This presentation covers DOM integration in feature pack 5.

Agenda

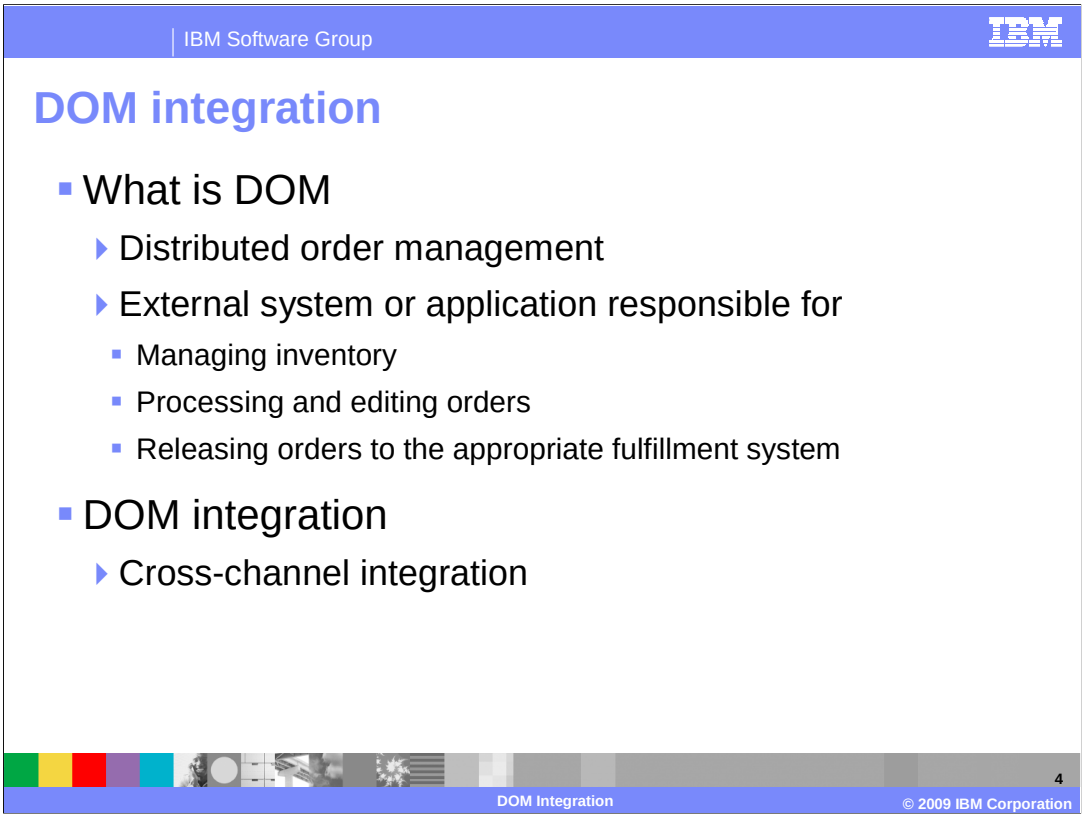
- DOM integration overview
- Store location
- Inventory availability
- Order fulfillment
- Integration points
- Restrictions and limitations

First, a brief discussion of DOM integration is provided. An introduction to the main components will follow, including store location, inventory availability, and order fulfillment. Next, integration points are reviewed. Finally, some restrictions and limitations are covered.

Section

DOM integration overview

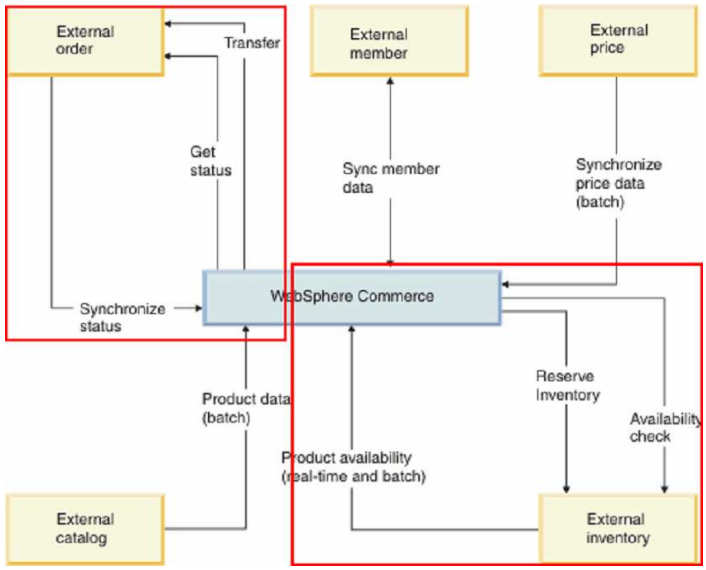
This section provides an overview of DOM integration.



DOM stands for distributed order management. A DOM system can be a customer's existing order management system or a vendor order management system. The external order management system is responsible for processing the order, editing the order and releasing the order to the appropriate fulfillment system. In most cases, inventory is also managed by this external system.

DOM integration allows customers to process their online orders captured by WebSphere® Commerce though an external order management system. DOM integration is part of the cross-channel integration strategy. It is a backend system integration which enables distributed order management systems to provide a comprehensive coverage of the order life cycle – from capture to fulfillment – across channels.

Services exposed by DOM integration



Services provided by the DOM integration are shown here. They include outbound services and inbound services.

With DOM integration, WebSphere Commerce is no longer responsible for managing inventory or orders. The areas highlighted in red show the commerce subsystems that are replaced by external systems with this solution. Two-way communication exists between WebSphere Commerce and the external systems that provides the online store with order status changes and updated inventory information.

New components and enhancements

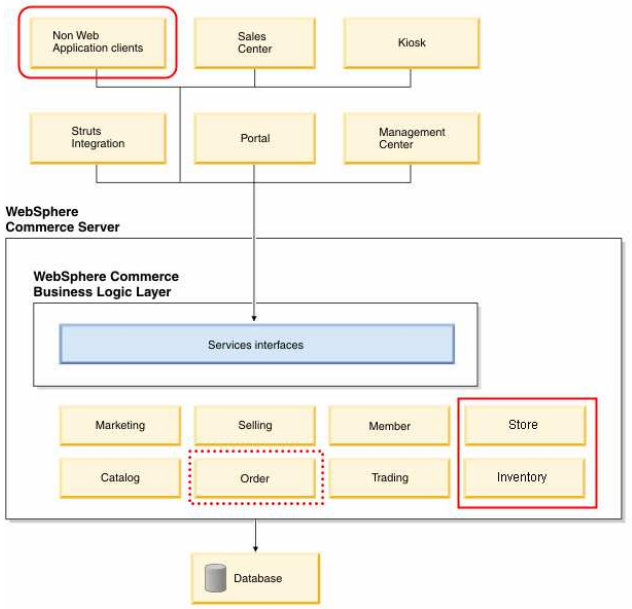
- New store component with services
  - Store location
- New inventory component with services for DOM integration
  - Cache inventory
  - Check inventory availability
  - Process inventory requirements
- Enhancements to the order component for DOM integration
  - Support buy online, pickup in-store and order online, pay in-store

New components and enhancements that have been added to support DOM integration are shown here.

Within WebSphere Commerce, there are three main areas impacted by the DOM integration solution. This presentation will discuss each of these three areas in more detail in the coming sections.

The new store component provides a set of services to support the store location feature. A new inventory component provides services to support inventory caching, inventory availability checking and inventory requirement processing. The existing order component has been enhanced to support the buy online, pickup in-store and order online, pay and pickup in-store checkout flows.

WebSphere Commerce changes



This diagram highlights the changes to WebSphere Commerce just discussed. From a services perspective, two new components have been added, Store and Inventory. They are highlighted by the red box in the bottom right. The order component, highlighted with a dotted red line has been modified. The DOM integration solution defines Web services the DOM system can use to communicate with WebSphere Commerce. These inbound services calls are represented by the Non Web Application clients box in the upper left.

DOM integration related store enhancements

- New shopping flows
  - Store location
- Enhanced and changed flows
  - Catalog browsing flows
  - Checkout flows: buy online, pickup in-store

With DOM integration enabled, the Madisons starter store provides a store location feature and new buy online, pickup in-store shopping flow.

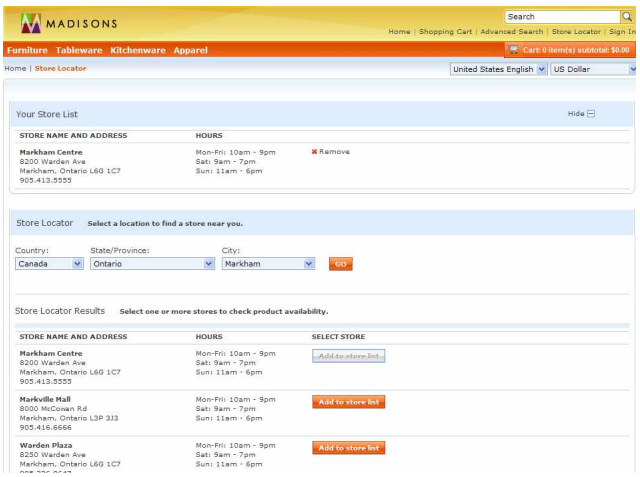
Store location allows shoppers to find nearby physical stores. It also allows shoppers to select their favorite stores and add them into their store list.

Catalog browsing enhancement enables shoppers to view product availability for stores in their store list.

Checkout flows have been changed for DOM integration. Shoppers can order products online then select a physical store to pick them up.

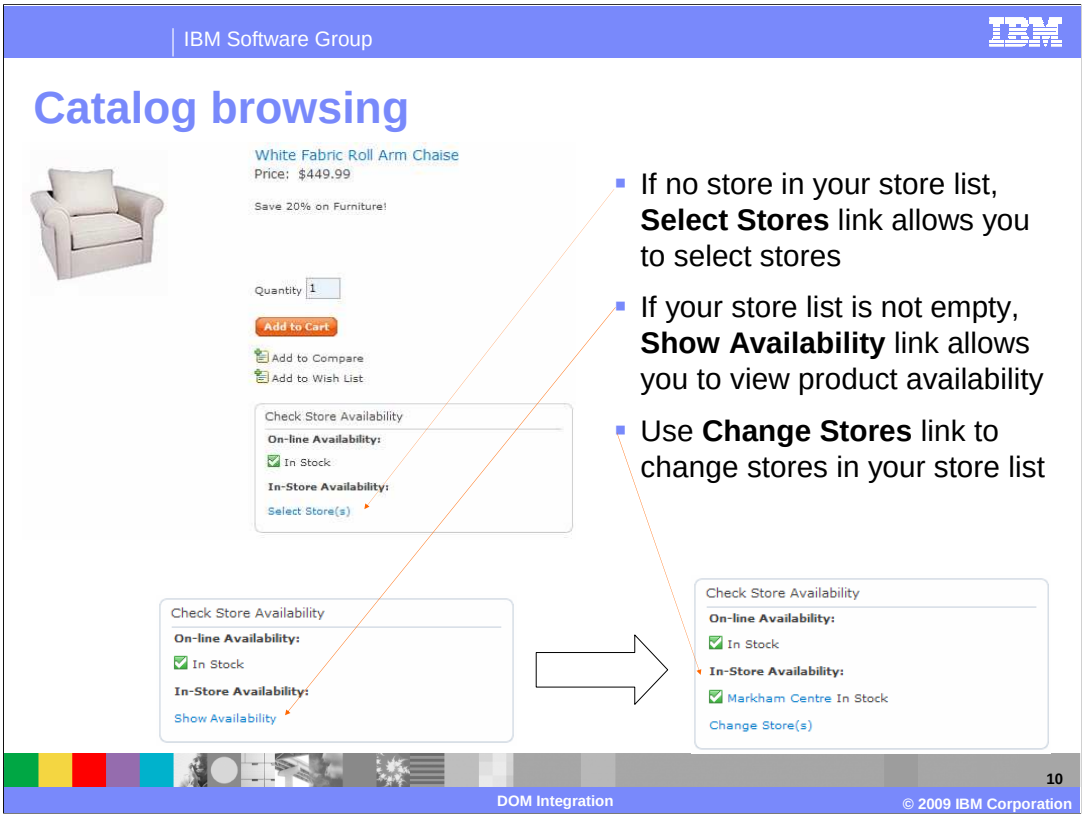
Store locator

- Locate physical stores
- Add physical stores into your store list
- Stores can be located by country, state and city
- Store locator can be accessed from many places



The store locator page of the Madisons starter store is shown here. This page allows you to search for physical stores based on your selection of country, state and city.

You can add the stores into your store list. When you browse the catalog, you can view product availability for stores in your store list. You can access the store locator from several places, such as the store header, product display page and the store select page during checkout.

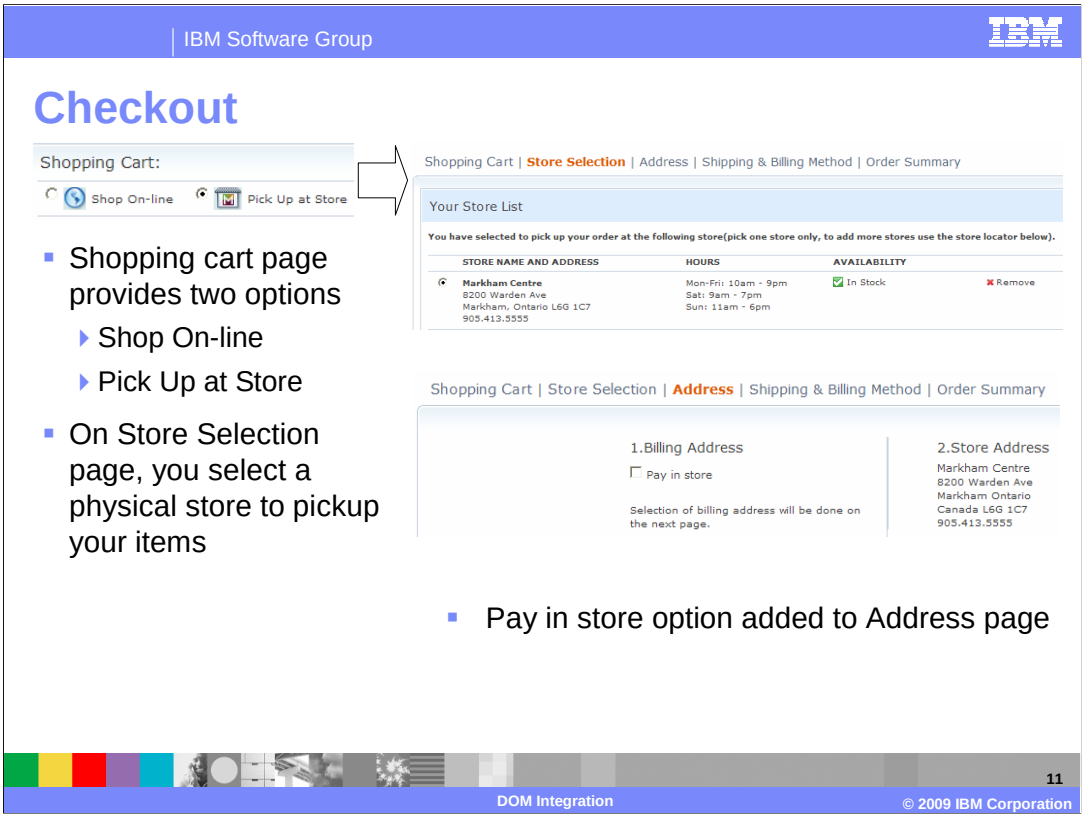


A DOM integration enabled product display page is shown here. On this page you can check store availability of a product both online and in physical stores.

If you have not added any stores to your store list, you can do so by clicking the Select Stores link in the Check Store Availability zone. This link will redirect you to the store locator page.

If you have stores in your store list, you see the Show Availability link instead of the Select Stores link. You can click it to view the product availability for your selected stores.

You can update your store list by clicking the Change Stores link.

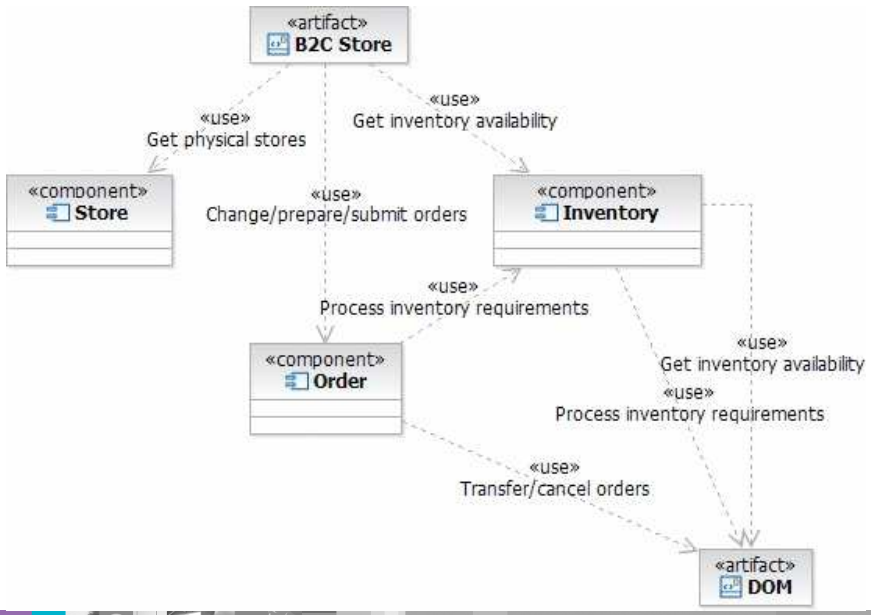


On the shopping cart page, the Pick up at Store option is provided. If you select Pick up at Store, the breadcrumb trail for checkout contains two more steps: Store Selection and Address.

On the Store Selection page, you can select a physical store to pickup your order.

The Address page displays the physical store's address and contact information. It also gives you the option to pay at the physical store. If you leave this option unchecked, you need to input billing information and continue the checkout process to pay online.

Components and interactions



The components involved in the DOM integration implementation are shown here. This diagram also shows how these components interact with each other.

You will see more details of each component in later sections.

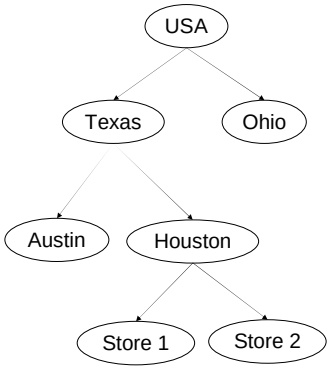
Section

Store location

This section introduces the store location feature.

Store location

- Component services
  - **GetGeoNode**: retrieve the nodes
  - **GetPhysicalStore**: retrieve the physical stores
- Physical store's data is managed by WebSphere Commerce
  - New database tables
    - STLOC, STLOCDS, STLOCATTR, GEONODE, GEONDDS
    - Geo code (latitude, longitude) in table STLOC supports integration with map provider interface
  - Geo tree is used to manage store locations
- Stores are located by narrowing down geographical region
  - For examples: USA > Texas > Houston



Two new component services are used to find the stores. The GetGeoNode service is used for retrieving regions, such as countries, states, and cities. The GetPhysicalStore service is used for retrieving a list of physical stores in a selected region.

A physical store's data, such as address, contact information and operation hours, are stored in the WebSphere Commerce database. Several new database tables have been added to capture the data.

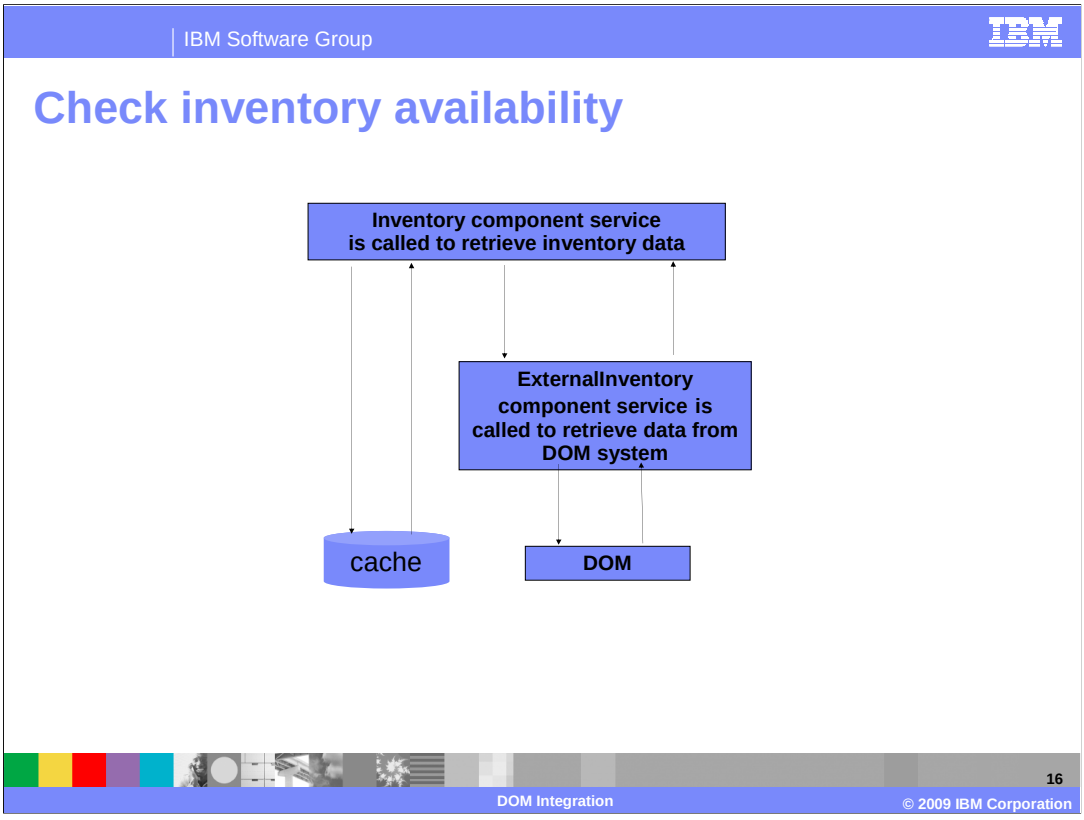
The STLOC table contains Geo code data, otherwise known as latitude and longitude. You can use Geo code data to integrate with a map provider's interfaces to provide additional services, such as showing a store location on a map.

Stores are organized by Geo tree. The root node of this tree is a country, and other nodes are states and cities.

Section

Inventory availability with caching

This section introduces inventory caching for DOM integration.

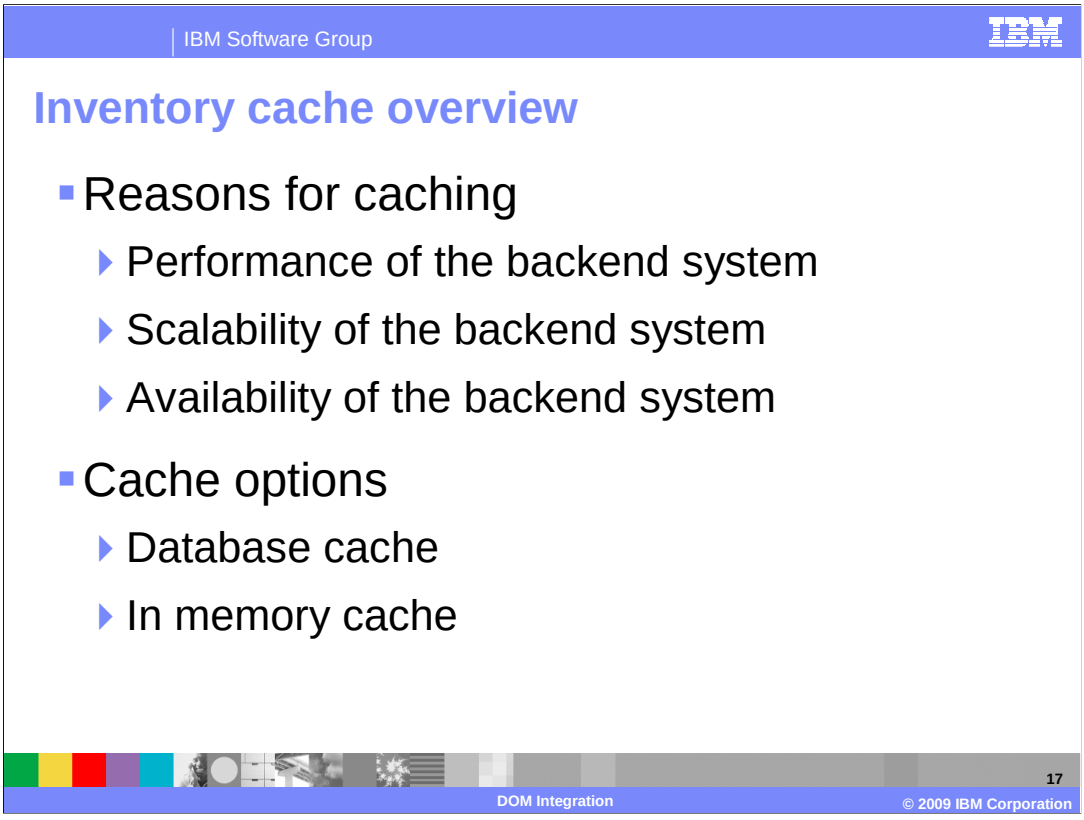


The new inventory component includes services for displaying inventory availability. The inventory availability data includes inventory status, available quantity, availability date and offset, and last update date.

In the store front, if a shopper triggers a check of an item's inventory availability the inventory component receives the request.

A new service is called to retrieve the inventory availability information. This service will retrieve the information either from the cache or from the DOM system, depending on your inventory configuration.

If information needs to be retrieved from the DOM system, an outbound service of the new **ExternalInventory** component is called.



With DOM integration enabled, WebSphere Commerce has to exchange inventory information with one or more backend systems. Since the reliability and performance of backend systems is not known, inventory data can be cached by WebSphere Commerce. Inventory caching addresses three backend concerns.

A backend system may not respond to real-time requests in a timely manner. It may be designed for batch processing, not for real-time requests.

A backend system may not be able to handle the amount of traffic generated by the Web channel during peak periods. It may be designed for back office traffic.

A backend system may require regular down times for maintenance. The network connection to the backend system might not be reliable.

DOM integration provides two options for inventory cache: database cache and in memory cache.

Database cache

- Cached data is available at system start up
- New table **INVAVL** is used for inventory cache
- Massload utility can be used to batch load the data into INVAVL table

The database cache provides inventory data locally to the WebSphere Commerce system. Cached data is available as soon as the system starts up. Inventory requests to the backend systems are only needed when cached data expires.

A new database table INVAVL has been added in this release to provide the database cache. You can use the massload utility to load initial data into this table.

In memory cache

- Alternate caching strategy when batch load is not required
- Uses WebSphere distributed object cache
  - Supports clustered environment
  - WebSphere Cache Monitor
- Cache instance is created when enabling dom-integration feature
- Cache entries are saved in cache mapping table

An in memory cache strategy can be used if batch load is not required. Records in memory can be retrieved faster than in the database, but there is a limit to the number of records in the cache. Records not in the cache need to be retrieved from the backend system.

In memory cache is implemented by WebSphere distributed object cache. This solution was chosen because it works in a clustered environment and can be monitored using the WebSphere Cache Monitor. WebSphere distributed object cache interfaces are simple interfaces for the dynamic cache. Using these interfaces, JEE applications and system components can cache and share Java objects by storing a reference to the object in the cache. This default instance is bound to the global Java Naming and Directory Interface (JNDI) namespace using the name services/cache/distributedmap.

Cache entries are saved in a cache mapping table. Each cache entry corresponds to an inventory availability record. You do not need to create the cache instance if you decide to use in memory cache, the dom-integration feature enablement process has created it for you.

Inventory cache record

- Information contained in a cached record
  - Catalog entry
  - Online store ID or physical store ID
  - Inventory status
  - Available quantity
  - Availability date
  - Availability offset
- For each combination of category entry and store location, there is only one record in the cache

The data in a cached record is shown here.

The value for inventory status can be available, backordered, or unavailable

Availability offset is used when the availability date is based on a lead time (offset) instead of an absolute date.

A catalog entry record in the cache can either belong to an online store or a physical store but it cannot belong to both.

Note that after DOM integration is enabled, the online store's inventory data is cached as well.

Inventory cache configuration

- Configuration steps
  - Define inventory cache policies in database table INVCNF
  - Assign inventory cache policy to catalog entries and store locations in database table INVCNFREL
- Column FLAG in table INVCNF uses bit field to specify inventory configuration options
  - 1 - Use WebSphere distributed object cache
  - 2 - Use database cache
  - 4 - Retrieve inventory availability from DOM
  - 8 - Decrease cached inventory when an order is submitted
  - 16 - Update cached inventory after information is retrieved from DOM

The DOM integration design allows you to create multiple inventory configuration policies. You can apply these policies at the catalog entry level for each store location.

To configure the inventory cache, you first need to create inventory cache policies, next you need to associate them with catalog entries and store locations.

The FLAG column in table INVCNF uses a bit field to specify inventory configuration options. The bit values and their corresponding configuration options are shown here for your reference.

Assigning cache configuration policies

- Associate a catalog entry or store location or both with an inventory configuration policy in table INVCNFREL
- NULL rules
  - If CATENTRY_ID is NULL- apply to all catalog entries in the store
  - If STORE_ID and STLOC_ID are NULL- apply to this catalog entry at all locations
  - If CATENTRY_ID, STORE_ID and STLOC_ID are NULL- apply to all catalog entries at all locations
  - STORE_ID or STLOC_ID - one **must** be NULL
- Example

INVCNFREL_ID	CATENTRY_ID	STORE_ID	STLOC_ID	INVCNF_ID
10001	11202	11301	NULL	10001
10002	NULL	11301	NULL	10002
10003	NULL	NULL	NULL	10003

After you create inventory configuration policies in table INVCNF, you need to associate the policies with your catalog entries and store locations.

NULL rules are shown here. These rules are used to simplify your configuration so that you can apply one inventory policy to a large number of catalog entries, stores or combination of catalog entries and stores.

Inventory cache update

- Cache is updated when
  - Cache record has expired
    - Retrieves the inventory availability records from DOM and refreshes the cache
  - An order is submitted for processing and the cache is decremented
  - Inventory records are batch loaded
  - The DOM system makes a SyncInventoryAvailability inbound service call

A cache record is updated after it expires. An outbound service GetInventoryAvailability is called to retrieve the inventory availability data from the DOM system. It then updates the record in the local cache.

Inventory cache expiration is decided by the configuration in database table INVCNF.

The cache is updated if an order is submitted causing the cached inventory to be decreased.

The cache can be also updated if a site administrator batch loads data into the cache or the DOM system calls the inbound service SyncInventoryAvailability to synchronize the cache with DOM inventory.

Section

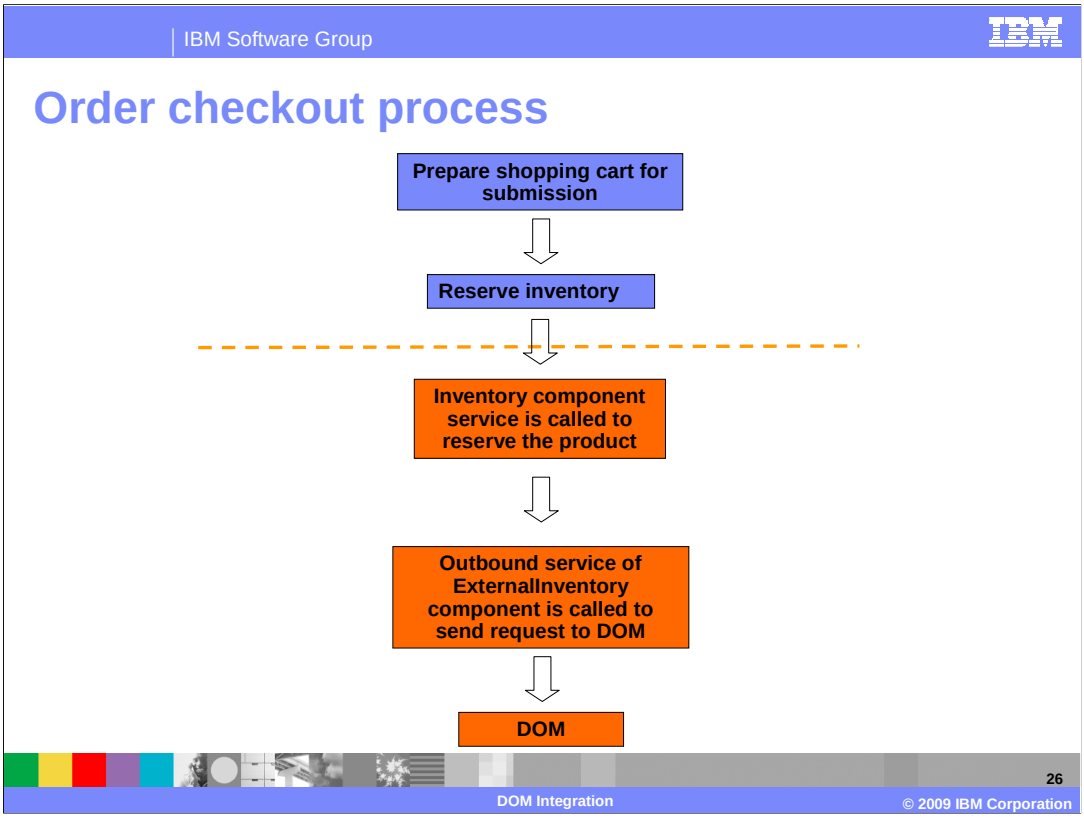
Order fulfillment

This section covers order fulfillment processes related with DOM integration.

Order fulfillment options

- Buy online, deliver to specified address
- Buy online, pickup in-store
- Order online, pay and pickup in-store

Three fulfillment options are provided with DOM integration. The first is the traditional online shopping fulfillment method, buy online and have the order shipped to an address. Two new fulfillment options are now supported as well. They are buy online, pickup in-store and order online, pay and pickup in-store. The details of each method are discussed later in this section.

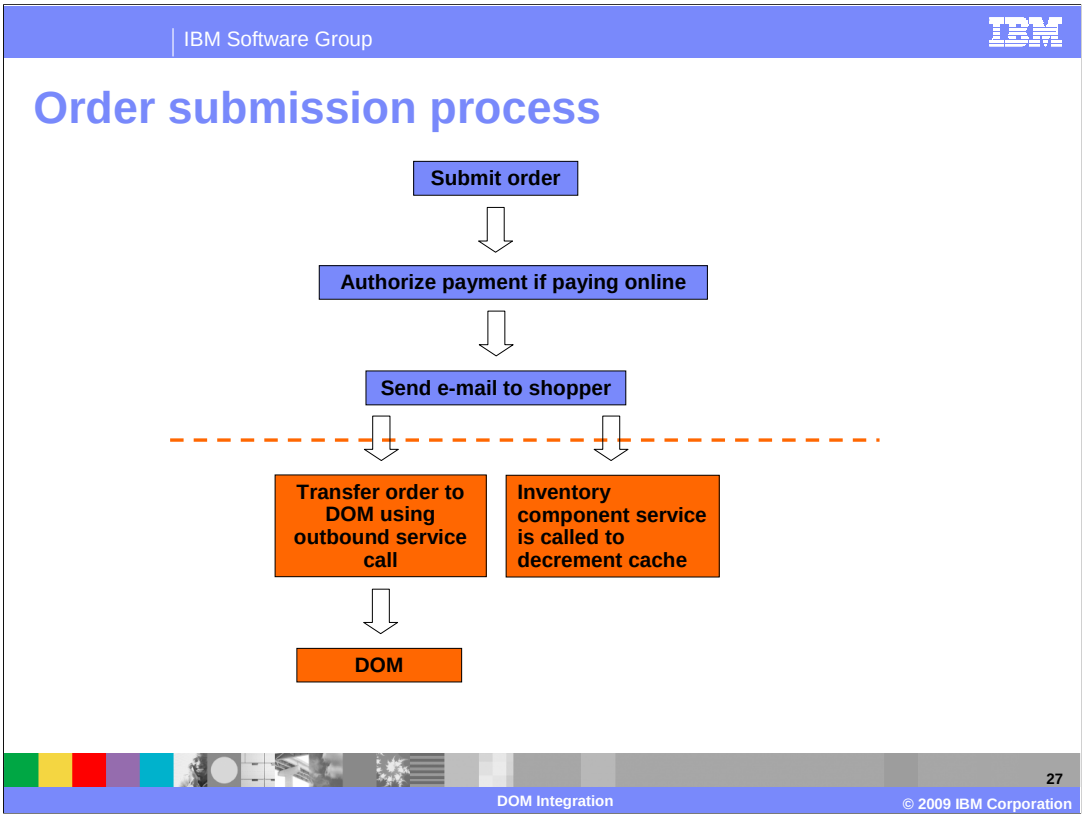


The order checkout process starts when a shopper clicks the check out link on a store page.

The online store uses an order component service to prepare the shopping cart for submission, this includes reserving inventory, evaluating promotions, and evaluating shipping and taxes.

If the store's inventory system is DOM, an Inventory component service is called. This service then calls an outbound service of the ExternalInventory component to send the reservation request to the DOM system. The returned information from the DOM system is used to update fulfillment options and inventory status of items in the order.

The resulting shopping cart is marked as locked. When all steps are finished, the online store redirects the shopper to the order summary page.



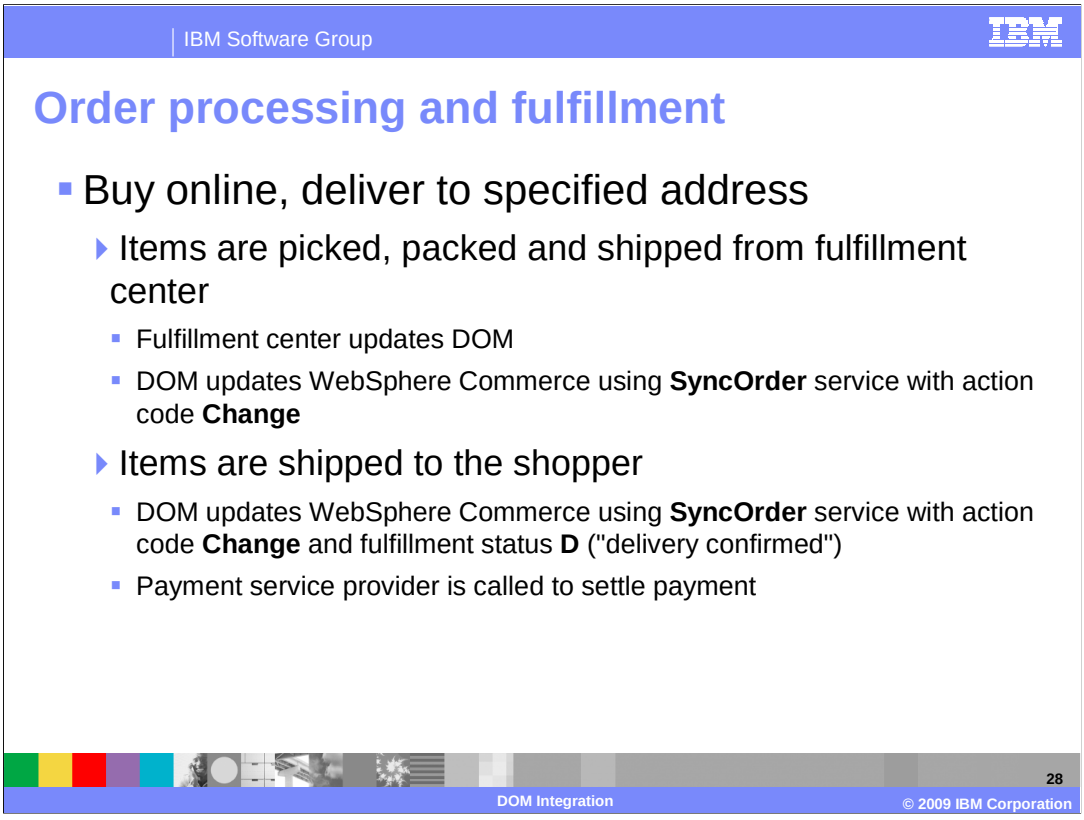
The order submission process is triggered when a shopper clicks the submit button after finalizing all order details.

The online store uses the Order component service or command to submit the order for processing.

The payment authorization command is called to authorize the payment if online payment is specified. It then updates the order status and sends an order confirmation e-mail to the shopper.

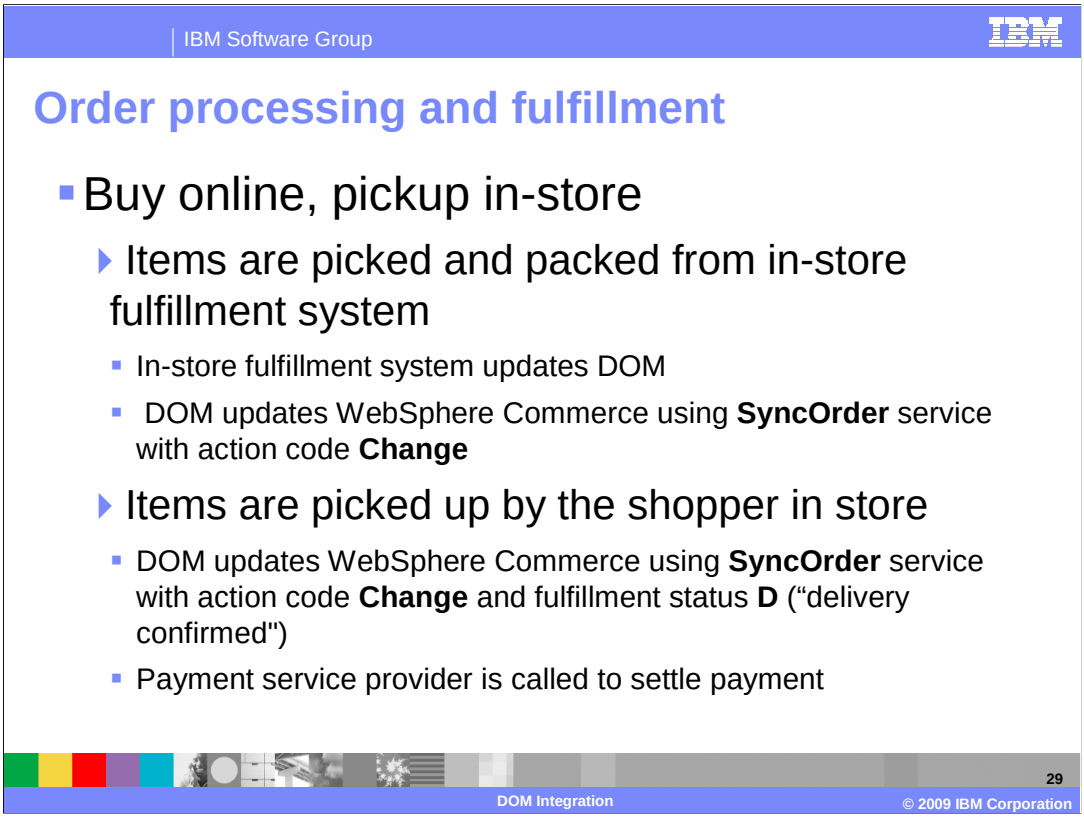
If the inventory system of the store is DOM, an outbound service is called to transfer the order to the DOM system for further processing. An Inventory component service is also called to decrement the cache.

When all steps have been finished, the online store redirects the shopper to the order confirmation page.



For a standard online order, once it is transferred to the DOM system the items are picked, packed and shipped from a fulfillment center. The fulfillment center updates the DOM system which propagates the updates (for example: order items shipped) to WebSphere Commerce by using the Order component's SyncOrder service with action code **Change**. The service updates the order and shipment records in WebSphere Commerce and sends a notification e-mail to the shopper about the update.

After the items are shipped to the shopper, the DOM system is updated and again propagates the updates to WebSphere Commerce by using the Order component's SyncOrder service with action code **Change** and fulfillment status D. The delivery confirmed status triggers the service to call the payment service provider to settle payment, update the order fulfillment records in WebSphere Commerce, and send a notification e-mail to the shopper about the update.



For the buy online, pickup in-store fulfillment option, the process is similar.

After the items are picked and packed from the in-store fulfillment system, the DOM system is updated. The DOM system propagates the update to WebSphere Commerce by using the Order component's SyncOrder service with action code **Change**. The service updates the order fulfillment records in WebSphere Commerce and sends a notification e-mail to shopper that the order is ready for pickup.

After the shopper picks up the items in-store, the DOM system is updated and again propagates the updates to WebSphere Commerce by using the Order component's SyncOrder service with action code **Change** and fulfillment status D. The delivery confirmed status triggers the service to call the payment service provider to settle payment, update the order fulfillment records in WebSphere Commerce, and send a notification e-mail to the shopper about the update.

Order processing and fulfillment

- Order online, pay and pickup in-store
  - Same as buy online, pickup in-store except payment authorization is done in the physical store

For the order online, pay and pickup in-store fulfillment option, the process is the same as buy online, pickup in-store except the payment authorization is done in the physical store.

The SyncOrder service is still used to update the order status in WebSphere Commerce.

Section

Integration points and methods

This section introduces DOM integration points and methods.

Service customization points

- Internal services

Component	BOD	Action Code / Access Profile	Description
Store	GetPhysicalStore	IBM_Store_Details	Finds physical stores by various means
	GetGeoNode	IBM_Store_All	Finds a node in a tree of geographical regions
Inventory	GetInventoryAvailability	IBM_Store_Details	Retrieves the inventory availability of an item at the locations specified.
	ProcessInventoryRequirement	CheckInventory	Checks the inventory availability of order items in an order or shopping cart.
		ReserveInventory	Reserves inventory for order items in an order or shopping cart.
		CancelInventoryReservation	Cancels the inventory reservations of order items in an order or shopping cart.
		DecrementCache	Decrements the cached inventory availability records by the requested quantities in an order or shopping cart.

This chart summarizes the new internal services for the DOM integration solution. As mentioned earlier, the store component provides two services. One service is for retrieving information about a physical store and the other is for accessing nodes in the geographical region tree.

The inventory component provides two services. GetInventoryAvailability makes use of the ExternalInventory component to contact the backend system. ProcessInventoryRequirement provides several action codes that support the inventory interactions that occur during the shopping process. These include checking if inventory is available, reserving inventory for items in an order or shopping cart, canceling reservations if they are no longer needed and decrementing the cache once inventory has been reserved.

Service integration points

- Outbound services

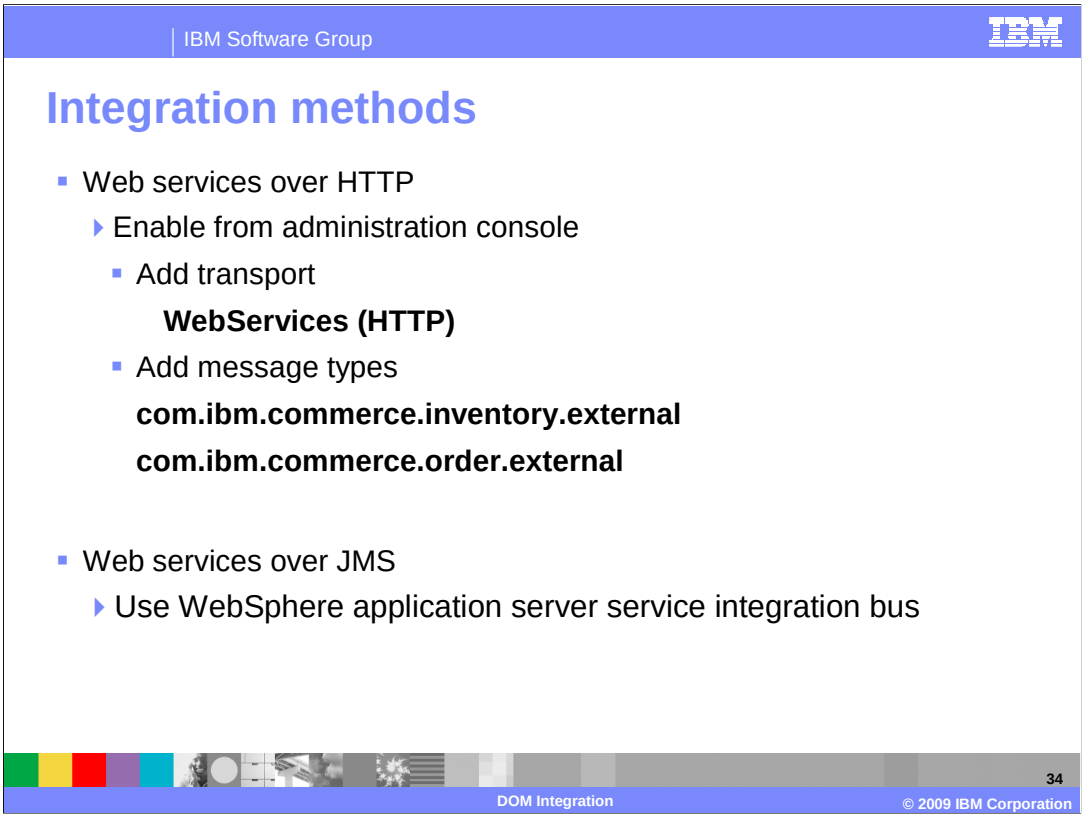
Component	BOD	Action Code / Access Profile	Description
ExternalInventory	GetInventoryAvailability	IBM_Store_Details	Retrieves the inventory availability of items at the locations specified.
	ProcessInventoryRequirement	ReserveInventory	Reserves inventory for order items in an order or shopping cart.
		CancelInventoryReservation	Cancels the inventory reservations of order items in an order or shopping cart.
ExternalOrder	ProcessOrder	Transfer	Transfers an order for further processing.

- Inbound services

Component	BOD	Action Code / Access Profile	Description
Inventory	SyncInventoryAvailability	Change	Updates the cached inventory availability records
Order	SyncOrder	Change	Updates the order fulfillment records
	ProcessOrder	Cancel	Cancels an order or shopping cart

Communication with a DOM system is provided through a series of inbound and outbound services shown here. Outbound services are provided by the ExternalInventory and ExternalOrder components which are used by the Inventory and Order components respectively. Outbound inventory services include requesting inventory information and reserving or canceling inventory for an order. The outbound order service transfers the online order to the DOM system for further processing.

Inbound service requests are handled by the Inventory and Order components. Supported inbound messages are SyncInventoryAvailability which updates the cached data, SyncOrder which updates the status of the order and ProcessOrder which can be used to cancel an order placed through the Web channel that cannot be fulfilled.



There are two methods you can use to integrate with a DOM system: Web services over HTTP and Web services over JMS.

Before you can use the Web services over HTTP method, you need to use administration console to add the WebServices(HTTP) transport. You also need to add two message types for external inventory and external order outbound services calls.

To integrate WebSphere Commerce with a DOM system over JMS, you need to use WebSphere Application Server Service integration bus. The bus is built on top of the Java™ Messaging Service (JMS) delivered in WebSphere Application Server. It is able to expose messaging artifacts such as queues and topics as Web services. For more information about this topic, refer to the WebSphere Application Server information center.

Section

Restrictions and limitations

This section covers restrictions and limitations related to DOM integration.

Restrictions and limitations

- Sales center does not support buy online, pickup in-store
- The **Store** component's **GetGeoNode** and **GetPhysicalStore** services do not have alternative language support
- Order items can not be split for partial delivery and partial pickup in-store
- If you choose to pickup in-store, you must select a single payment method

Buy online, pickup in-store is not supported by Sales Center for this release. If you want to use Sales Center to manage buy online, pickup in-store orders, customization is needed.

The Store component services GetGeoNode and GetPhysicalStore do not have alternative language support. For example, if a physical store does not have a description in the context language, the GetPhysicalStore service will not automatically retrieve its description in an alternative language.

As a shopper, you can either choose to have all order items delivered, or pickup all items in a store, you cannot split the order items for partial delivery and partial pickup in-store. You also cannot split order items for pickup at two or more stores.

If you choose to pickup items in-store during checkout, you can only use a single payment method. Payment can be online or at the store but it cannot be split.

Summary

- DOM integration
- New component services
- Restrictions and limitations

In summary, feature pack 5 has enabled WebSphere Commerce to integrate with external distributed order management systems. Several new component services have been created for this implementation. There are a few restrictions and limitations to be aware of in this release.

References

- WebSphere distributed object cache

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tdyn_distmap.html

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/udyn_cacheinstancescollection.html

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.javadoc.doc/public_html/api/com/ibm/websphere/cache/DistributedObjectCache.html

- Service integration technologies

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.pmc.doc/tasks/tji0000_.html

- SOA Logical Architecture and WebSphere Commerce back-office integration

http://publib.boulder.ibm.com/infocenter/wchelp/v6r0m0/index.jsp?topic=/com.ibm.commerce.developer.doc/concepts/csdsoasoi.htm

Here are some links to related resources.

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 e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WCS6005_DOMIntegration.ppt

This module is also available in PDF format at: ../WCS6005_DOMIntegration.pdf

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

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. 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

J2EE, Java, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.