-Than-Load carrier services are displayed. Once the load is confirmed, the manageLoad API is called. This API updates the load. To confirm the shipment, the receiveIntransitUpdates API is called.

If you save the load, the manageLoad API is called where the value of the generateBOLNo attribute is set to "N".

You can remove shipments from a load by removing the shipment lines. The splitShipment API is called internally. Using the manageLoads API, the shipment is removed from a load.

## Load End-User Impact

If you do not want to use the reference implementation provided by the IBM Sterling Store Inventory Management, you can create store-specific pipelines.

To create a pipeline for outbound execution:

1. Log in to the IBM Sterling Multi-Channel Fulfillment solution as the Enterprise user.
2. From the Sterling Multi-Channel Fulfillment Solution Consoles menu bar, select Configuration > Launch Sterling Store Inventory Management Configurator. The Sterling Store Inventory Management Configurator displays.
3. From the Configuration Setup screen, select Configure Outbound Process Modeling > Configure Shipping Process.
  - a. In the Store condition group, enable the ON\_SUCCESS event of the CREATE SHIPMENT transaction.
  - b. Use the IS STORE condition to conditionally call the SET Bp FLAG action.  
For more information about configuring pipelines, see the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

## Load Implementation

To print a BOL:

- Configure a printer for the store.
- Add the StorePrinterAlias parameter to the device attributes of the printer, and assign the printer name.
- Associate the printer to print BOL.
- Select "Use IBM Sterling Multi-Channel Fulfillment Solution Resource Bundle" for the jasper print component of the service.
- Specify the number of copies to print for the jasper print component.

## Load Reference Implementation

The following figure illustrates the shipment pipeline for a transfer order.

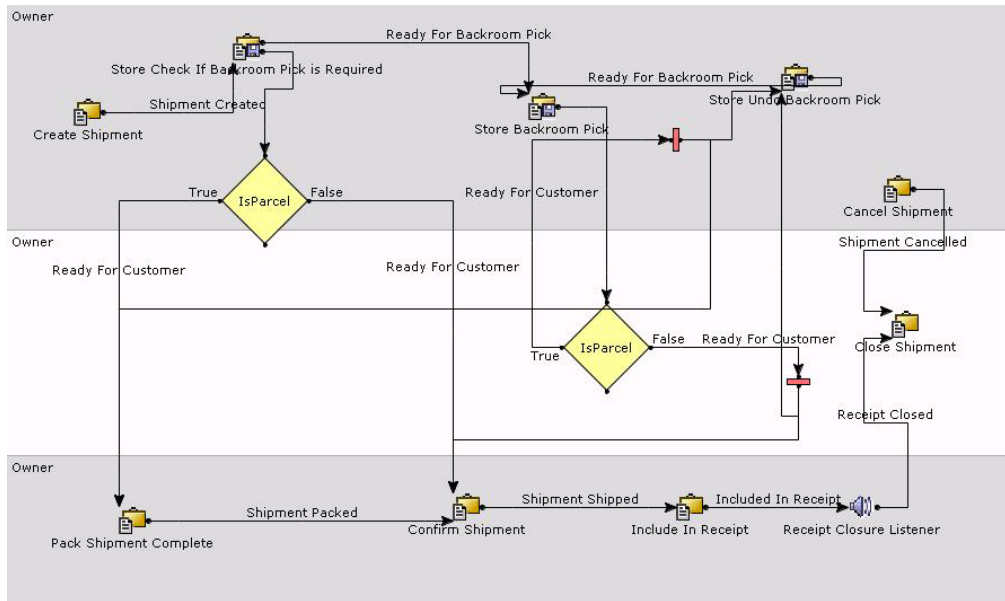


Figure 7. Shipment Pipeline

## Adding to a Manifest

The store associate must create a manifest for parcel manifest shipments. If the store associate wants to select a carrier for the shipment, and if a manifest is already open for that carrier, shipments are added to that carrier manifest. The store associate can close any open manifest when necessary.

### Adding Manifest Solution

When adding a shipment to a manifest, or creating a new manifest, you can select the carrier. Depending on the carrier, the manifest information displays. You can select the SCAC and Services for the shipment from the drop-down list. Only parcel carrier services are displayed. When you change the value of SCAC and Services, the Manifest Found panel refreshes depending on the output of the manifestList API. You can add containers to a manifest using the addContainerToManifest API. You cannot create a new manifest for a carrier, that already has an open manifest.

You can view all open and closed manifests. To display this information, the getManifestList API is called where the shipment status is less than the confirm ship status. When closing the manifest, the manifest is printed for that carrier.

### Adding Manifest End-User Impact

None.

### Adding Manifest Implementation

When executing the closeManifest transaction, the ON\_SUCCESS event prints the manifest. If you do not want to print the manifest, you can disable the ON\_SUCCESS event.

## Adding Manifest Reference Implementation

None.

---

## Closing a Manifest

You can close a manifest once you add a shipment to it. However, IBM Sterling Store Inventory Management also provides the ability to keep the manifest open until the arrival of the truck. Once you close a manifest, it indicates the shipment has been shipped.

## Closing Manifest Solution

The Close Manifest screen displays a list of all the open manifests. The getManifestList API provides the list of all shipments with the status less than the confirm ship status. Once you click the Close Manifest button, the closeManifest API is invoked to close the selected manifest.

## Closing Manifest End-User Impact

None.

## Closing Manifest Implementation

When executing the closeManifest transaction, the ON\_SUCCESS event prints the manifest. If you do not want to print the manifest, you can disable the ON\_SUCCESS event.

## Closing Manifest Reference Implementation

None.



---

## Chapter 12. Alert Console

---

### Alert Console

The Alert management module of the IBM Sterling Store Inventory Management enables a store associate to view and resolve alerts. Alerts may be automatically generated when certain events occur on a system, or when created by users. A store associate can view and resolve alerts that are assigned to them. Alerts are sent to different queues depending on the configured notification definitions. If alerts are assigned to a queue, all users who have subscribed to the queue can view them.

---

### Alert Console Solutions

The IBM Sterling Store Inventory Management provides the ability to create appropriate types of alerts, and ensures that these alerts are processed in a timely and efficient manner.

Using the Alert Type configuration, you can define the alert types. You can use queues to group related alerts. Users can be subscribed to one or more queues and can process alerts from their subscribed queues.

---

### Alert Console End-User Impact

None.

---

### Alert Console Implementation

This section describes how to configure alerts and queues.

#### Configuring Alerts

##### About this task

You can define different alert types and assign them to the SOP\_INFOALERTS queue.

To configure an alert type:

##### Procedure

1. Log in to the IBM Sterling Multi-Channel Fulfillment Solution as the Enterprise user.
2. From the Sterling Multi-Channel Fulfillment Solution Consoles menu bar, click Configuration > Launch IBM Sterling Store Inventory Management Configurator. The Sterling Store Inventory Management Configurator displays.
3. From the Configuration Setup screen, click Configure Alert Management > Define Alert Types.

For more information about configuring alert, see the *Sterling Store Inventory Management: Configuration Guide*.

## Configuring Queues

### About this task

All informational alerts are grouped under the SOP\_INFOALERTS queue. All queues are defined at the Enterprise level. Any additional queues configured for informational alerts need to be grouped under the SOP\_INFOALERTS queue.

To configure queues:

### Procedure

1. Log in to the IBM Sterling Multi-Channel Fulfillment Solution as an Enterprise user.
2. From the Sterling Multi-Channel Fulfillment Solution Consoles menu bar, click Configuration > Launch IBM Sterling Store Inventory Management Configurator. The Sterling Store Inventory Management Configurator displays.
3. From the Configuration Setup screen, click Configure Alert Management > Define Alert Queues.

For more information about configuring queues, see "Configuring Alert Management".

## Triggering the Agent

### About this task

This section provides information on the time-triggered transactions.

### Shipment Monitor

#### About this task

The Shipment Monitor transaction is used to trigger an agent which raises the Delayed Customer Pick alert. You need to run the agent server to trigger an agent.

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

*Table 34. Shipment Monitor Attributes*

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

The following are the criteria parameters for this transaction:

*Table 35. Shipment Monitor 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 will not collect information on the pending jobs for that time-triggered transaction.



Table 35. 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.

## SOP Availability Monitor

### About this task

The SOP Availability Monitor transaction is used to trigger an agent which raises the Inventory Shortage alert. You need to run the agent server to trigger an agent.

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

Table 36. SOP Availability Monitor Attributes

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

The following are the criteria parameters for this transaction:

Table 37. SOP Availability Monitor 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 will not collect information on the pending jobs for that time-triggered 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.
InventoryOrganizationCode	Required. Inventory organization code to use when shipment monitor criteria is passed.
StoreID	Required. If the store identifier is not passed, then the alert is not raised for the store.

---

## Alert Console Reference Implementation

By default, the SOP\_INFOALERTS queue is provided by the IBM Sterling Store Inventory Management to post informational alerts. The default queue provided for system alerts is SOP\_GENERAL\_ALERT.

Alerts, alert types, and queue subscriptions are provided for the Matrix-R organization.

By default, the following alert types are provided:

- SOP\_DELAYED\_CUST\_PICK (DELAYED CUSTOMER PICK) — This exception is raised whenever there is a delayed customer pick.

- SOP\_ADJ\_OVER\_VARIANCE (INVENTORY ADJUSTMENT VARIANCE VALUE ABOVE THRESHOLD VALUE) — This exception is raised whenever the adjustment variance value is above the threshold value.
- SOP\_ITEM\_SHORT (ITEM SHORTAGE) — An exception type is created for item shortage.
- SOP\_GENERAL (STORE GENERAL EXCEPTION) — This queue is provided for posting general alerts.

In the Exception Type Details, the Expiration days and Followup Hours are always set to zero.

---

## Chapter 13. Reference Implementation

---

### Reference Implementation

The reference implementation provided by the IBM Sterling Store Inventory Management provides transactional and configurational data to demonstrate how the features that are introduced in the Sterling Store Inventory Management . In addition to the reference implementation such as Participant Modeling, User Groups and Users, Queue Management, and Catalog Management.

Additionally, this reference implementation is used as a starting point for your implementation of the Sterling Store Inventory Management. Although, the data provided need not be used exactly as it is given, this reference implementation gives you an idea of how to configure the Sterling Store Inventory Management suitable to your business.

---

### Reference Implementation Setup

The reference implementation installation scripts set up configuration data in order to demonstrate the IBM Sterling Store Inventory Management features.

Using the Sterling Store Inventory Management Configurator, the Sterling Store Inventory Management is configured by modules that reflect different aspects of the product.

For more information about the reference implementation such as Participant Modeling, User Groups and Users, Queue Management, and Catalog Management, see the *Sterling Selling and Fulfillment Suite: Applications Reference Implementation Guide*.

---

### Store Layout Configuration

In the IBM Sterling Store Inventory Management, it is assumed that a store has only one zone, and the zone name is same as the name of the store.

#### Location Size Code

The location size codes are configured in terms of standard pallet size. The location size code is the same as the store name and has infinite capacity.

#### Locations

In the Sterling Store Inventory Management, a store can be modelled as:

- No Location — Here, a store can be configured with no locations. In this kind of store, you cannot perform any inventory actions.
- Single Location — This is mainly applicable in a store with small setup. The store has only one location.
- Store is modelled with three locations — Here, one location is designated as dock location, one as back location, and one as front location.
- Store tracks inventory in multiple locations — Here, one location is designated as dock location and one as front location. The remaining locations can be marked as back locations.

## Devices

Printers are used in a store to print labels or reports. The printers that the Sterling Store Inventory Management supports are:

Table 38. Printers

Printer Name	Print Documents
HP-LASERJET-4300	PICKTICKET, VICS_BOL, COUNTSHEET, RECEIVING WORKSHEET
Zebra- 140	Store_Acknowledgement

---

## Inventory Rules Configuration

The IBM Sterling Store Inventory Management defines certain inventory attributes to track inventory in a store.

### Product Class

The product classes refers to the product classification of an item. The typical product class are first quality, second quality, or finished good.

In the Sterling Store Inventory Management, the product class defined is Good. This indicates that the inventory is of good quality.

### Inventory Status

The inventory status refers to the availability of inventory in a store. The inventory status defined in the Sterling Store Inventory Management is:

Table 39. Inventory Status

Status	Description	Supply Type
N	Normal	ONHAND

### Inventory Adjustment Reasons

Inventory adjustments are associated with a reason code and reason text. These codes are used as tracking mechanisms for the exceptions that require an adjustment. The inventory adjustment reason enables tracking and reporting of adjustments at the reason code level.

Typically, inventory adjustment transactions are written for exceptions where inventory is found or not found.

The following table lists the default configurations for inventory adjustment reasons.

Table 40. Inventory Adjustment Reasons

Reason Code	Accounting bin (Y/N)	Description
SHIP		This reason code is used by the system when confirming shipments.

Table 40. Inventory Adjustment Reasons (continued)

Reason Code	Accounting bin (Y/N)	Description
RECEIPT		This reason code is used by the system when receiving inventory.
BLIND		This reason code is used by the system during blind receiving.
STORE-PHYSICAL-COUNT	Y	Indicates that a different bin location is associated with the physical count.
STORE-CYCLE-COUNT	Y	Indicates that a different bin location is associated with the cycle count.
Missing		Indicates that the inventory is missing.
STORE-MOVE		Indicates that the inventory is moved for some other reasons.
Re-Classify		Indicates that the adjustment is due to reclassification of items.
Found-New		Indicates that the adjustment is due to new items found in a location.
Scrap		Indicates that the item has been scrapped.
SOP_ADJUSTMENT		Indicates the host adjustment reason code used for performing adjustment operations such as Re-Classify, Found-New, and Scrap.

## Inventory Transitions

Inventory status transitions may be specified at the zone level to allow you to automatically change the status of inventory in a store.

The default inventory status transition in the Sterling Store Inventory Management is shown the table below:

Table 41. Inventory Status Transitions

Zones	Transition
For all zones	Blank to Normal

## Count Process

Count process ensures accurate stock level information in a store. The user can perform either cycle count and/or physical count.

- **Count Program**

A default STORE\_CYCLE\_COUNT count program and STORE\_CALENDAR calendar is provided to ensure that every item in a store is counted at least once in a quarter.

- **Count Request Cancellation Reasons**

The cancellation reason code for cycle count is: STORE\_INCOMPLETE\_COUNT.

The cancellation reason code for physical count is: STORE\_INVALID\_COUNT.

- **Count Task Type**

Different task types are created for cycle count and physical count. Counts can be performed either through paper-based or mobile devices.

*Table 42. Count Task Types*

Task Type	Description
COUNT	Cycle count performed through mobile devices.
PCCOUNT	Physical count performed through mobile devices.

- **Paper-Based Counting**

In a paper-based counting, count worksheets are printed that provides a list of items to count. Based on the count program, a count request is generated. The count request creates count worksheets, which are then printed.

The purge criteria for cycle count is 30 days.

## Inbound Rules Configuration

The inbound rules configured in the IBM Sterling Store Inventory Management are:

### Receiving Process

The various methods of receiving processes that a store can follow are:

- Receive inventory against a purchase order.
- Stores receive ASN that are sent through E-mail or Fax. The same process is followed for receiving transfer orders.
- Blind Receiving is receiving inventory when no BOL or order number exists for the inventory in the system.

Once the inventory is received, a receiving worksheet is printed.

### Node Receiving Preferences

You can receive a case or pallet, or both. The configuration set as part of receiving preferences are:

*Table 43. Node Receiving Preferences*

Case/Pallet	Build Preferences
Case	Optional
Pallet	Optional

Check the Shipment Entry Allowed box for blind receiving.

The disposition code for receipt with QC or without QC is set to QC-PASSED.

## Receiving Disposition Setup

A disposition code identifies the product class and inventory status to be assigned to the inventory being received.

The standard disposition configuration are:

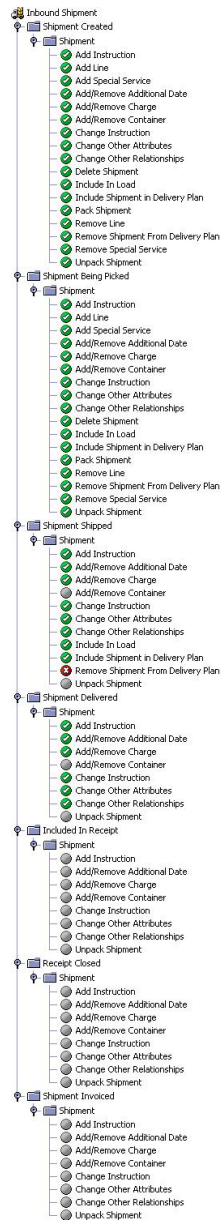
*Table 44. Disposition Configuration*

Disposition Code	Product Class	Inventory Status	Final Disposition
QC-PASSED	Good	Blank	Y

## Shipment Modification Rules

The Shipment Modification Rules apply to the following document types:

- Purchase Order
- Transfer Order



## Receipt Modification Rules

The Receipt Modification Rules apply to the following document types:

- Purchase Order
- Transfer Order





## Sourcing Rules

Sourcing rules are necessary to create chained orders for a sales order. The Matrix-R Enterprise has sourcing rules configured for products being shipped and for procured.

The fulfillment type is defined at the Enterprise level. The Matrix-R Enterprise has sourcing rules configured.

In the reference implementation, a procurement rule is configured for all items and node types as a Store. Based on the requirement, an Enterprise can change its sourcing and procurement rules.

## Print Services

For the Inbound process the following print services are provided:

- Customer Order Label
- Receiving Worksheet
- Receiving Discrepancy Report

---

## Outbound Rules Configuration

The Outbound process includes managing and controlling the outgoing materials from the point when shipments are created, to the point when shipments are picked up by a customer or loaded into trailers. Once the shipment is created, the shipment is ready for backroom pick or customer pick.



---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing*

*IBM Corporation*

*North Castle Drive*

*Armonk, NY 10504-1785*

*U.S.A.*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing*

*Legal and Intellectual Property Law*

*IBM Japan Ltd.*

*1623-14, Shimotsuruma, Yamato-shi*

*Kanagawa 242-8502 Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation*

*J46A/G4*

*555 Bailey Avenue*

*San Jose, CA 95141-1003*

*U.S.A.*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2012. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2012.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

#### **Trademarks**

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)<sup>®</sup> are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center<sup>®</sup>, Connect:Direct<sup>®</sup>, Connect:Enterprise<sup>®</sup>, Gentran<sup>®</sup>, Gentran<sup>®</sup>:Basic<sup>®</sup>, Gentran:Control<sup>®</sup>, Gentran:Director<sup>®</sup>, Gentran:Plus<sup>®</sup>, Gentran:Realtime<sup>®</sup>, Gentran:Server<sup>®</sup>, Gentran:Viewpoint<sup>®</sup>, Sterling Commerce<sup>™</sup>, Sterling Information Broker<sup>®</sup>, and Sterling Integrator<sup>®</sup> are trademarks or registered trademarks of Sterling Commerce<sup>™</sup>, Inc., an IBM Company.

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

---

# Index

## A

adjustment  
    reason codes 7

## B

blind receipts 29

## C

Configure Inventory Adjustment  
    Rules 8

## I

inventory  
    adjustment reason 7, 54  
inventory adjustment  
    end user impact 8  
    reference implementation 8  
    solution 7  
Inventory Adjustment End-User  
    Impact 8  
inventory adjustment reasons 7  
Inventory Adjustment Reasons 7  
Inventory Adjustment Reference  
    Implementation 8  
Inventory Adjustment Solution 7  
Inventory Adjustments 7  
inventory transitions 9

## P

process  
    receiving 29

## R

receipts  
    with Order or Shipment 29  
Receiving  
    Blind 29  
    End User Impact 32  
    Implementation 32  
    Print 29  
    Record 29  
    Solutions 30  
reference implementation  
    receiving 33  
retrieving item details 11

## S

searching for items 11  
searching inventory  
    solution 11









Printed in USA