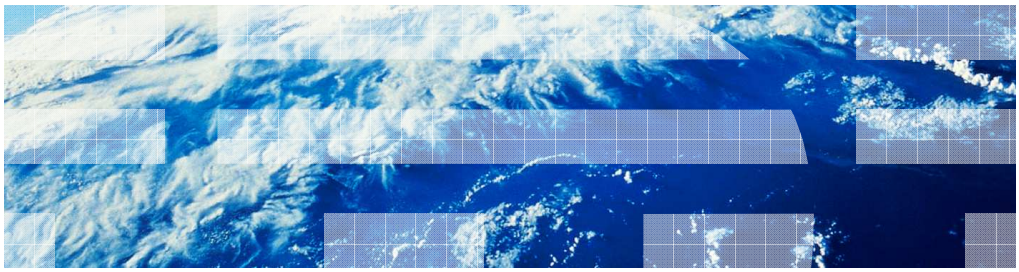
WebSphere Commerce V7 Feature Pack 3

Sterling Commerce DOM integration enhancement

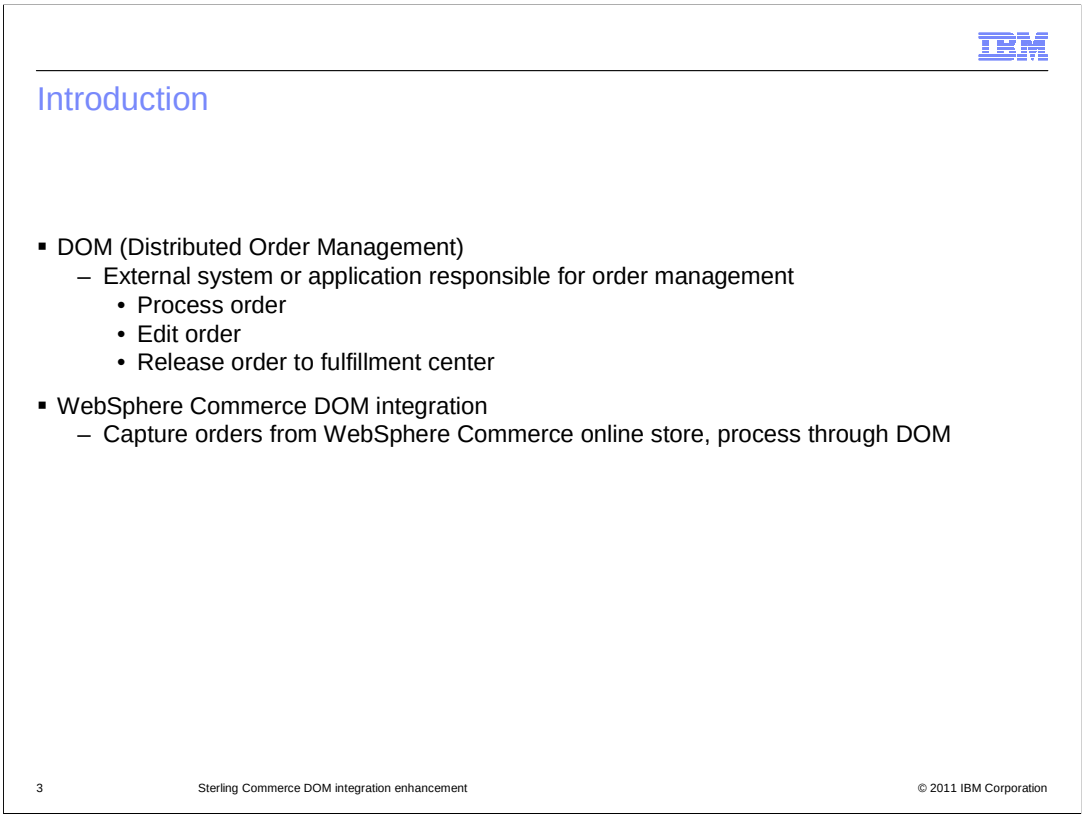


This presentation provides an overview of the Sterling Commerce® DOM Integration Enhancement to the WebSphere® Commerce V7.0 Feature Pack 3.

Table of content

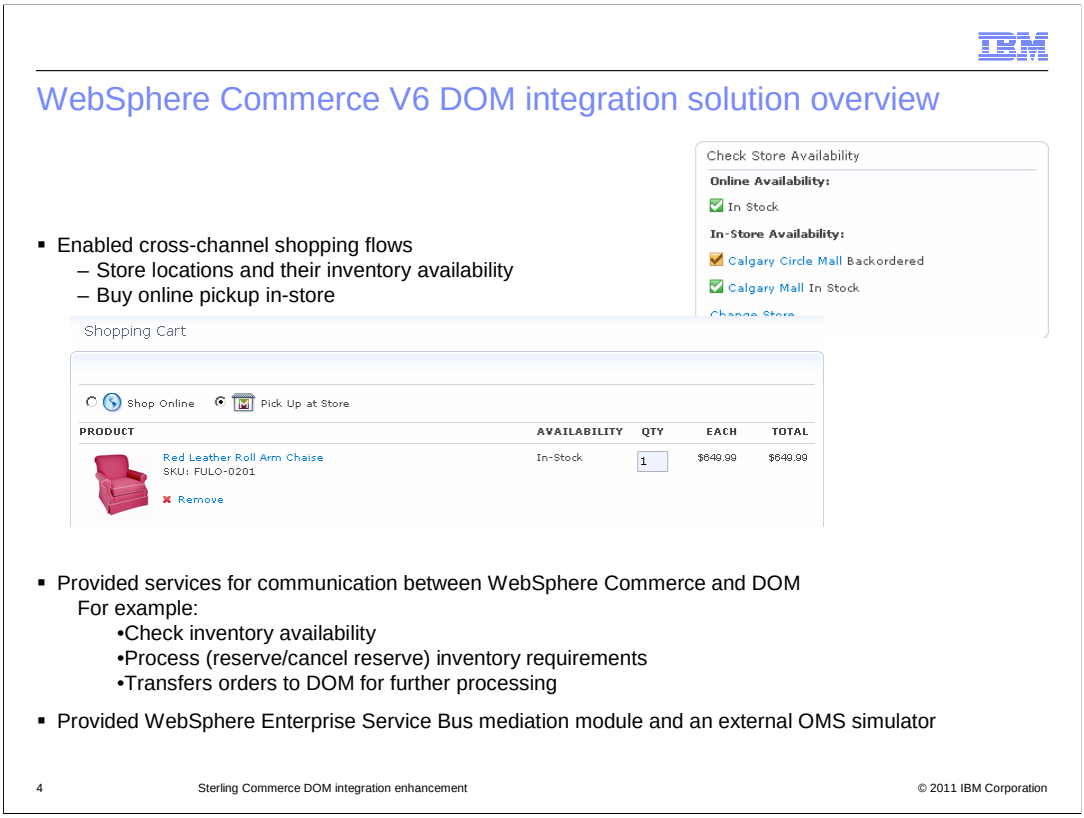
WebSphere Commerce started to support DOM integration in version 6 Feature Pack 5, an OMS simulator was provided as a DOM system so that you can test and learn this integration solution. In V7 Feature Pack 2, WebSphere Commerce implemented the integration with Sterling Commerce and Sterling Selling and Fulfillment Suite (SSFS), as the out-of-box solution for DOM integration. The Feature Pack 3 release enhances Feature Pack 2's solution with several scenarios that were not covered in Feature Pack 2.

In order for you to better understand Sterling DOM integration solution, this presentation starts with an overview of DOM integration in V6. It then highlights the Sterling DOM integration features of V7 Feature Pack 2, and concludes with an introduction of Sterling DOM enhancements in Feature Pack 3.



A distributed order management (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, editing and releasing the orders to the appropriate fulfillment system. In most cases, inventory is also managed by this external DOM system.

WebSphere Commerce DOM integration allows you to process your online orders captured by WebSphere Commerce through 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.



WebSphere Commerce version 6 DOM integration solution enabled cross-channel shopping flows which allowed shoppers to find physical stores according to the stores' location. Shoppers can add stores to their store list. During check-out, shoppers can select a physical store to pick up their order items.

A series of inbound and outbound services have been provided to handle the communication between WebSphere Commerce and a DOM system. The services include ones for checking the DOM's inventory availability, reserving inventory on the DOM, and transferring the orders to the DOM for further processing.

Communications between WebSphere Commerce and the DOM system is handled by a WebSphere Enterprise Service Bus mediation module. The message format that WebSphere Commerce provides or requires cannot be the same as the one that DOM requires or provides. This mediation module translates the BOD OAGIS messages that WebSphere Commerce uses to the message format required by the DOM system.

The solution provided in V6 is a generic solution, it is not for any specific DOM system. A sample WebSphere Enterprise Service Bus mediation module and a sample external order management system (OMS) is provided.

Outbound and inbound services for integration

Outbound services

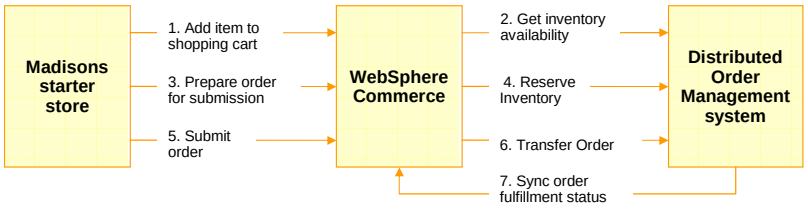
Component	BOD	Action Code / Access Profile	Description
ExternalInventory	GetInventoryAvailability	IBM_Store_Details	Retrieves the inventory availability of items at the locations specified.
	ProcessInventoryRequirement	ReserveInventory	Reserves inventory for order items in an order or shopping cart.
		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

The outbound and inbound services provided for the DOM integration in version 6 are listed here.

Service requests used in checkout flow



The checkout flow diagram shows how the DOM integration services are used in a DOM integrated environment. To make it simple, the mediation module is not added into the diagram.

In this diagram, when a shopper adds an item into the shopping cart, WebSphere Commerce Inventory component gets the inventory information either from cache or from DOM by making an outbound service call to the DOM. When the shopper initiates the checkout process, the Inventory component makes another service call to DOM to reserve inventory for order items. After the shopper submits the order, the order component makes a service call to transfer the order to DOM.

As DOM processes the order, DOM sends the order fulfillment status updates, for example, **order items shipped**, to WebSphere Commerce by using the order component's SyncOrder service.

Inventory caching

- Two caching options are implemented – Database cache and In memory cache
- Cache configuration is stored in these two tables
  - INVCNF
  - INVCNFREL
- Database cache
  - Inventory availability data is saved in table INVAVL
  - Massload utility can be used to batch load the data
- In memory cache
  - WebSphere distributed object cache

Inventory is the most often requested data from the DOM. The inventory availability is checked in many places such as product page and checkout page. Inventory caching has been implemented to make sure you get the best performance when you integrate your site with a DOM.

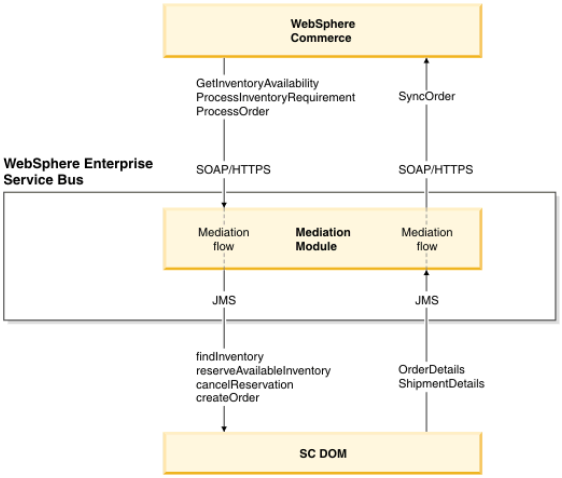
Two inventory cache options are provided: database cache and in memory cache. Cache configuration is stored in inventory configuration table (INVCNF) and inventory configuration relationship table (INVCNFREL). When you configure your inventory cache, you first need to define your cache policies in table INVCNF, such as what cache options you want to use and when the inventory data in the cache expires. Then you associate the inventory cache policy to catalog entries and store locations in table INVCNFREL.

If database cache is selected, the inventory availability data will go to table INVAVL. You can use Massload utility to batch load the inventory data into this table when you configure your DOM integration.

In memory cache is implemented by WebSphere distributed object cache. You can use memory cache strategy if the batch load is not required.

Sterling DOM integration for Feature Pack 2

- Feature Pack 2 supports Sterling Selling and Fulfillment Suite (SSFS) as DOM
- Integration is handled by WebSphere Enterprise Service Bus mediation module
  - WebSphere Enterprise Service Bus mediation module maps WC services to SSFS API calls
  - Sample mediation module project **WCToSSFSMediationModule** is provided:
    - <wc>/components/sterling-integration/
  - SOAP over HTTP(S) to WebSphere Commerce, JMS to SSFS



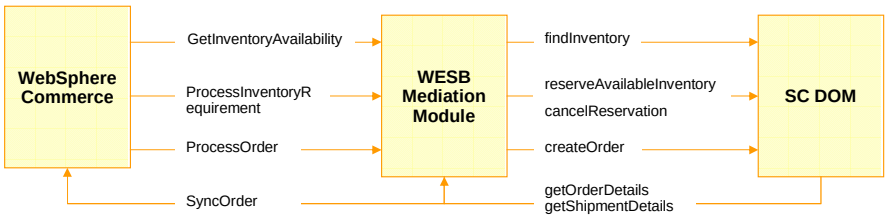
When the DOM integration solution was implemented in version 6, there was no specific DOM system specified. An OMS simulator was provided as DOM for you to test and learn this solution. WebSphere Commerce version 7 Feature Pack 2 integrated with Sterling Commerce and Sterling Selling and Fulfillment Suite (SSFS), as the out-of-box solution for DOM integration.

Integration with SSFS is accomplished through several inbound and outbound service calls that were developed during V6 Feature Pack 5. Integration between WebSphere Commerce and the SSFS is handled by a WebSphere Enterprise Service Bus mediation module. This mediation module translates WebSphere Commerce BOD OAGIS messages to SSFS API calls.

A sample mediation module project WCToSSFSMediationModule is provided. Several JSP files are also provided for the storefront to display the order status. You can use this code as a starting point when you integrate with the SSFS.

The Sterling DOM integration uses SOAP over HTTP(S) for the communication between mediation module and WebSphere Commerce, and uses JMS for the communication between WebSphere Commerce and SSFS.

Integration flows and services mapping



Communication	WebSphere Commerce request	SSFS API call
Get inventory availability	GetInventoryAvailability	findInventory
Reserve Inventory	ProcessInventoryRequirement	reserveAvailableInventory cancelReservation
Transfer order	ProcessOrder	createOrder
Sync order fulfillment status	SyncOrder	getOrderDetails getShipmentDetails

More information is provided on this slide to show your how the WebSphere Commerce requests are mapped to SSFS API calls in the mediation module.

Limitations and supported software for Feature Pack 2

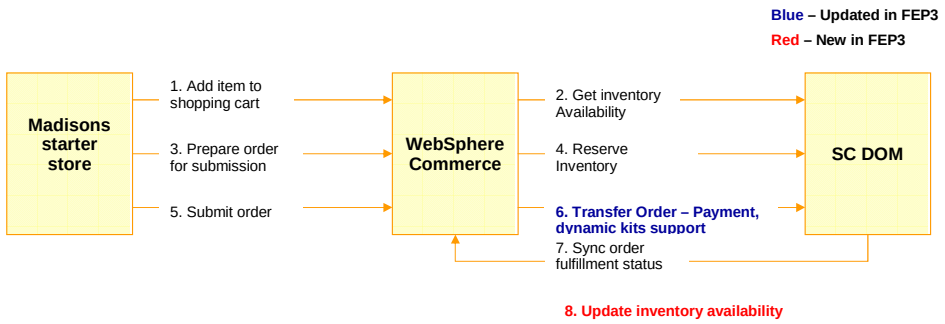
- Limitations
  - An order item can only be shipped to one ship node (fulfillment center)
  - WebSphere Commerce handling payments
  - Catalog does not contain bundles or kits
- Supported software
  - WebSphere Integration Developer version 7.0.0.3
  - WebSphere Enterprise Service Bus (WESB) version 7.0.0.0
  - Sterling Selling and Fulfillment Suite (SSFS) version **9.0**

This slide shows several limitations in Feature pack 2 Sterling DOM integration. Sterling Commerce supports an order item shipping to multiple fulfillment centers, while the Sterling DOM integration feature provided by Feature Pack 2 only supports an order item with a single fulfillment center. The Feature Pack 2 solution does not transfer payment information, such as credit card information, from WebSphere Commerce to SSFS. If you use the Feature Pack 2 solution, you must let WebSphere Commerce handle the payments. Feature Pack 2 also does not support catalog containing bundles or kits.

The supported software and the versions are listed here for Feature Pack 2.

Sterling Commerce DOM integration enhancements in Feature Pack 3

- Inventory cache push synchronization support using Sterling Commerce's Real-time Availability Monitor (RTAM)
- Payment integration support
- Feature Pack 3 supports SSFS V9.1 only
- Updated order status display page for the Madison's starter store



Feature Pack 3 enhances the Sterling DOM integration features by adding several additional scenarios not covered in V7 Feature Pack 2.

WebSphere Commerce inventory cache in the Feature Pack 2 solution is populated and refreshed through polling only. WebSphere Commerce initiates the inventory update request, the inventory data is retrieved from SSFS and the cache is updated on WebSphere Commerce. Feature Pack 3 provides an additional way for inventory cache synchronization using Sterling Commerce Real-time Availability Monitor (RTAM) – Inventory pushing. Inventory pushing allows Sterling to initiate inventory update instead of WebSphere Commerce. This feature reduces the polling and provides better control keeping the WebSphere Commerce inventory cache up-to-date.

WebSphere Commerce Order noun and order transfer logic in Feature Pack 2 does not support transferring payment-related information from WebSphere Commerce to Sterling DOM because the feature pack did not have the ability to tokenize the primary account number. Feature Pack 3 added payment integration support. You can transfer the payment information to Sterling DOM and let it handle the payment.

Inventory cache push synchronization support

- **Activity-based sync**: only push the updated inventory data to WebSphere Commerce
  - Inbound service **SyncInventoryAvailability** is used to update WebSphere Commerce inventory cache

Component	BOD	Action Code / Access Profile	Description
Inventory	SyncInventoryAvailability	Change	Updates the cached inventory availability records

- **Full sync:** push the whole shipping node (fulfillment center) inventory availability to WebSphere Commerce
  - WebSphere Enterprise Service Bus mediation module save the inventory into a CSV file. Upload CSV to WebSphere Commerce using the Massload utility

Two inventory cache push synchronization scenarios are covered for the Feature Pack 3 release: Activity-based synchronization and Full synchronization.

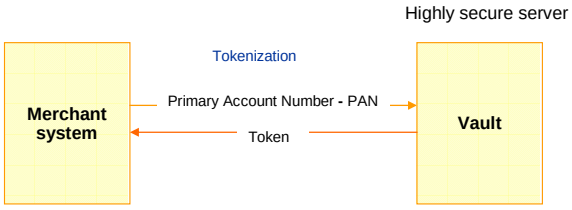
When the Activity-based synchronization takes place, only updated inventory data is pushed to the WebSphere Commerce. The Inventory cache push request invokes the WebSphere Commerce inbound service SyncInventoryAvailability to update WebSphere Commerce inventory cache.

The **Full synchronization** updates the whole shipping node's inventory. In order to get better performance, instead of pushing the inventory data directly to the WebSphere Commerce inventory cache, a CSV file is generated from the WebSphere Enterprise Service Bus mediation module. You can load the CSV file to WebSphere Commerce using the Massload utility.

For both scenarios, synchronization can be scheduled at specific time interval or triggered manually from the SSFS.

Payment integration support

- Sterling DOM requires PAN to be tokenized
- Payment tokenization



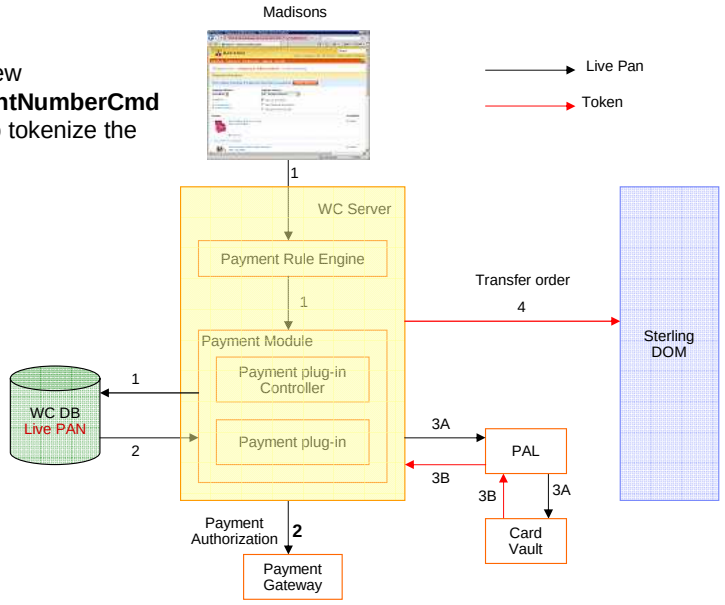
- Customization point to the WebSphere Commerce order transfer logic to allow you to supply your PAN tokenization logic
- **Order** noun and order transfer logic have been extended to support payment information passing

Tokenization and encryption are different ways to secure sensitive data. Payment tokenization is the process of replacing sensitive data, such as a credit card number, with a unique token. The sensitive data is stored in a centralized and highly secure server named Vault. The sensitive data can be retrieved by sending the token back to the external vault.

Sterling DOM requires the primary account number (PAN) to be tokenized before passing it to Sterling DOM. Feature Pack 3 has added a customization point to the WebSphere Commerce order transfer logic to allow you to implement your own PAN tokenization logic. To support the payment integration, WebSphere Commerce Order noun and order transfer logic are also extended to support the payment-related information transferring.

Payment integration customization point

- Implement the new **TokenizeAccountNumberCmd** task command to tokenize the PAN



To enable payment integration support, you need to implement the new **TokenizeAccountNumberCmd** task command to call the PAL (Sterling Payment Abstraction Layer) to tokenize the primary account number. The token is then composed into an Order noun before transferring the order from WebSphere Commerce to Sterling DOM.

For this payment integration, you have to be aware of several things. First, WebSphere Commerce leaves its current behavior on PAN handling unchanged. Live PAN still goes through WebSphere Commerce application and is stored in the WebSphere Commerce database with encrypted format. Secondly, when an authorization request is raised in WebSphere Commerce, the live PAN is retrieved from the database and sent to the Payment Gateway to process the payment request. Finally, TokenizeAccountNumberCmd task command is only called just before transferring the order to Sterling DOM. The token is sent to Sterling DOM along with the order information.

Reference

- **Distributed Order Management (DOM) integration**

http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.dom-integration.doc/concepts/csmdomintro.htm

- **Distributed Order Management (DOM) integration flows**

http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.dom-integration.doc/concepts/csmdomintegrationflow.htm

This slide contains some useful references for understanding the DOM Integration solution.

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_DOMIntegrationEnhancement.ppt

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