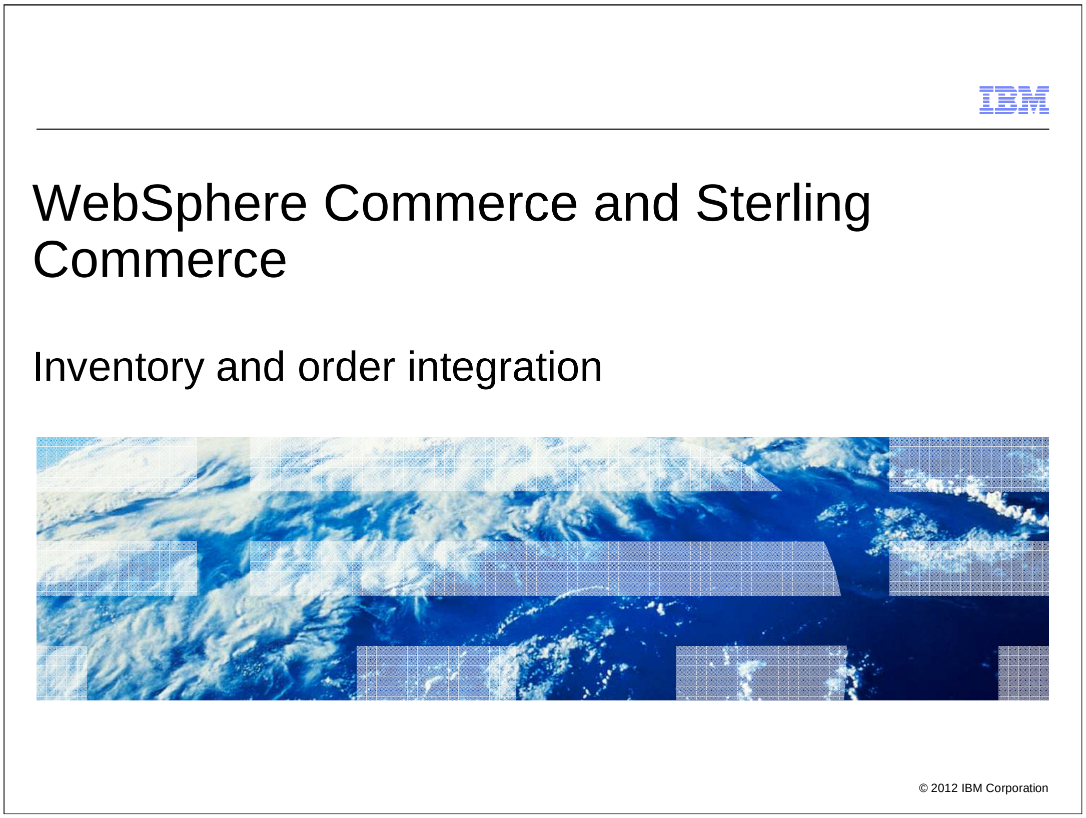
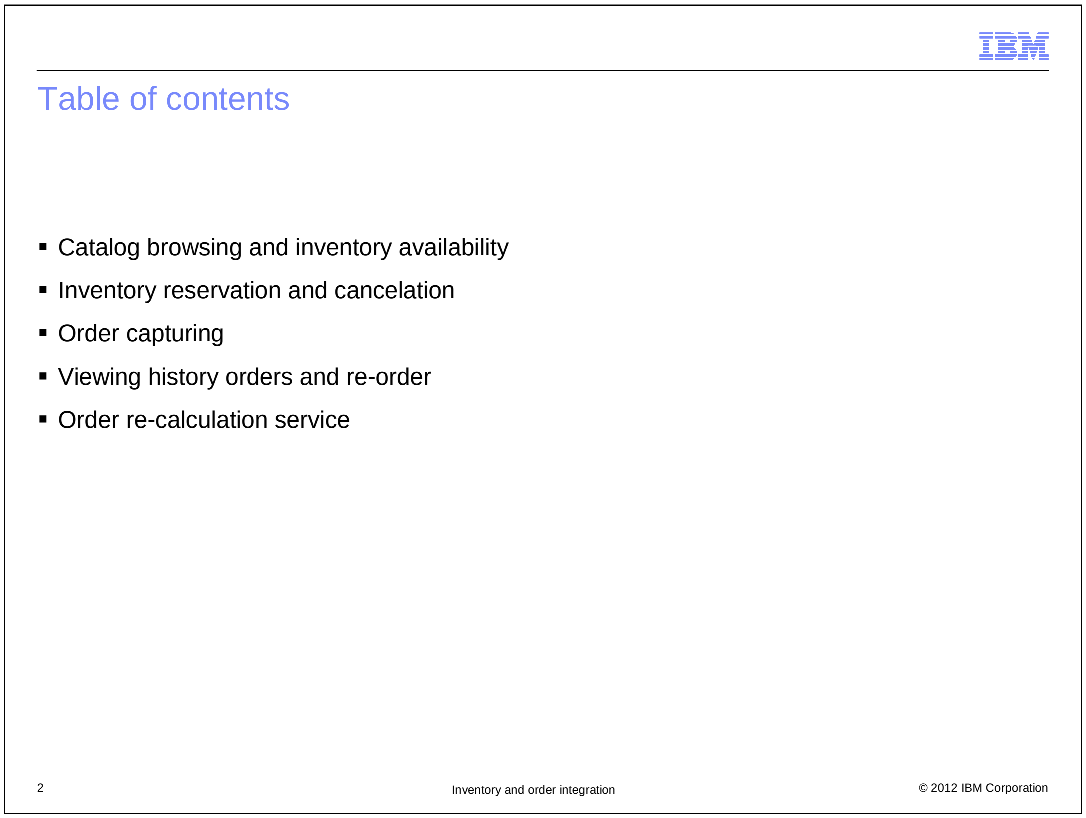


# WebSphere Commerce and Sterling Commerce

## Inventory and order integration

This presentation introduces WebSphere® Commerce and Sterling Commerce® inventory and order integration.



## Table of contents

- Catalog browsing and inventory availability
- Inventory reservation and cancelation
- Order capturing
- Viewing history orders and re-order
- Order re-calculation service

This presentation discusses how the Inventory and order integration are implemented. The integration flows covered in this presentation include viewing inventory availability flow, inventory reservation and cancelation flow, order capturing flow, viewing history orders and re-order flow. Last the problem determination topics are covered.

Section

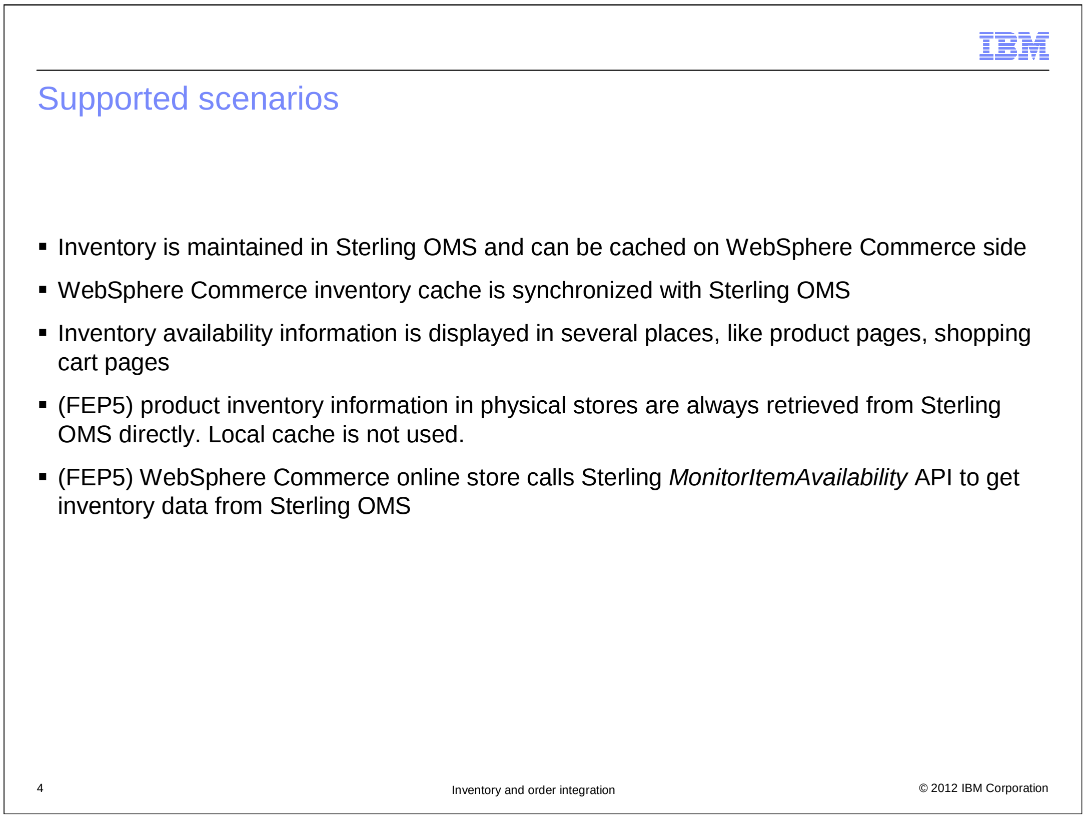
## *Catalog browsing and inventory availability*

3

Inventory and order integration

© 2012 IBM Corporation

First the presentation will take you through the catalog browsing and inventory availability flow.



## Supported scenarios

- Inventory is maintained in Sterling OMS and can be cached on WebSphere Commerce side
- WebSphere Commerce inventory cache is synchronized with Sterling OMS
- Inventory availability information is displayed in several places, like product pages, shopping cart pages
- (FEP5) product inventory information in physical stores are always retrieved from Sterling OMS directly. Local cache is not used.
- (FEP5) WebSphere Commerce online store calls Sterling *MonitorItemAvailability* API to get inventory data from Sterling OMS

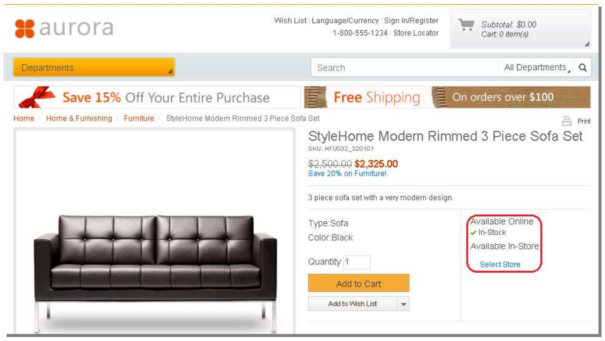
This slide lists the supported scenarios for Catalog browsing and inventory availability. WebSphere Commerce and Sterling Commerce integration uses Sterling as the inventory management system. Whenever WebSphere Commerce needs to know the inventory information, it either gets inventory from the local cache or from Sterling Commerce by making an outbound service call. To get better performance, Commerce provides inventory caching capability.

In integrated environment RTAM is used to populate WebSphere Commerce cache and in special scenarios real time calls are made from WebSphere Commerce.

All these scenarios mentioned are already supported before feature pack 5. In this feature pack 5 release, some changes have been made regarding getting the inventory availability data from Sterling. One change is that for the physical stores, WebSphere Commerce gets the inventory information directly from Sterling other than using the local cache. The second change is that for online stores, when a particular item's inventory is expired or not in WebSphere Commerce cache, WebSphere Commerce *MonitorItemAvailability* API is called to sync the inventory cache with Sterling, instead of the findInventory API used in Feature Pack 3.

## Inventory cache

- Two caching options – Database cache and In memory cache
- Tables for cache configuration
  - INVCNF - inventory configuration
  - INVCNFREL - inventory configuration relationship
- Table for database cache
  - INVAVL - Inventory availability
  - Contains inventory record for every item in the catalog

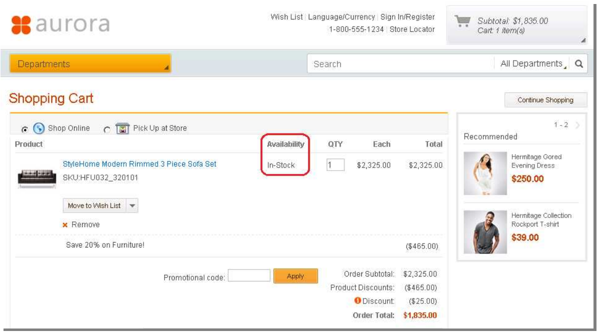


StyleHome Modern Rimmed 3 Piece Sofa Set
SKU: HFU032_320101
~~$2,500.00~~ $2,325.00
Save 20% on Furniture!

3 piece sofa set with a very modern design.

Type:Sofa
Color:Black
Quantity 1

Available Online
✓ In-Stock
Available In-Store



Shopping Cart

● Shop Online ○ Pick Up at Store

| Product | Availability | QTY | Each | Total |
|---|---|---|---|---|
| StyleHome Modern Rimmed 3 Piece Sofa Set<br>SKU:HFU032_320101 | In-Stock | 1 | $2,325.00 | $2,325.00 |
| Save 20% on Furniture! | | | | ($465.00) |

Promotional code:

| | |
|---|---|
| Order Subtotal: | $2,325.00 |
| Product Discounts: | ($465.00) |
| Discount: | ($25.00) |
| Order Total: | $1,835.00 |

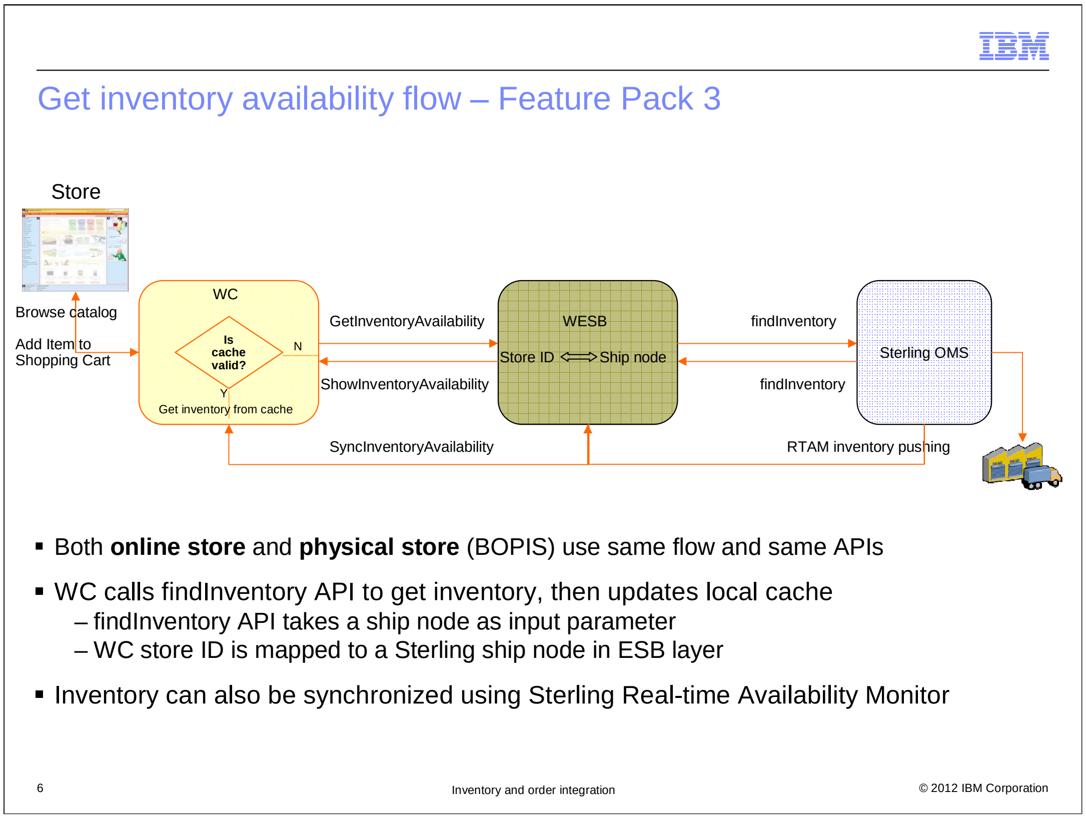
Hermitage Gored Evening Dress $250.00
Hermitage Collection Rockport T-shirt $39.00

Inventory cache, implemented in a previous release, is briefly introduced here so that you can better understand the Feature Pack 5 solution.

The inventory availability information can be viewed in many places on the storefront, such as product detail page and shopping cart page. If the inventory cache feature is enabled for a store, when a WebSphere Commerce store page displays the inventory availability information, WebSphere Commerce reads the inventory data from the cache first. If the cache data is expired, WebSphere Commerce makes a service call to Sterling to get inventory data. Then WebSphere Commerce updates the data in local cache and returns the inventory data to the store page.

Two inventory cache options have been implemented: database cache and in memory cache. To let WebSphere Commerce know which cache option is used, and how your local cache is managed, you will need manually configure inventory configuration table and inventory configuration relationship table.

If database cache is chosen, the inventory availability data is saved into inventory availability table. In the runtime, WebSphere Commerce gets all catalog entry’s inventory data from Sterling and saves them into the table.



## Get inventory availability flow – Feature Pack 3

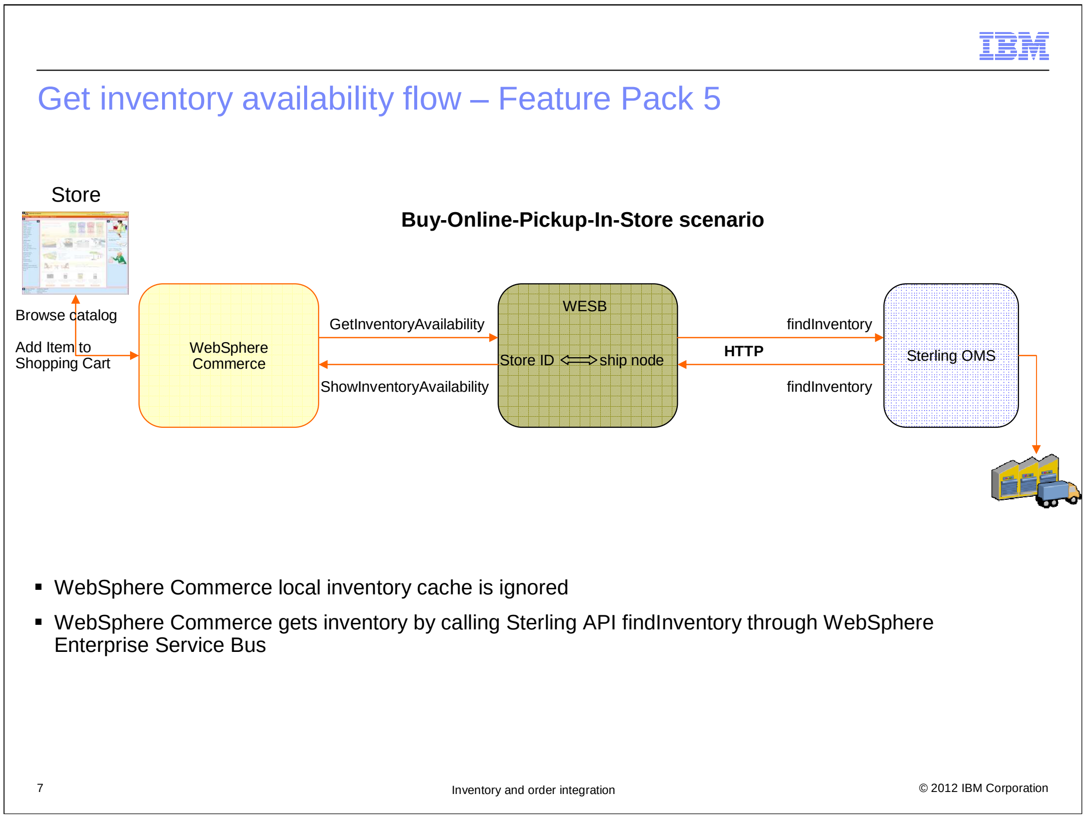
- Both **online store** and **physical store** (BOPIS) use same flow and same APIs
- WC calls findInventory API to get inventory, then updates local cache
  - findInventory API takes a ship node as input parameter
  - WC store ID is mapped to a Sterling ship node in ESB layer
- Inventory can also be synchronized using Sterling Real-time Availability Monitor

6 Inventory and order integration

The diagram on this slide shows the inventory synchronization flow in Feature Pack 3.

The flow for getting inventory availability flow is no different between physical and online stores. Whenever a store needs to get inventory information from Sterling, WebSphere Commerce calls the GetInventoryAvailability outbound service, which in turn calls Sterling's findInventory API to get the inventory from Sterling.

There is a limitation in Feature Pack 3. Whenever WebSphere Commerce calls Sterling findInventory APIs, whether for on-line or physical stores, the store ID is always passed into the message. The store ID is mapped to a ship node in the WebSphere Enterprise Service Bus layer. WebSphere Commerce can only find the inventory availability in the mapped ship node. This means in Feature Pack 3, only one fulfillment center per store is supported. This makes sense for physical stores because in real cases typically one physical store is mapped to one ship node. But for on-line stores, this limitation does not make sense because for an on-line store might have multiple fulfillment centers.



## Get inventory availability flow – Feature Pack 5

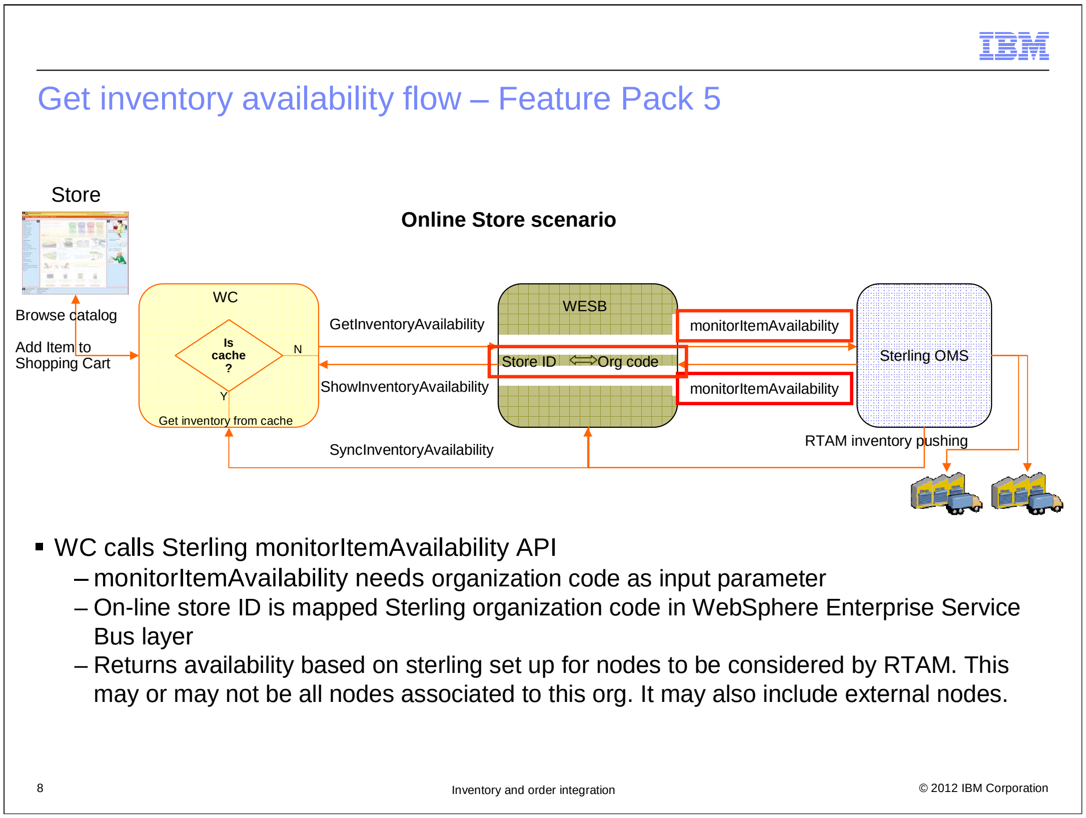
**Buy-Online-Pickup-In-Store scenario**

- WebSphere Commerce local inventory cache is ignored
- WebSphere Commerce gets inventory by calling Sterling API findInventory through WebSphere Enterprise Service Bus

7 Inventory and order integration © 2012 IBM Corporation

In Feature Pack 5, the physical and online stores are handled separately. For physical stores’ Buy Online Pick Up In Store scenario, Feature Pack 5 still calls Sterling’s findInventory API to get the inventory availability data. Physical store ID is still mapped to one ship node, however, using the local cache for physical store is not recommended. We do this because inventory cache on WebSphere Commerce side is "overall" inventory, it is not at various store levels. We do this because for store pick up a realtime call is desirable as value in cache is for shipping not pick up at a particular store.

You should be aware this is just a configuration change in the INVCNF table. You can always switch back to use the local cache if you don’t satisfy the direct call’s performance.



## Get inventory availability flow – Feature Pack 5

- WC calls Sterling monitorItemAvailability API
  - monitorItemAvailability needs organization code as input parameter
  - On-line store ID is mapped Sterling organization code in WebSphere Enterprise Service Bus layer
  - Returns availability based on sterling set up for nodes to be considered by RTAM. This may or may not be all nodes associated to this org. It may also include external nodes.

Some changes have been made for online stores in Feature Pack 5.

In Feature Pack 3, online stores call Sterling findInventory API to get the inventory availability data. In Feature Pack 5 WebSphere Commerce online stores call Sterling's monitorItemAvailability API to get the inventory data. This API takes Organization code as input. The online store ID and organization code is mapped in WebSphere Enterprise Service Bus mediation module.

In Sterling OMS, one organization can have multiple ship nodes. In Feature Pack 5, WebSphere Commerce online store's inventory information is based on sterling set up for nodes to be considered by RTAM. This may or may not be all nodes associated to this organization. It may also include external nodes.

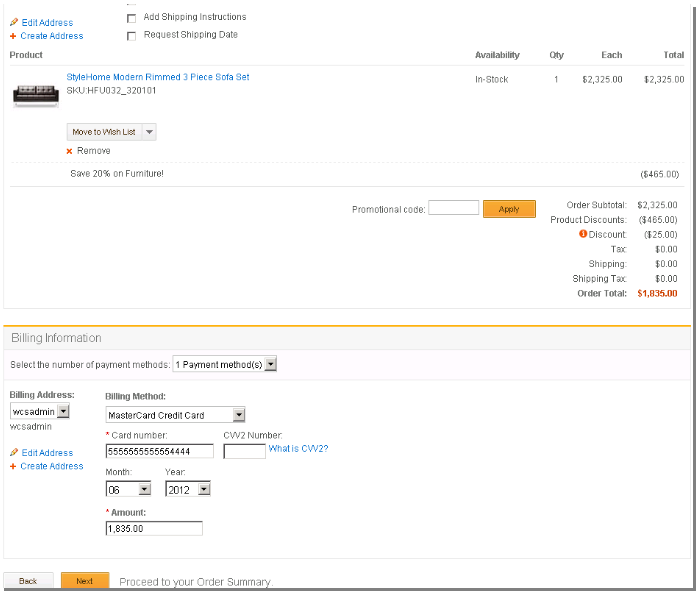
Section

## *Inventory reservation and cancelation*

This section introduces inventory reservation and cancelation flow.

## Supported scenarios

- Inventory is reserved in Sterling OMS during check-out process
- Inventory reservation is canceled when shopping cart is changed (add, update or delete items)



Edit Address
+ Create Address

□ Add Shipping Instructions
□ Request Shipping Date

| Product | Availability | Qty | Each | Total |
|---|---|---|---|---|
| StyleHome Modern Rimmed 3 Piece Sofa Set<br>SKU:HFU032_320101 | In-Stock | 1 | $2,325.00 | $2,325.00 |
| Save 20% on Furniture! | | | | ($465.00) |

Promotional code: Apply

| | |
|---|---|
| Order Subtotal: | $2,325.00 |
| Product Discounts: | ($465.00) |
| Discount: | ($25.00) |
| Tax: | $0.00 |
| Shipping: | $0.00 |
| Shipping Tax: | $0.00 |
| **Order Total:** | **$1,835.00** |

Billing Information

Select the number of payment methods: 1 Payment method(s)

**Billing Address:** wcsadmin
wcsadmin

Edit Address
+ Create Address

**Billing Method:** MasterCard Credit Card

* Card number: 5555555555554444

CVV2 Number: What is CVV2?

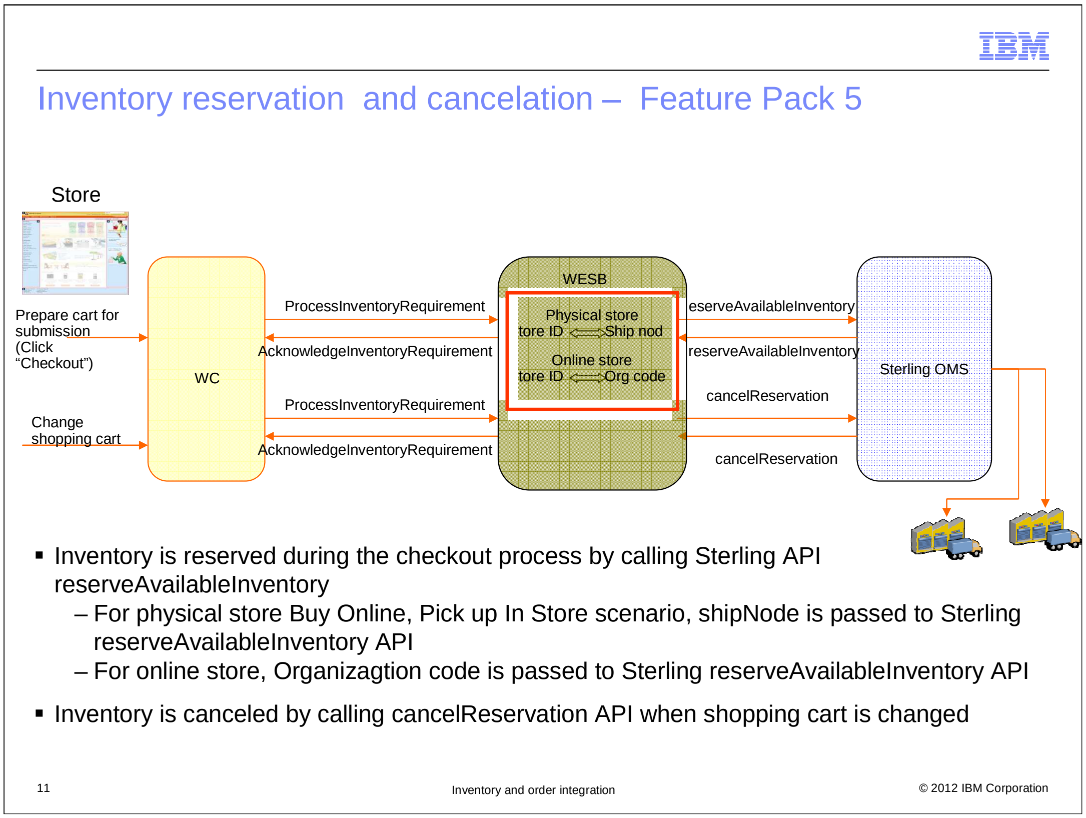
Month: 06

Year: 2012

* **Amount:** 1,835.00

Reserving inventory from Sterling happens when a shopper checks out an order. The screen capture shown on the slide is one of the checkout pages. The reservation happens after a shopper clicks the Next button in the page, WebSphere Commerce will send request to Sterling to reserve the inventory for all items in the shopping cart.

Canceling inventory happens when a shopper changes his shopping cart, for example, adding a new item to shopping cart, updating an item quantity, or deleting an item from the cart. When any of these actions happen, WebSphere Commerce will check if any item in this shopping cart has been reserved. If yes, WebSphere Commerce will ask Sterling to cancel the reservation for this order.



## Inventory reservation and cancelation – Feature Pack 5

Store
Prepare cart for submission (Click “Checkout”)
Change shopping cart
WC
ProcessInventoryRequirement
AcknowledgeInventoryRequirement
ProcessInventoryRequirement
AcknowledgeInventoryRequirement
WESB
Physical store
tore ID ⟺ Ship nod
Online store
tore ID ⟺ Org code
eserveAvailableInventory
reserveAvailableInventory
cancelReservation
cancelReservation
Sterling OMS

- Inventory is reserved during the checkout process by calling Sterling API reserveAvailableInventory
  - For physical store Buy Online, Pick up In Store scenario, shipNode is passed to Sterling reserveAvailableInventory API
  - For online store, Organizagtion code is passed to Sterling reserveAvailableInventory API
- Inventory is canceled by calling cancelReservation API when shopping cart is changed

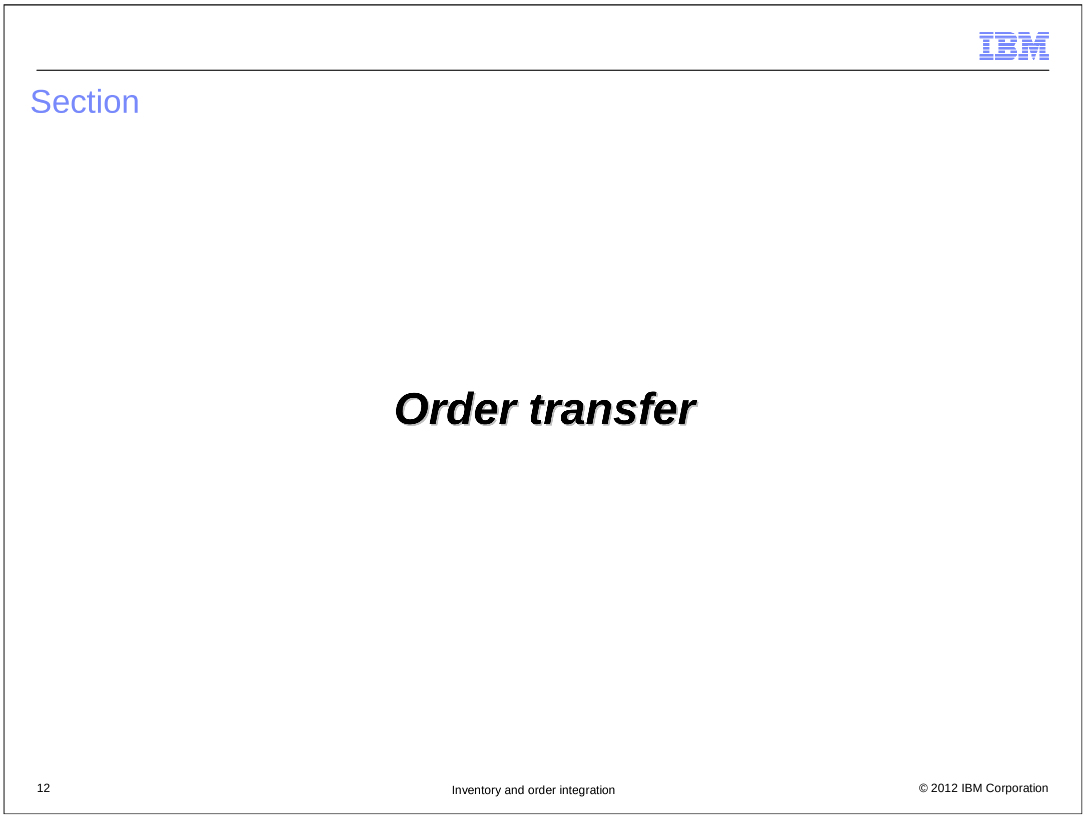
11 Inventory and order integration © 2012 IBM Corporation

The diagram on the slide shows inventory reservation and inventory cancelation flows, and the APIs used in the flows.

For physical stores, WebSphere Commerce store ID is mapped to a Sterling ship node ID. For the online store, the store ID is mapped to Sterling’s organization code, which can have multiple ship nodes.

There is a limitation for the online store. A WebSphere Commerce online store is mapped to an enterprise that may fulfill from multiple ship nodes, but for the order submitted from WebSphere Commerce storefront, an item can only be reserved from one ship node.

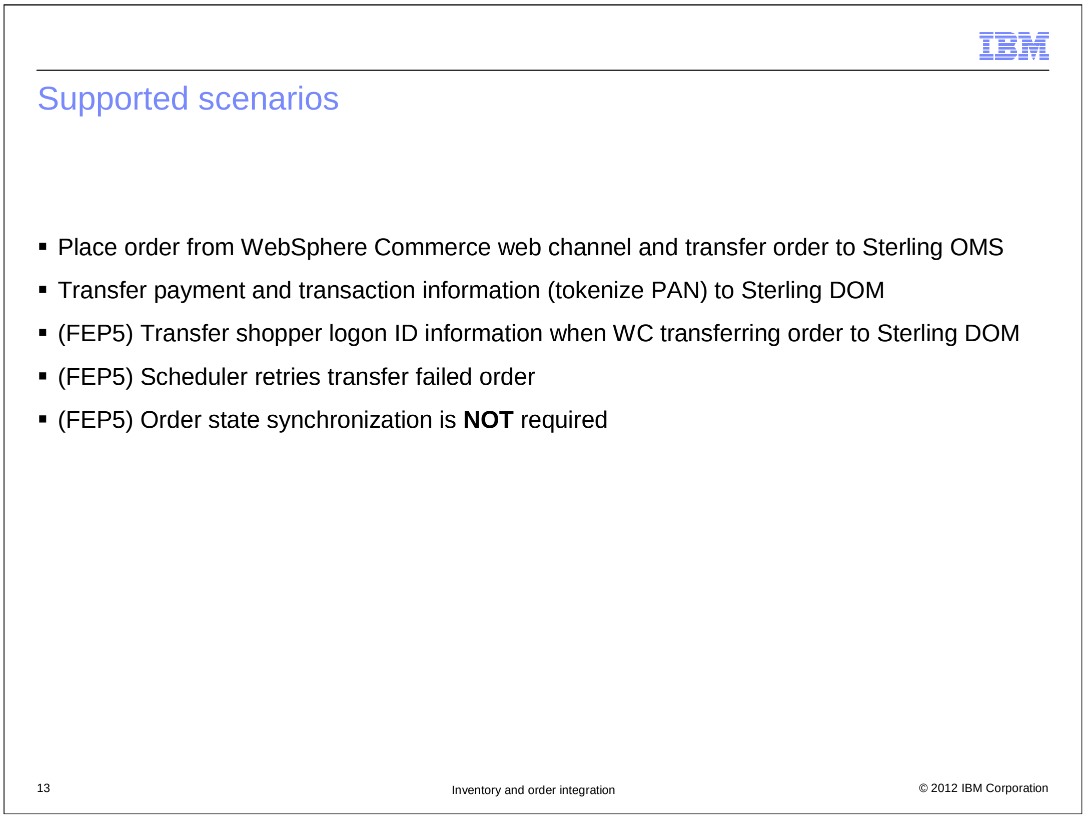
When you change the shopping cart, for example, adding new items into the cart, update an item’s quantity, or delete an item,  the previous reservation is canceled. If a shopper abandons checkout after reservation occurs, WebSphere Commerce will not cancel the reservation. The reservation will expire depending on the expiration rule that you defined in the Sterling OMS.



Section

## *Order transfer*

This section discusses order transfer flow.

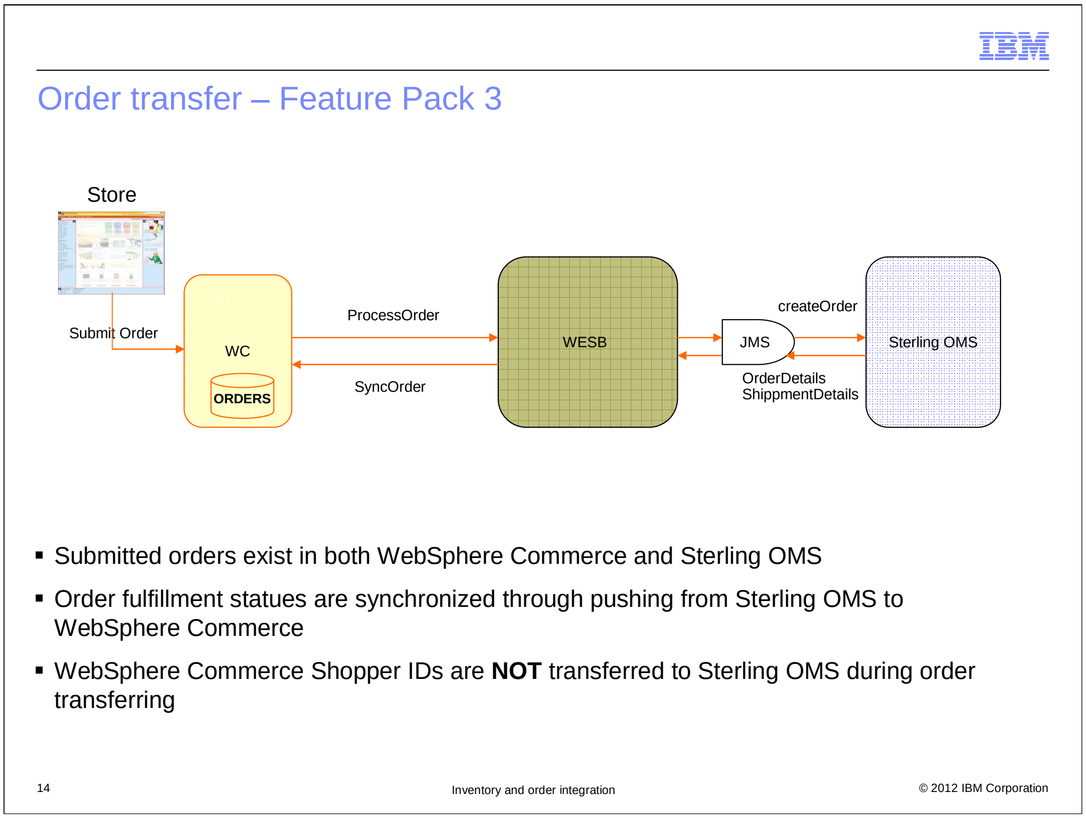


## Supported scenarios

- Place order from WebSphere Commerce web channel and transfer order to Sterling OMS
- Transfer payment and transaction information (tokenize PAN) to Sterling DOM
- (FEP5) Transfer shopper logon ID information when WC transferring order to Sterling DOM
- (FEP5) Scheduler retries transfer failed order
- (FEP5) Order state synchronization is **NOT** required

Order transferring flow was implemented before Feature Pack 5. After an order is submitted by a shopper, WebSphere Commerce saves the order into its own database. In the meantime, WebSphere Commerce transfers the order to Sterling OMS. Along with the order transferring, payment information is sent to the Sterling too.

In Feature Pack 5, along with the order transferring, the shopper ID is transferred to the Sterling as well. After an order is created in Sterling OMS, the order's status is not synchronized between Commerce and Sterling. This is different from what is done in Feature Pack 3.

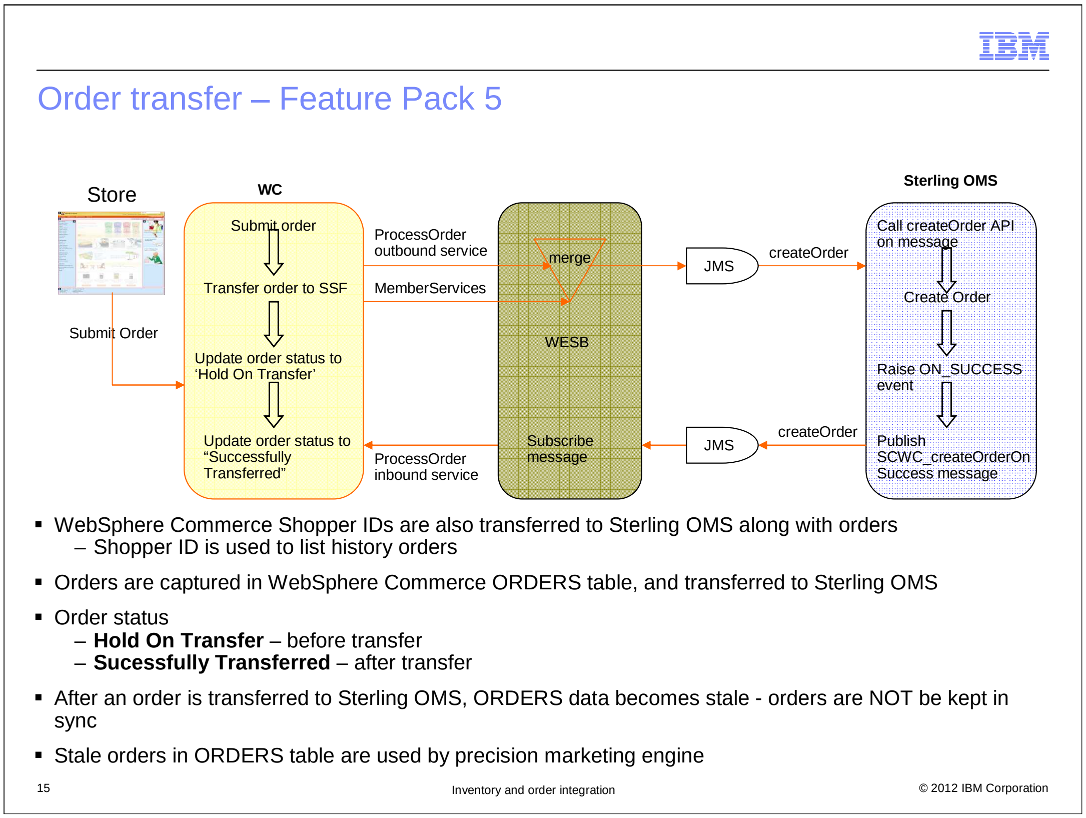


## Order transfer – Feature Pack 3

- Submitted orders exist in both WebSphere Commerce and Sterling OMS
- Order fulfillment statues are synchronized through pushing from Sterling OMS to WebSphere Commerce
- WebSphere Commerce Shopper IDs are **NOT** transferred to Sterling OMS during order transferring

14 Inventory and order integration

Order transfer flow in Feature Pack 3 is shown here. WebSphere Commerce calls its outbound service ProcessOrder to pass the order to the WebSphere Enterprise Service Bus mediation module, then Sterling createOrder API is called to create a new order in the Sterling OMS. Whenever the transferred order status is changed, Sterling calls WebSphere inbound service SyncOrder to update the status in WebSphere Commerce's local database. The order status in WebSphere Commerce database and Sterling database are the same. When WebSphere Commerce wants to get order's information, it gets it from its local database.



IBM

## Order transfer – Feature Pack 5

Store

Submit Order

WC

Submit order

Transfer order to SSF

Update order status to 'Hold On Transfer'

Update order status to "Successfully Transferred"

ProcessOrder outbound service

MemberServices

ProcessOrder inbound service

merge

WESB

Subscribe message

JMS

createOrder

JMS

createOrder

Sterling OMS

Call createOrder API on message

Create Order

Raise ON_SUCCESS event

Publish SCWC_createOrderOnSuccess message

- WebSphere Commerce Shopper IDs are also transferred to Sterling OMS along with orders
  - Shopper ID is used to list history orders
- Orders are captured in WebSphere Commerce ORDERS table, and transferred to Sterling OMS
- Order status
  - **Hold On Transfer** – before transfer
  - **Sucessfully Transferred** – after transfer
- After an order is transferred to Sterling OMS, ORDERS data becomes stale - orders are NOT be kept in sync
- Stale orders in ORDERS table are used by precision marketing engine

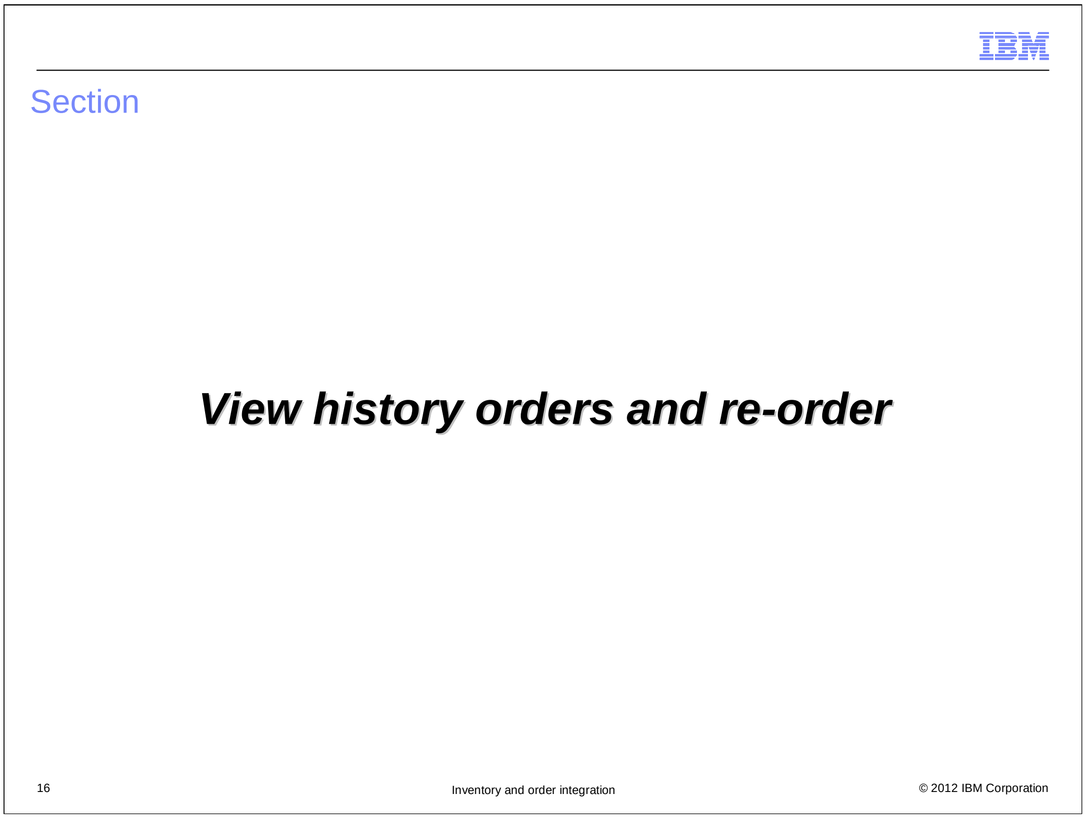
15 Inventory and order integration © 2012 IBM Corporation

Feature Pack 5 still uses the WebSphere Commerce ProcessOrder service and Sterling's createOrder API to transfer the order, but several changes have been made in Feature Pack 5.

In Feature Pack 5, during order transfer to Sterling, WebSphere Commerce also transfers shopper logon ID and shopper detail information to Sterling. The existing order logical schema does not support transferring shopper's detailed information. So from the WebSphere Enterprise Service Bus mediation module, WebSphere Commerce member service is called to get detailed member information. This information is merged with the existing order message, then transferred to the input of Sterling createOrder API. The transferred shopper ID is mapped to Sterling's buyerUserID. This attribute is used to identify owner of order. WebSphere Commerce will use shopper ID to retrieve all shopper orders from Sterling when the shopper views the store's Order History page.

Since the orders in the WebSphere Commerce database and in Sterling OMS are not synchronized, when WebSphere Commerce needs the order information, it gets them directly from Sterling. This does not mean you can safely delete the orders from the WebSphere Commerce database after the orders are transferred to Sterling. The orders in the WebSphere Commerce database are still used by the WebSphere Commerce marketing and promotion engine.

Before an order is transferred to Sterling, WebSphere Commerce still needs to maintain the order's status, because its status affects where WebSphere Commerce should get the order data for order history display. After a shopper submits an order, the order is saved into the WebSphere Commerce ORDERS table, and the order status is set to Hold on Transfer. This status means the order is created, but has not been transferred to Sterling. After the order is transferred to Sterling, Sterling publishes a message to notify WebSphere Commerce that the order has been received and created successfully. After WebSphere Commerce gets the message, WebSphere Commerce updates the local order status to Successfully Transferred.



Section

## *View history orders and re-order*

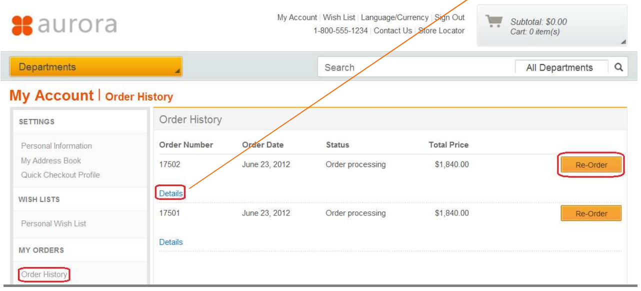
16

Inventory and order integration

© 2012 IBM Corporation

This section introduces view history orders and re-order flows.

# View history orders and details



aurora
My Account | Wish List | Language/Currency | Sign Out
1-800-555-1234 | Contact Us | Store Locator
Subtotal: $0.00
Cart: 0 item(s)
Departments
Search
All Departments

My Account | Order History

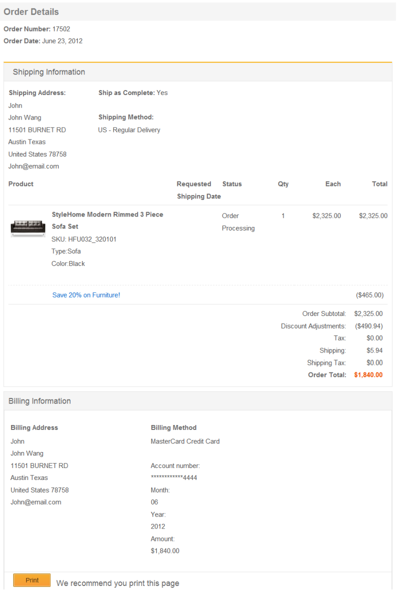
SETTINGS
Personal Information
My Address Book
Quick Checkout Profile

WISH LISTS
Personal Wish List

MY ORDERS
Order History

Order History

| Order Number | Order Date | Status | Total Price | |
|---|---|---|---|---|
| 17502<br>Details | June 23, 2012 | Order processing | $1,840.00 | Re-Order |
| 17501<br>Details | June 23, 2012 | Order processing | $1,840.00 | Re-Order |



Order Details

**Order Number:** 17502
**Order Date:** June 23, 2012

Shipping Information

**Shipping Address:**
John
John Wang
11501 BURNET RD
Austin Texas
United States 78758
John@email.com

**Ship as Complete:** Yes

**Shipping Method:**
US - Regular Delivery

| Product | Requested Shipping Date | Status | Qty | Each | Total |
|---|---|---|---|---|---|
| **StyleHome Modern Rimmed 3 Piece Sofa Set**<br>SKU: HFU032_320101<br>Type:Sofa<br>Color:Black | | Order Processing | 1 | $2,325.00 | $2,325.00 |
| Save 20% on Furniture! | | | | | ($465.00) |

| | |
|---|---|
| Order Subtotal: | $2,325.00 |
| Discount Adjustments: | ($490.94) |
| Tax: | $0.00 |
| Shipping: | $5.94 |
| Shipping Tax: | $0.00 |
| **Order Total:** | **$1,840.00** |

Billing Information

**Billing Address**
John
John Wang
11501 BURNET RD
Austin Texas
United States 78758
John@email.com

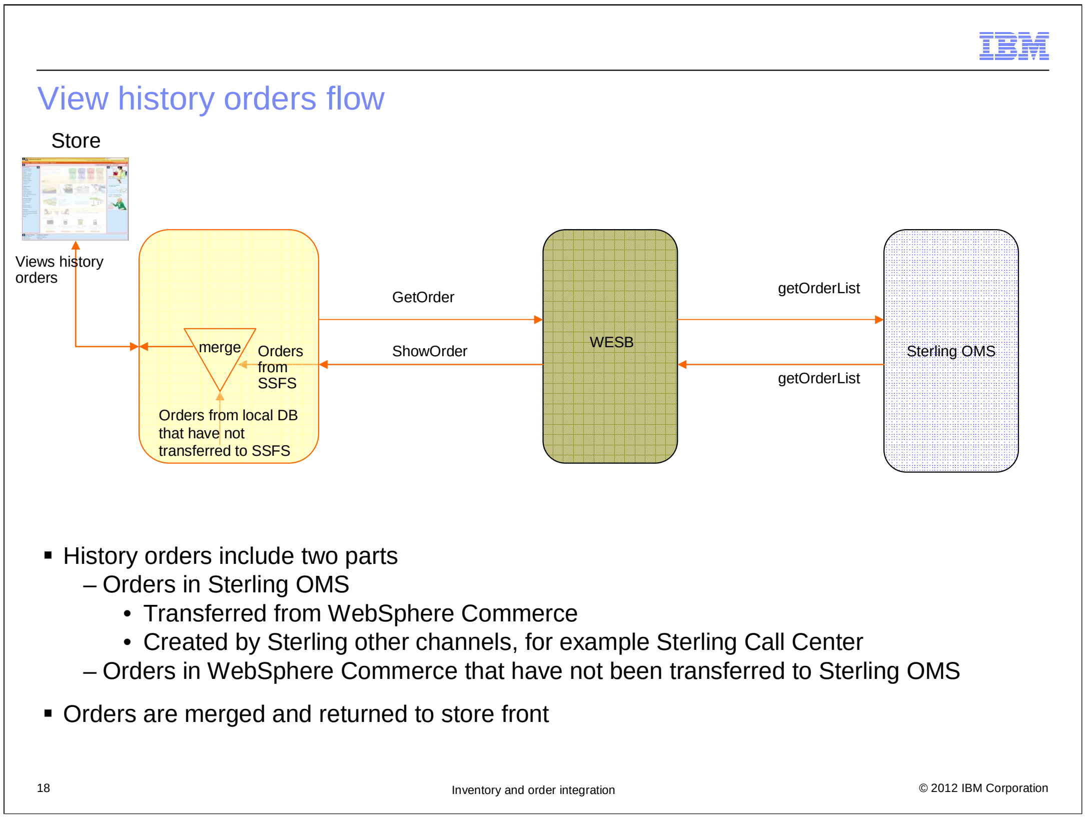
**Billing Method**
MasterCard Credit Card

Account number:
************4444
Month:
06
Year:
2012
Amount:
$1,840.00

We recommend you print this page

- Before Feature Pack 5, order history and details are get from WebSphere Commerce local database
- Feature Pack 5 WC will interact with both Sterling OMS and WebSphere Commerce
  - View history orders
  - View order details
  - Re-order

When a shopper goes to the **My Account** page and the **Order History** page, the shopper can see his history orders. On that page, the shopper can either view an order's details or do the re-order. Next you will see how WebSphere Commerce displays the history orders, order details, and does the re-order.



## View history orders flow

Store

Views history orders

merge

Orders from SSFS

Orders from local DB that have not transferred to SSFS

GetOrder

ShowOrder

WESB

getOrderList

getOrderList

Sterling OMS

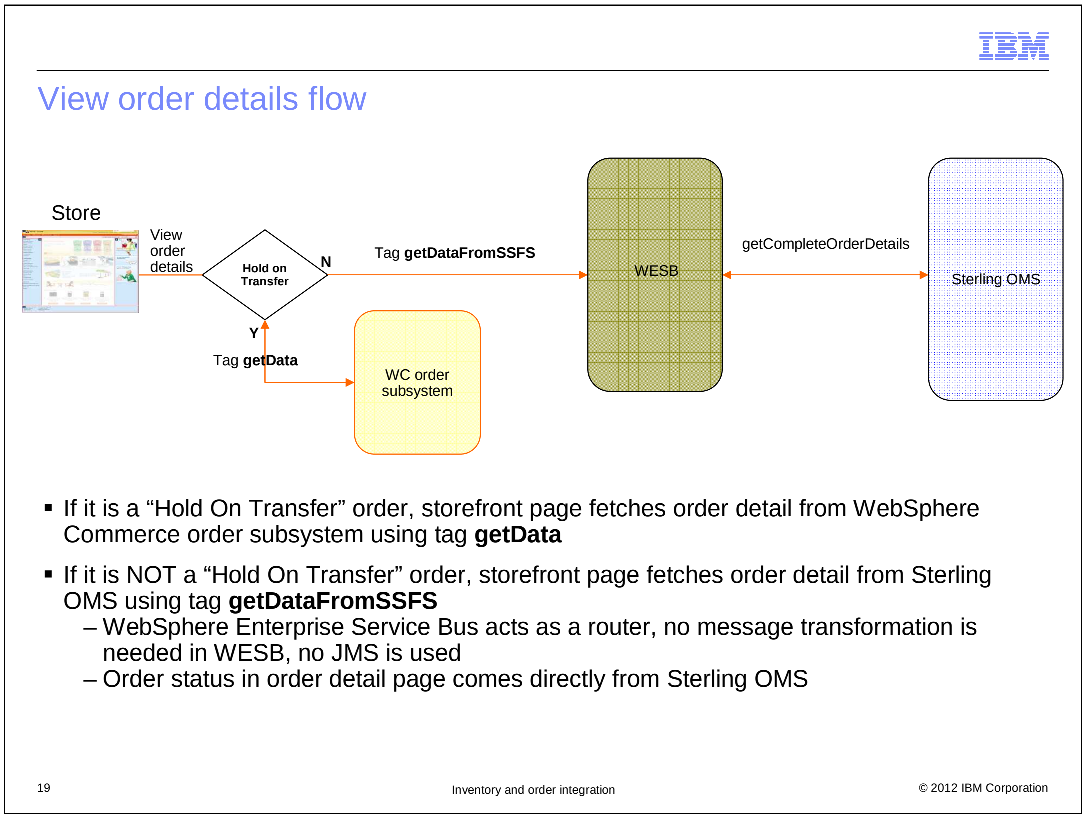
- History orders include two parts
  - Orders in Sterling OMS
    - Transferred from WebSphere Commerce
    - Created by Sterling other channels, for example Sterling Call Center
  - Orders in WebSphere Commerce that have not been transferred to Sterling OMS
- Orders are merged and returned to store front

18 Inventory and order integration

A shopper’s order history may include the orders submitted from WebSphere Commerce store, and orders placed from Sterling’s Call Center.

After a shopper submits an order from WebSphere Commerce store, the order is supposed to be transferred to Sterling. But sometimes, when the shopper views the order history from WebSphere Commerce store, that order may not be transferred to the Sterling.

To display the order history**,** WebSphere Commerce first gets the orders from its local database for the orders that were submitted from the WebSphere Commerce store, but have not been transferred in Sterling OMS. WebSphere Commerce then calls its getOrder outbound service and Sterling getOrderList API to get the orders that were already transferred to Sterling, and the orders submitted from Sterling Call center. WebSphere Commerce merges these orders on the store front and displays to the shopper.



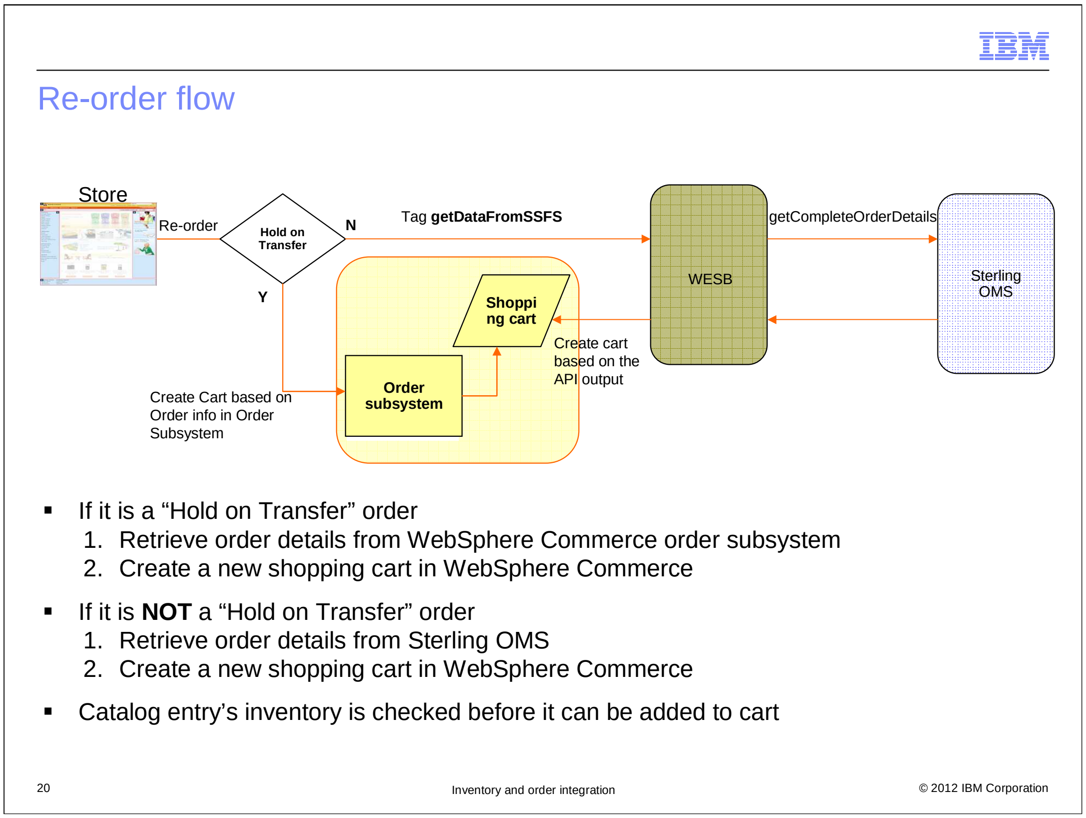
## View order details flow

- If it is a “Hold On Transfer” order, storefront page fetches order detail from WebSphere Commerce order subsystem using tag **getData**
- If it is NOT a “Hold On Transfer” order, storefront page fetches order detail from Sterling OMS using tag **getDataFromSSFS**
  - WebSphere Enterprise Service Bus acts as a router, no message transformation is needed in WESB, no JMS is used
  - Order status in order detail page comes directly from Sterling OMS

19 Inventory and order integration © 2012 IBM Corporation

Same as view History Orders flow, when a shopper views an order details, WebSphere Commerce will get order’s details from WebSphere Commerce local database if a order is “**Hold on Transfer”** order. WebSphere Commerce store uses getData tag to get the order data from WebSphere Commerce order subsystem.

If an order is NOT a “Hold On Transfer” order, the WebSphere Commerce store gets the order details from Sterling OMS. WebSphere Commerce store uses a new tag **getDataFromSSFS** to get the order details from Sterling by calling the Sterling **getCompleteOrderDetails** API. From the diagram you can see that this call goes through the Enterprise Service Bus, but the WebSphere Enterprise Service Bus acts as a router, so no message transformation is needed on Enterprise Service Bus.



## Re-order flow

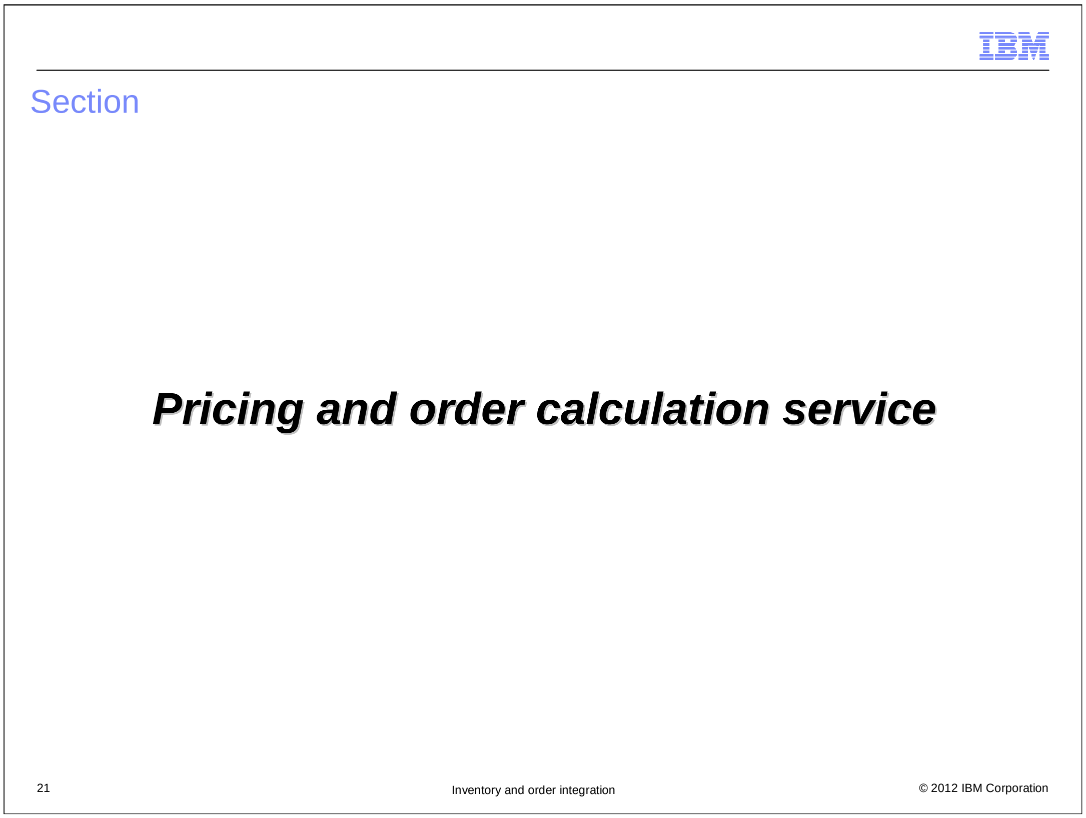
- If it is a "Hold on Transfer" order
  1. Retrieve order details from WebSphere Commerce order subsystem
  2. Create a new shopping cart in WebSphere Commerce
- If it is **NOT** a "Hold on Transfer" order
  1. Retrieve order details from Sterling OMS
  2. Create a new shopping cart in WebSphere Commerce
- Catalog entry's inventory is checked before it can be added to cart

20 Inventory and order integration © 2012 IBM Corporation

On the Order History page, if a shopper selects a previous order and clicks **Re-Order**, WebSphere Commerce gets all catalog entries from the existing order, and then adds them into a new shopping cart.

Based on the existing order status, WebSphere Commerce either gets the order details from the local system or from Sterling OMS. If an order is a "Hold on Transfer" order, WebSphere Commerce gets an order item's details from WebSphere Commerce. Otherwise, WebSphere Commerce uses tag getDataFromSSFS to get the order items by calling Sterling **getCompleteOrderDetails** API.

Before WebSphere Commerce adds an item into the cart, WebSphere Commerce checks an item's inventory status regardless of where the order detail came from, to decide whether it's allowed to be added to the cart.

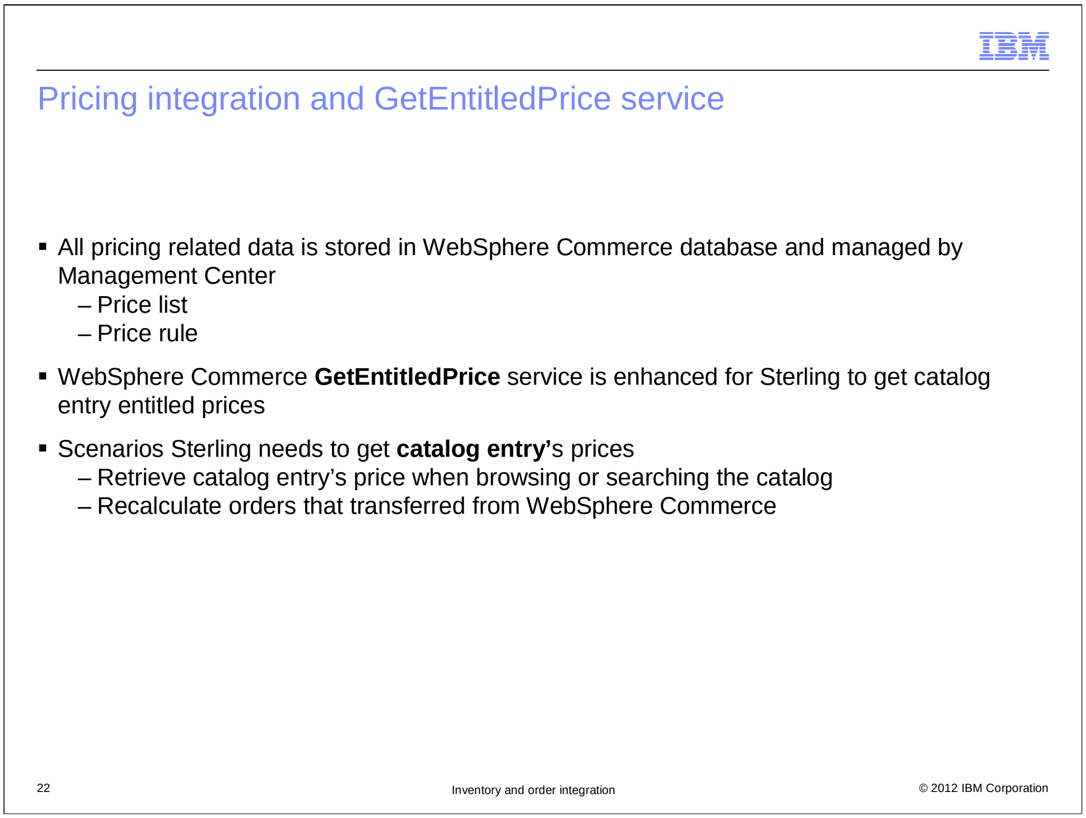


Section

# *Pricing and order calculation service*

21 Inventory and order integration © 2012 IBM Corporation

This section introduces two new services, pricing service and order calculation service, that WebSphere Commerce offers in Feature Pack 5.



## Pricing integration and GetEntitledPrice service

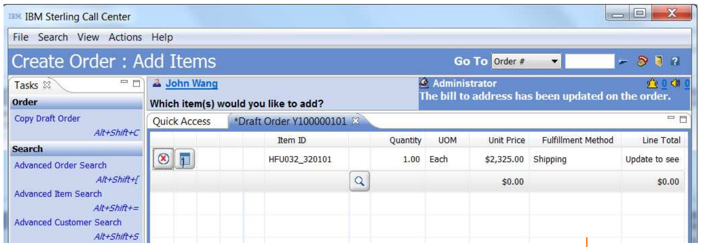
- All pricing related data is stored in WebSphere Commerce database and managed by Management Center
  - Price list
  - Price rule
- WebSphere Commerce **GetEntitledPrice** service is enhanced for Sterling to get catalog entry entitled prices
- Scenarios Sterling needs to get **catalog entry'**s prices
  - Retrieve catalog entry's price when browsing or searching the catalog
  - Recalculate orders that transferred from WebSphere Commerce

For Sterling integration, all the pricing data, including price rules, price lists and price entries are stored in WebSphere Commerce database and managed by WebSphere Commerce Management Center.

In Feature Pack 5, submitting orders from the Sterling Call Center is also supported. Since all pricing related data are managed by WebSphere Commerce, WebSphere Commerce exposes its pricing service GetEntitledPrice, so that Sterling can use it to get catalog entry's entitled price.

This service is called when Sterling browses and searches catalog entries, and when Sterling needs to do order recalculation.

## Pricing integration flow


IBM Sterling Call Center
File Search View Actions Help
Go To Order #
Tasks
Order
Copy Draft Order Alt+Shift+C
Search
Advanced Order Search Alt+Shift+[
Advanced Item Search Alt+Shift+=
Advanced Customer Search Alt+Shift+S
Quick Access
*Draft Order Y100000101

Create Order : Add Items

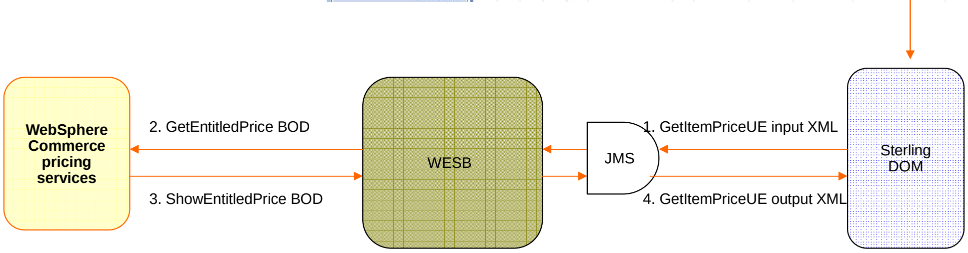
John Wang

Which item(s) would you like to add?

Administrator

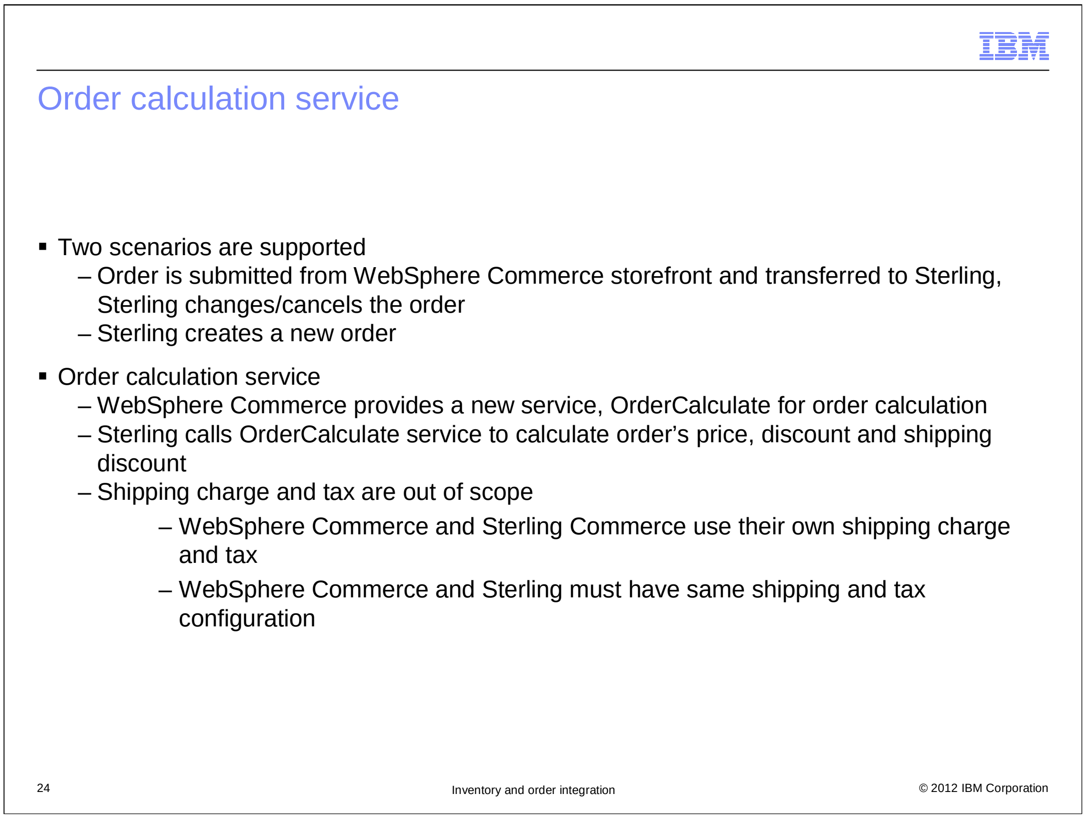
The bill to address has been updated on the order.

| Item ID | Quantity | UOM | Unit Price | Fulfillment Method | Line Total |
|---|---|---|---|---|---|
| HFU032_320101 | 1.00 | Each | $2,325.00 | Shipping | Update to see |
| | | | $0.00 | | $0.00 |


WebSphere Commerce pricing services
2. GetEntitledPrice BOD
3. ShowEntitledPrice BOD
WESB
JMS
1. GetItemPriceUE input XML
4. GetItemPriceUE output XML
Sterling DOM

The diagram on the slide shows how the GetEntitledPrice service is used for pricing.

Whenever Sterling Call Center needs to get a catalog entry price, for example when an item is added into the order, Sterling getItemPriceUE user sends a request to WebSphere Commerce GetEntitledPrice service through synchronous JMS. WebSphere Commerce returns the catalog entry offer price back to Sterling. For this flow, Sterling acts as a service consumer.



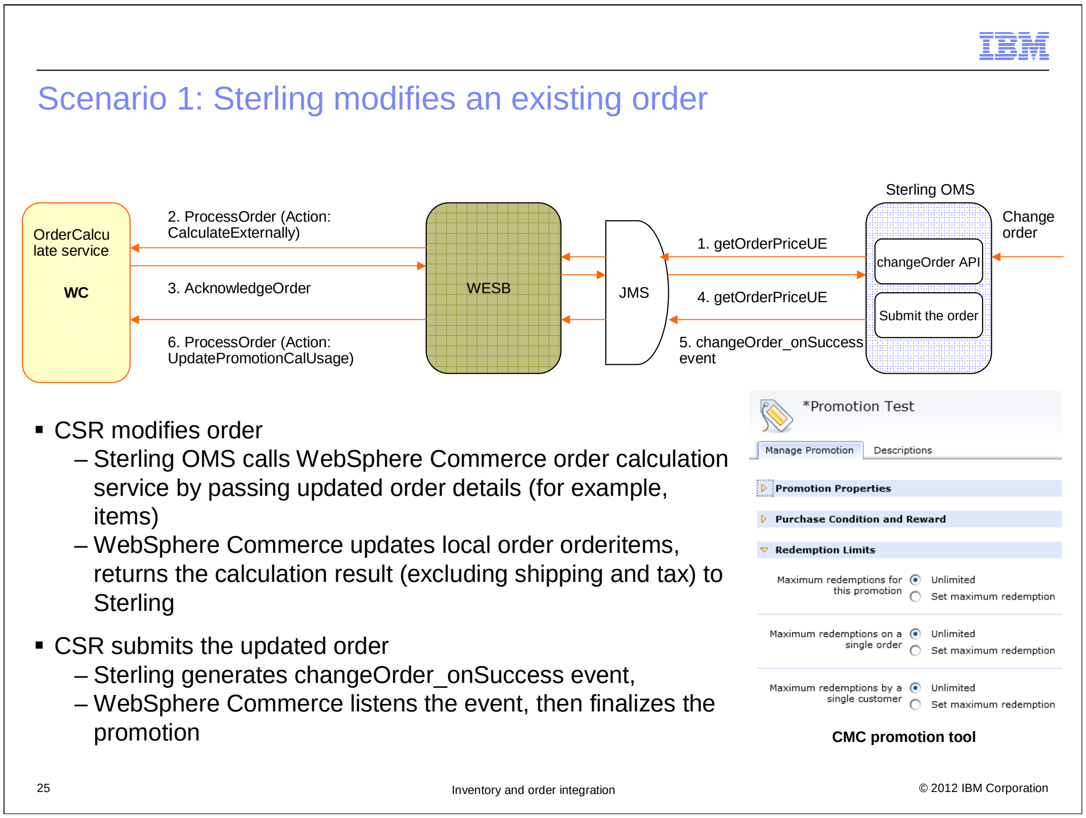
## Order calculation service

- Two scenarios are supported
  - Order is submitted from WebSphere Commerce storefront and transferred to Sterling, Sterling changes/cancels the order
  - Sterling creates a new order
- Order calculation service
  - WebSphere Commerce provides a new service, OrderCalculate for order calculation
  - Sterling calls OrderCalculate service to calculate order's price, discount and shipping discount
  - Shipping charge and tax are out of scope
    - WebSphere Commerce and Sterling Commerce use their own shipping charge and tax
    - WebSphere Commerce and Sterling must have same shipping and tax configuration

24 Inventory and order integration © 2012 IBM Corporation

Besides capturing and calculating orders from WebSphere Commerce, in some situations, Sterling also needs to calculate the order total using WebSphere Commerce pricing and promotions. For example, when Sterling Call Center creates a new order, or when Sterling Call Center changes or cancels an existing order, Sterling needs to calculate or re-calculate the order.

Feature Pack 5 provides a new service, OrderCalculate service. This service allows Sterling to get order calculation from WebSphere Commerce. This service calculates order price, discount and shipping discount, but not the shipping charge and tax. In order to get consistent prices from WebSphere Commerce and Sterling, WebSphere Commerce and Sterling Commerce must have the same shipping and tax configuration.



## Scenario 1: Sterling modifies an existing order

Sterling OMS

Change order

OrderCalculate service

WC

2. ProcessOrder (Action: CalculateExternally)

3. AcknowledgeOrder

6. ProcessOrder (Action: UpdatePromotionCalUsage)

WESB

JMS

1. getOrderPriceUE

4. getOrderPriceUE

5. changeOrder_onSuccess event

changeOrder API

Submit the order

- CSR modifies order
  - Sterling OMS calls WebSphere Commerce order calculation service by passing updated order details (for example, items)
  - WebSphere Commerce updates local order orderitems, returns the calculation result (excluding shipping and tax) to Sterling
- CSR submits the updated order
  - Sterling generates changeOrder_onSuccess event,
  - WebSphere Commerce listens the event, then finalizes the promotion

*Promotion Test

Manage Promotion | Descriptions

Promotion Properties

Purchase Condition and Reward

Redemption Limits

Maximum redemptions for this promotion: ● Unlimited ○ Set maximum redemption

Maximum redemptions on a single order: ● Unlimited ○ Set maximum redemption

Maximum redemptions by a single customer: ● Unlimited ○ Set maximum redemption

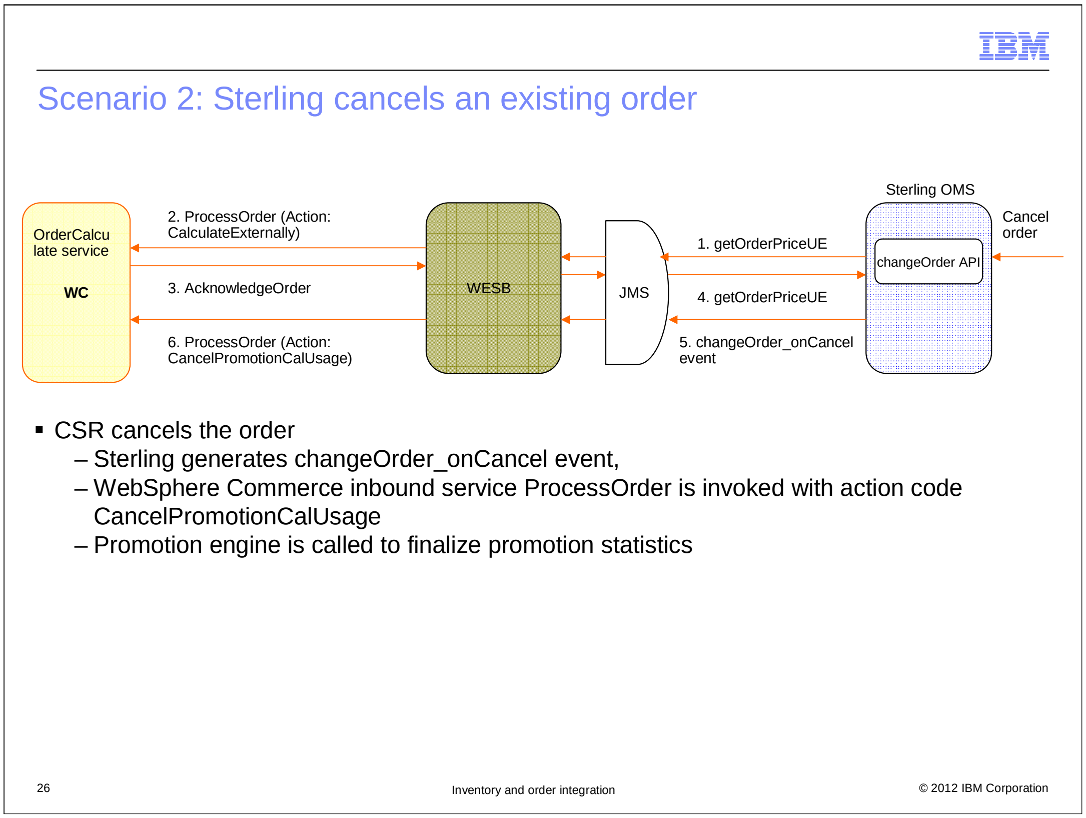
**CMC promotion tool**

25 Inventory and order integration

This slide shows how Sterling calls the OrderCalculate service during order modification.

When Sterling updates an existing order, Sterling passes all order items in the order to WebSphere Commerce by calling ProcessOrder inbound service. The OrderCalculate service is called to recalculate the order, the new price is returned to Sterling.

When Sterling submits the final order, ChangeOrder_ON_SUCCESS event is triggered. WebSphere Commerce listens for the event, and invokes the ProcessOrder service with UpdatePromotionCalUsage action code. From there the promotion statistics gets updated. If a promotion code is used for the order, the promotion engine marks this promotion code as used. If a promotion has Redemption Limits defined**,** the redemption usage count is updated accordingly by the promotion engine.

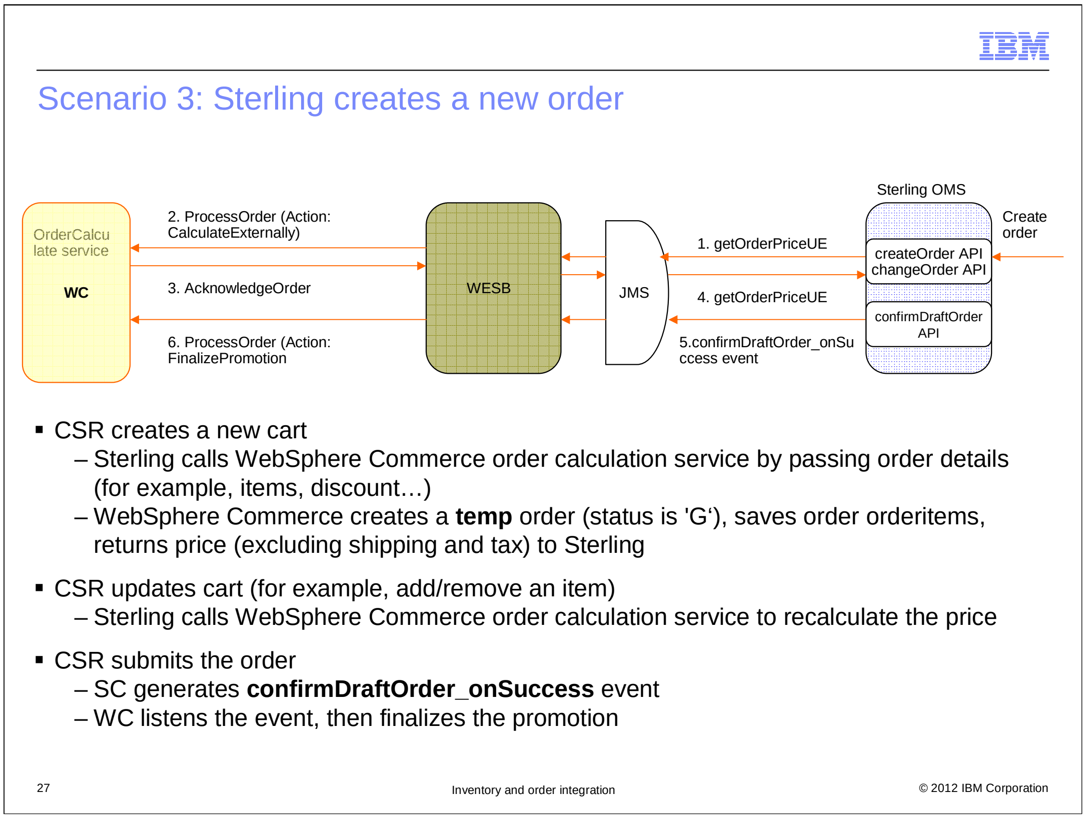


## Scenario 2: Sterling cancels an existing order

- CSR cancels the order
  - Sterling generates changeOrder_onCancel event,
  - WebSphere Commerce inbound service ProcessOrder is invoked with action code CancelPromotionCalUsage
  - Promotion engine is called to finalize promotion statistics

26 Inventory and order integration © 2012 IBM Corporation

The order cancelation is initialized from Sterling Call Center. When Sterling cancels an order, it triggers the ChangeOrder_ON_CANCEL event. WebSphere Commerce listens for the event and calls the WebSphere Commerce ProcessOrder service with action code CancelPromotionCalUsage. Then the promotion engine is called to finalize the promotion statistics. For example, if a promotion code was used for the order, promotion code is marked as unused code. If a promotion has Redemption Limits defined, the promotion statistics is updated.



## Scenario 3: Sterling creates a new order

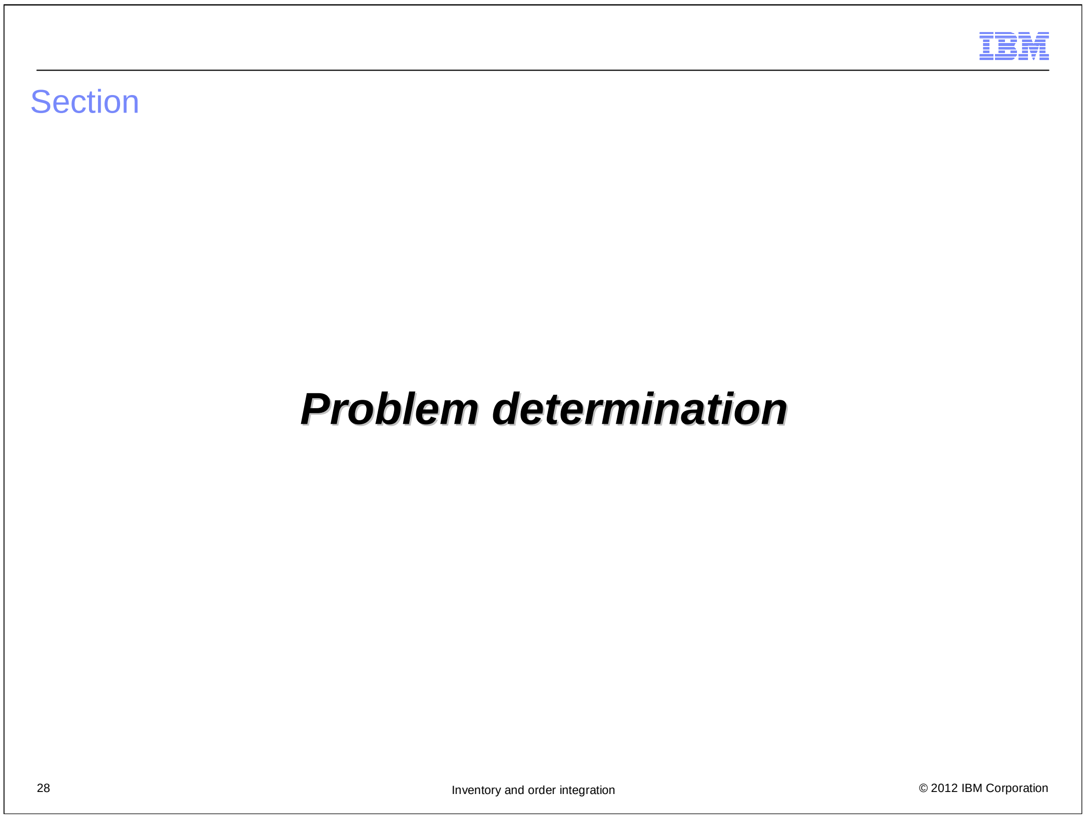
- CSR creates a new cart
  - Sterling calls WebSphere Commerce order calculation service by passing order details (for example, items, discount…)
  - WebSphere Commerce creates a **temp** order (status is 'G'), saves order orderitems, returns price (excluding shipping and tax) to Sterling
- CSR updates cart (for example, add/remove an item)
  - Sterling calls WebSphere Commerce order calculation service to recalculate the price
- CSR submits the order
  - SC generates **confirmDraftOrder_onSuccess** event
  - WC listens the event, then finalizes the promotion

This slide shows how Sterling uses OrderCalculation service to create a new order. When Sterling creates a new order, Sterling calls its getOrderPriceUE user exit to pass order details to WebSphere Commerce. The order details include order line items, discount, promotion code, order number, et cetera.

WebSphere Commerce inbound service ProcessOrder is called with action code CalculateExternally. After WebSphere Commerce gets the order details, WebSphere Commerce creates a temporary order in its local database, calculates the order price, and returns order price to Sterling.

Sterling may call the OrderCalculate service several times before it submits the order. For example, when CSR add a product to a cart, or remove an item from the cart, CSR may refresh the cart several times. Each time Sterling gets a new order calculation from WebSphere Commerce.

After Sterling submits the final order, confirmDraftOrder_onSuccess event is triggered by the Sterling. WebSphere Commerce listens for this event and invokes the WebSphere Commerce ProcessOrder service to finalize the promotion.

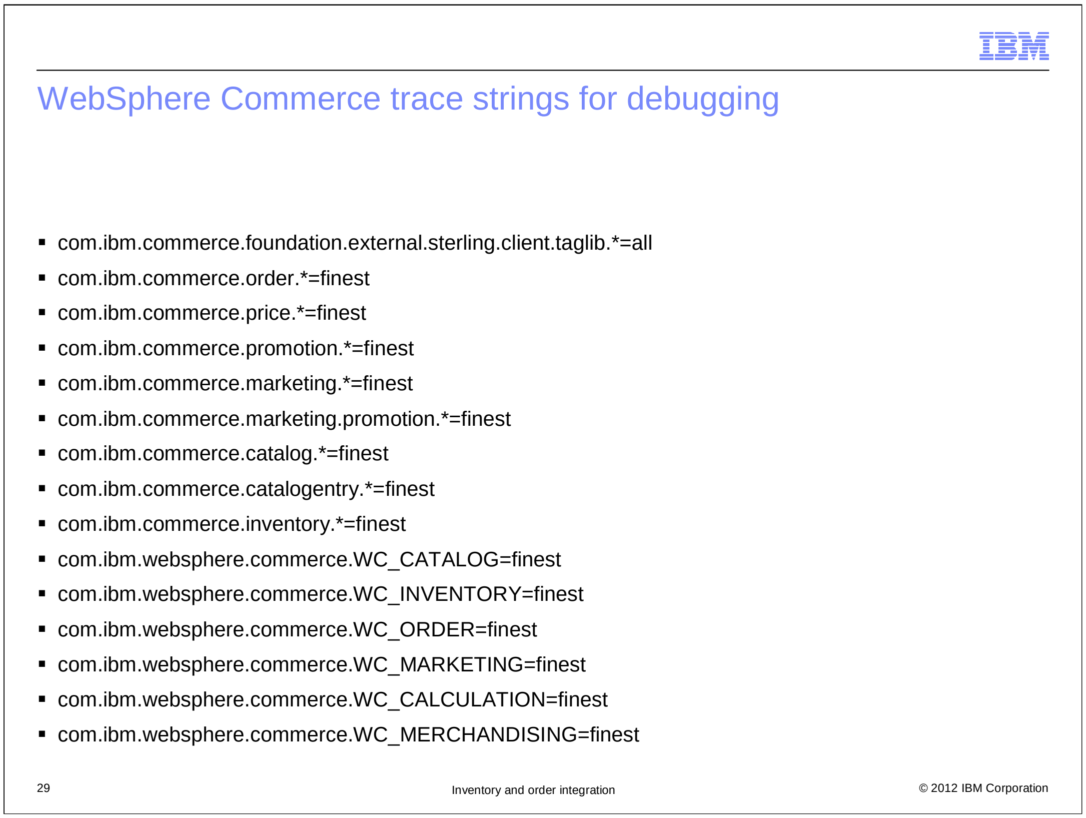


Section

## *Problem determination*

28 Inventory and order integration © 2012 IBM Corporation

This section shows you how to enable integration related trace strings from WebSphere Commerce server, WebSphere Enterprise Service Bus mediation module and Sterling integration server.



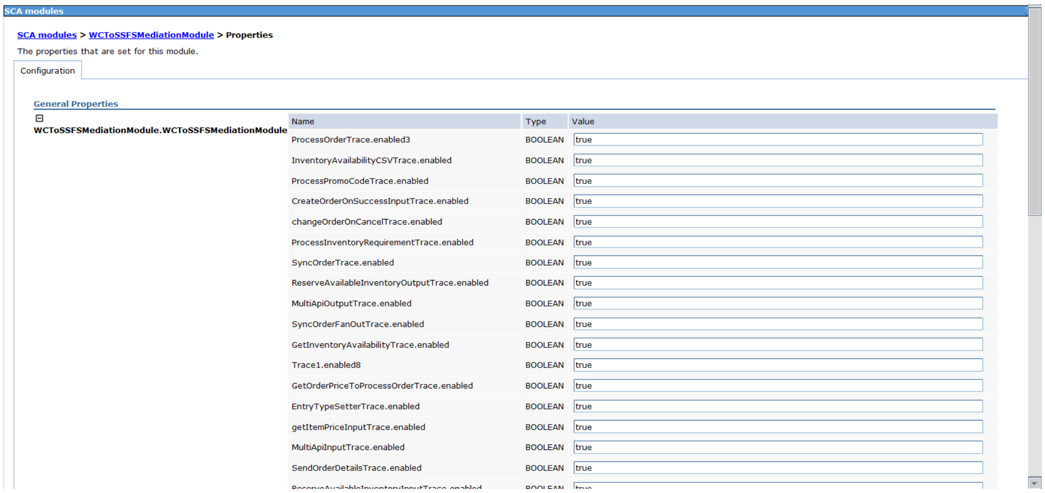
## WebSphere Commerce trace strings for debugging

- com.ibm.commerce.foundation.external.sterling.client.taglib.*=all
- com.ibm.commerce.order.*=finest
- com.ibm.commerce.price.*=finest
- com.ibm.commerce.promotion.*=finest
- com.ibm.commerce.marketing.*=finest
- com.ibm.commerce.marketing.promotion.*=finest
- com.ibm.commerce.catalog.*=finest
- com.ibm.commerce.catalogentry.*=finest
- com.ibm.commerce.inventory.*=finest
- com.ibm.websphere.commerce.WC_CATALOG=finest
- com.ibm.websphere.commerce.WC_INVENTORY=finest
- com.ibm.websphere.commerce.WC_ORDER=finest
- com.ibm.websphere.commerce.WC_MARKETING=finest
- com.ibm.websphere.commerce.WC_CALCULATION=finest
- com.ibm.websphere.commerce.WC_MERCHANDISING=finest

This slide lists all the trace strings you need for debugging WebSphere Commerce related issues. You can enable all of them or some of them from WebSphere Application Server administration console depending on what areas your issues are.

## WebSphere Enterprise Service Bus trace

- WebSphere Commerce business objects and Sterling business objects are transformed by XSLT files
  - \WCToSSFSMediationModuleApp.ear\WCToSSFSMediationModule.jar\xslt\
- Applications --> SCA modules --> WCToSSFSMediationModule --> Module Properties



SCA modules

SCA modules > WCToSSFSMediationModule > Properties

The properties that are set for this module.

Configuration

General Properties

⊟ WCToSSFSMediationModule.WCToSSFSMediationModule

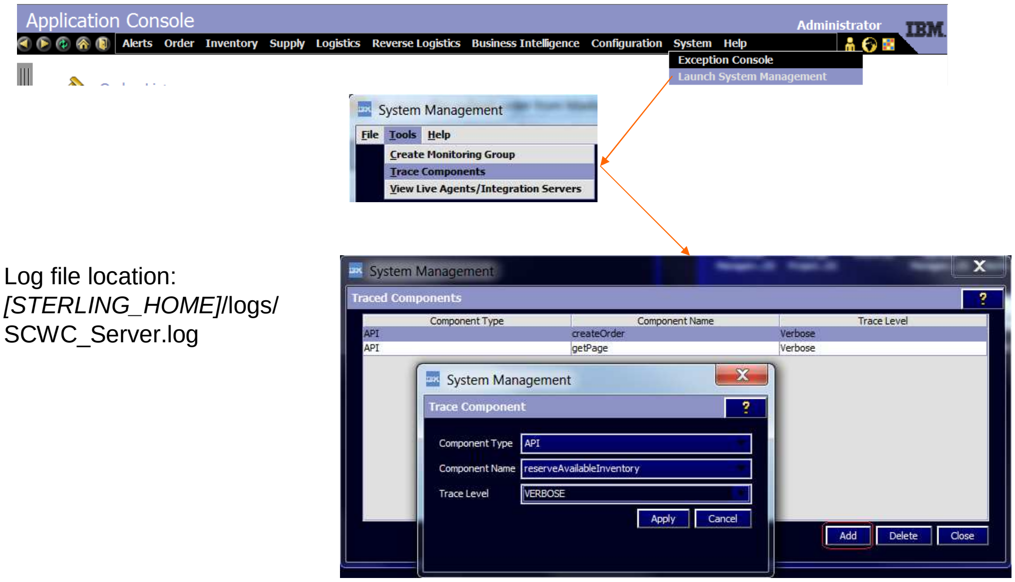
| Name | Type | Value |
|---|---|---|
| ProcessOrderTrace.enabled3 | BOOLEAN | true |
| InventoryAvailabilityCSVTrace.enabled | BOOLEAN | true |
| ProcessPromoCodeTrace.enabled | BOOLEAN | true |
| CreateOrderOnSuccessInputTrace.enabled | BOOLEAN | true |
| changeOrderOnCancelTrace.enabled | BOOLEAN | true |
| ProcessInventoryRequirementTrace.enabled | BOOLEAN | true |
| SyncOrderTrace.enabled | BOOLEAN | true |
| ReserveAvailableInventoryOutputTrace.enabled | BOOLEAN | true |
| MultiApiOutputTrace.enabled | BOOLEAN | true |
| SyncOrderFanOutTrace.enabled | BOOLEAN | true |
| GetInventoryAvailabilityTrace.enabled | BOOLEAN | true |
| Trace1.enabled8 | BOOLEAN | true |
| GetOrderPriceToProcessOrderTrace.enabled | BOOLEAN | true |
| EntryTypeSetterTrace.enabled | BOOLEAN | true |
| getItemPriceInputTrace.enabled | BOOLEAN | true |
| MultiApiInputTrace.enabled | BOOLEAN | true |
| SendOrderDetailsTrace.enabled | BOOLEAN | true |
| ReserveAvailableInventoryInputTrace.enabled | BOOLEAN | true |

In order to debug the WebSphere Enterprise Service Bus, you need to know how the messages are mapped in the Mediation module, and how to turn on the message trace.

In the WebSphere Enterprise Service Bus mediation module, Sterling business objects and WebSphere Commerce business objects are transformed using XSLT files. You can find the XSLT files in the folder shown on the slide.

In the WebSphere Enterprise Service Bus mediation module, all the trace nodes are turned off by default. No message specific trace is printed out in system log file. If you need to enable message trace for debug purposes, you can enable it from WebSphere Enterprise Service Bus integration solution console. You need to navigate to the Module Properties screen by following the path shown on the slide, change the value for that specific trace node to “true.” You don’t need to restart the server after you change the trace setting.

# Sterling and WebSphere Commerce integration server trace



| Component Type | Component Name | Trace Level |
|---|---|---|
| API | createOrder | Verbose |
| API | getPage | Verbose |

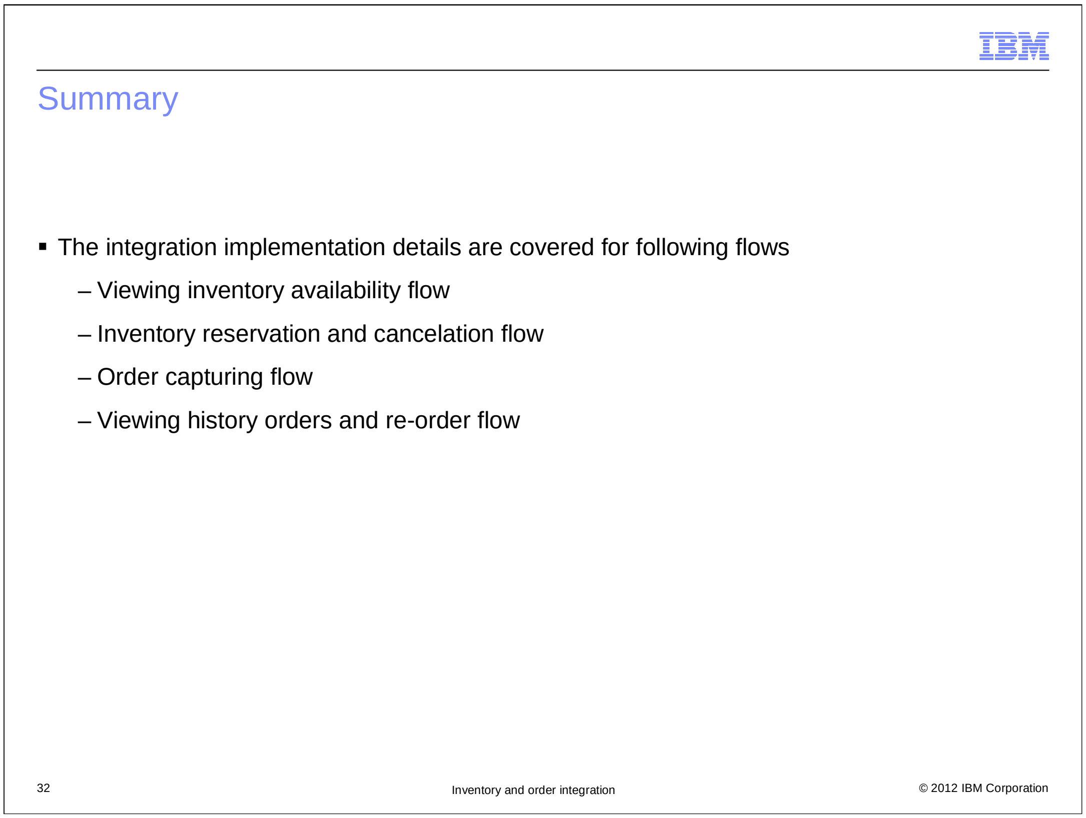
Component Type: API

Component Name: reserveAvailableInventory

Trace Level: VERBOSE

- Log file location: *[STERLING_HOME]*/logs/ SCWC_Server.log

WebSphere Commerce integrates with Sterling Commerce by calling Sterling APIs. To enable Sterling APIs trace and set the trace level, you need to do in Sterling’s System Management Console. You can follow the path shown on the slide to find the System Management console. The location for the trace file is shown on this slide too.

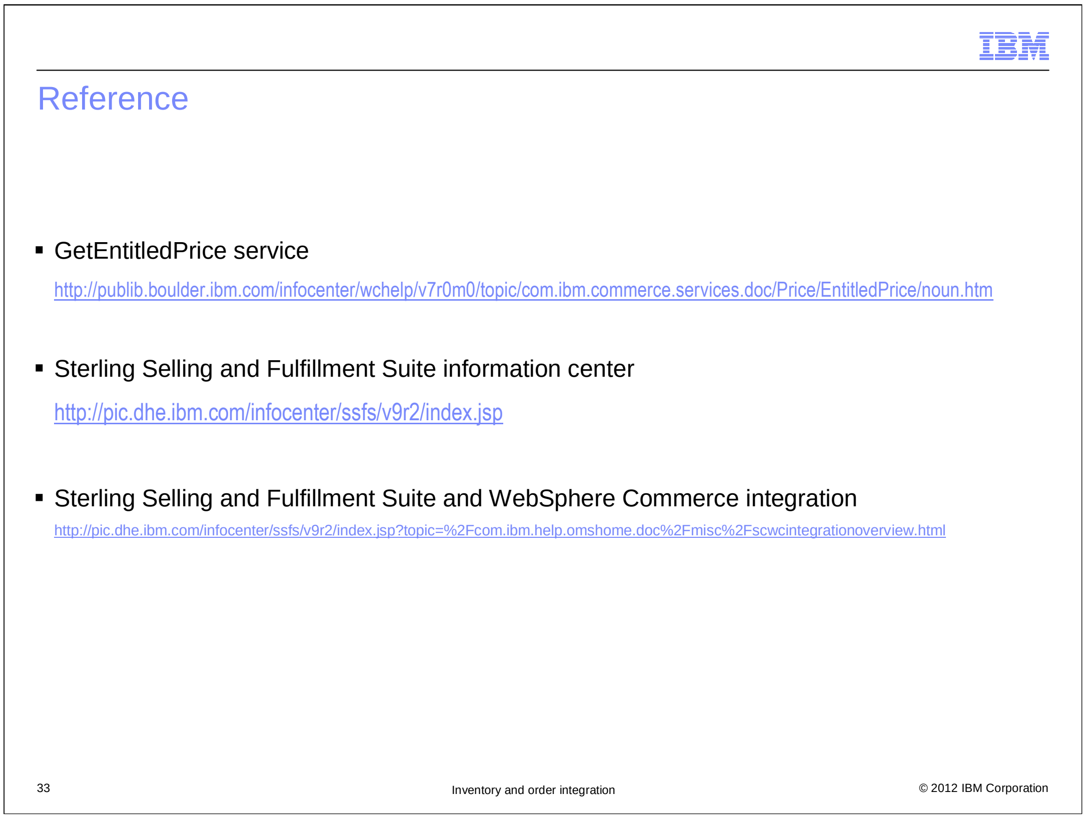


## Summary

- The integration implementation details are covered for following flows
  - Viewing inventory availability flow
  - Inventory reservation and cancelation flow
  - Order capturing flow
  - Viewing history orders and re-order flow

32 Inventory and order integration © 2012 IBM Corporation

During this presentation you have learned how inventory and order integration are implemented. The integration implementation details are covered for viewing inventory availability flow, inventory reservation and cancelation flow, order capturing flow, and viewing history orders and re-order flow



## Reference

- GetEntitledPrice service

  http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.services.doc/Price/EntitledPrice/noun.htm

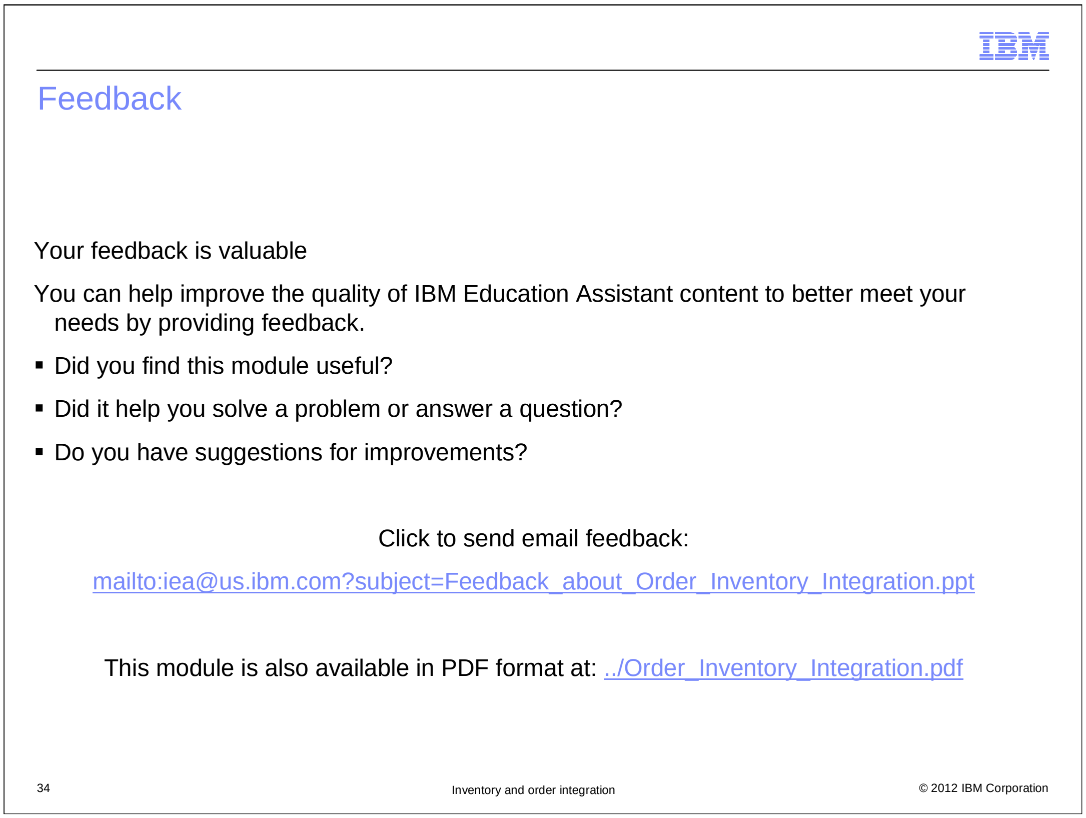
- Sterling Selling and Fulfillment Suite information center

  http://pic.dhe.ibm.com/infocenter/ssfs/v9r2/index.jsp

- Sterling Selling and Fulfillment Suite and WebSphere Commerce integration

  http://pic.dhe.ibm.com/infocenter/ssfs/v9r2/index.jsp?topic=%2Fcom.ibm.help.omshome.doc%2Fmisc%2Fscwcintegrationoverview.html

This slide contains some useful references for understanding the WebSphere Commerce and Sterling Commerce Integration. You should be aware that all integration related documents are now in the Sterling Information Center. WebSphere Commerce Information Center only keeps the integration documents for the releases before Feature Pack 5.



## Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

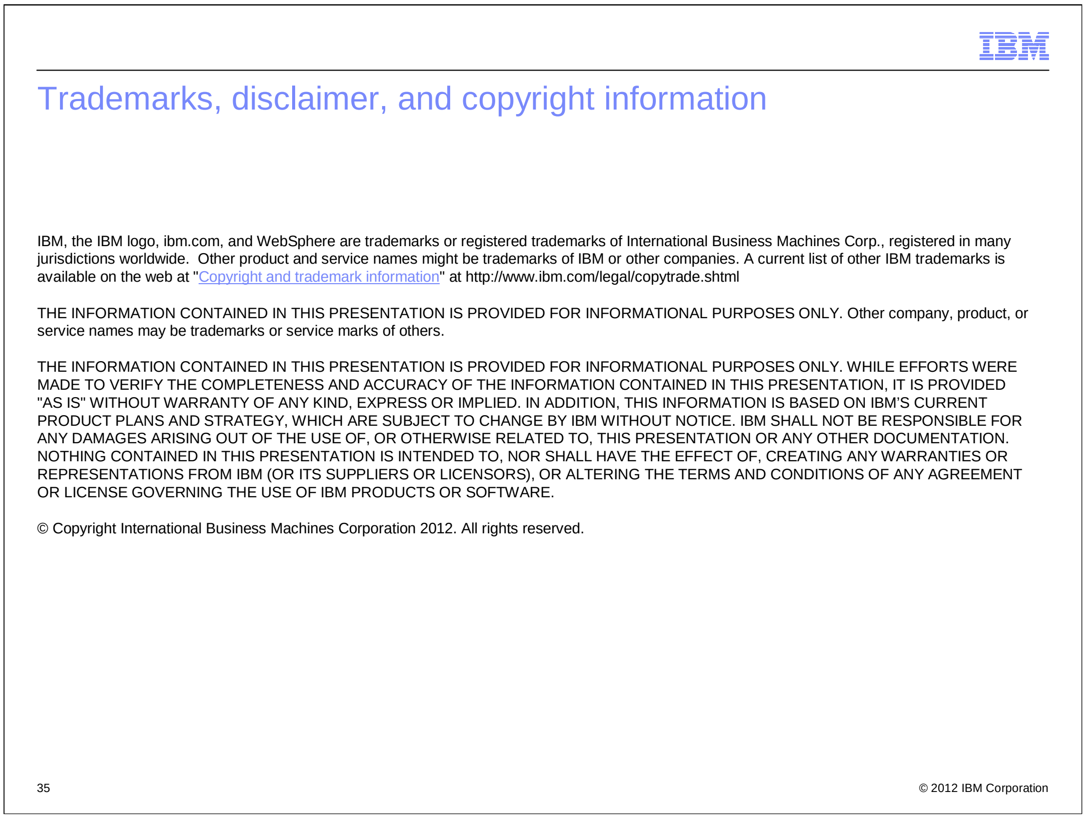
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_Order_Inventory_Integration.ppt

This module is also available in PDF format at: ../Order_Inventory_Integration.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere 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 other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.