Integration Guide for WebSphere Commerce with SAP R/3 using WebSphere MQ Integrator

Version 5.5

Before using this information and the product it supports, read the information in “Notices”.

First Edition (August 2003)

This edition applies to IBM WebSphere Commerce Version 5.5 and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure that you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office that serves your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. You can send your comments by using the online IBM WebSphere Commerce documentation feedback form, available at the following URL:

http://www.ibm.com/software/commerce/rcf.html

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Contents .... **iii**
Before you begin .... **v**
Conventions used in this guide vi
Chapter 1. Introduction **9**
Conventions and terminology used in this book 9
Overview 9
References 12
Chapter 2. Prerequisites **15**
Chapter 3. Sample topology **17**
Chapter 4. Message flow **19**
Message flow from and to WebSphere Commerce 20
Chapter 5. Organization and process mapping **21**
Assumptions 23
Chapter 6. Installing and configuring **25**
Installing and configuring WebSphere Commerce 25
Installing and configuring WebSphere MQ 25
Configuring the WebSphere Commerce messaging system 27
Installing and configuring WMQI Version 2.1 27
IDoc parser 29
Literals 29
Installing and configuring MQSeries link for R/3 31
Publishing the sample store 31
WebSphere Commerce program adapter security 33
Modifying WMQI ESQLs for credentials 33
Configuring the SAP R/3 system 34
Chapter 7. Loading master data **37**
Loading SAP materials and customer data onto WebSphere Commerce 37
Loading reference data onto SAP 44
Store customizations 47
Chapter 8. Verification procedure **49**
Running the servers 49
Testing individual messages 50
Verifying the master data upload 53
Chapter 9. Adding new messages to WebSphere Commerce **55**
Inbound message into WebSphere Commerce 55
Outbound message from WebSphere Commerce 56
Appendix A. WMQI message sets and message flows **57**
WMQI message sets 57
WMQI message flows 57
Appendix B. Mapping information **59**
Customer Create/Update Message (DEBMAS05) 60
Order Confirmation Status Message (SISCSO01) 62
Order Delivery Status Message (SISDEL01) 66

Order Invoice Status Messages (SISINV01) 70
Order Create Message (ORDERS05) 74
Product Price Update Message (COND_A02) 95
Product Quantity Update Message (INVCON01) 97
Notices 99

Before you begin

The *Integration Guide for WebSphere® Commerce with SAP® R/3® using WebSphere MQ Integrator* is intended for those who want to integrate WebSphere Commerce version 5.5 to a backend system using WebSphere MQ Integrator (WMQI). This guide describes how IBM® WebSphere Commerce version 5.5 can be integrated in particular with the SAP R/3 4.6 Enterprise Resource Planning (ERP) system using WMQI. The approach described for integrating to SAP R/3 can be used for other backend systems using the WMQI broker. This guide will assist developers and engagement teams developing an integrated solution for any backend system like SAP R/3. Additionally, demonstrators or marketing personnel who want to demonstrate the integration functionality of WebSphere Commerce and WMQI can use this guide.

Note: Knowledge of WebSphere Commerce Business Edition 5.5, SAP R/3 Enterprise, and the WMQI is assumed.

In place of WebSphere Commerce 5.5 Business Edition you can also use WebSphere Commerce 5.5 Professional Edition. This document provides information on how asynchronous and synchronous messages can be exchanged between WebSphere Commerce and SAP through WMQI. It gives an overview of a typical end-to-end flow between the two systems, installation and configuration of various components, and pointers to related sources of information.

The samples provided with this reference application are for WebSphere Commerce 5.5, Business Edition. However, the same principal and pattern can apply to WebSphere Commerce 5.5, Professional Edition too.

This guide is divided into the following sections:

Chapter 1. Introduction

A brief overview of the integration of WebSphere Commerce Version 5.5, Business Edition with SAP R/3 using the WMQI as well as the definition of the terms used in this book, and references to other related documents.

Chapter 2. Prerequisites

Lists the software and hardware prerequisites for this reference application.

Chapter 3. Sample topology

A description of the sample topology used to integrate WebSphere Commerce with SAP using WMQI.

Chapter 4. Message flow

Describes how WebSphere Commerce and SAP interact through messages.

Chapter 5. Organization and process mapping

Describes how the entities and organization hierarchies in WebSphere Commerce and SAP R/3 are mapped.

Chapter 6. Installing and configuring

Contains installation and configuration instructions for the components of this reference application.

Chapter 7. Loading master data

Describes how to synchronize product data between WebSphere Commerce and SAP.

Chapter 8. Verification procedure

A list of instructions to verify the synchronous and asynchronous message flows.

Chapter 9. Adding new messages to WebSphere Commerce

Describes how to add new messages to WebSphere Commerce.

Appendix A. WMQI message sets and message flows

An overview of WMQI message sets and message flows.

Appendix B. Mapping information

Includes the DTDs used for messages supported by this reference application, and the necessary mapping information for these messages.

Conventions used in this guide

This guide uses the following conventions:

Boldface type indicates graphical user interface (GUI) controls such as names of fields, buttons, or menu choices.

`Monospaced type` indicates examples of text you enter exactly as shown.

Italic type is used for emphasis and variables for which you substitute your own values.

Default paths

This guide uses the following default installation paths:

WC_installdir This indicates the installation path for WebSphere Commerce. When you see this variable, substitute the installation path for your installation of WebSphere Commerce. For Windows®, substitute

C:\IBM\WebSphere\CommerceServer55

WAS_installdir This indicates the installation path for WebSphere Application Server. When you see this variable, substitute the installation path for your installation of WebSphere Application Server. For Windows, substitute

C:\IBM\WebSphere\AppServer

MQlink_installdir This indicates the installation path for MQSeries® link for R/3. When you see this variable, substitute the installation path for your installation of WebSphere Application Server. For Windows, substitute

C:\smq

Chapter 1. Introduction

This chapter gives an overview of the integration of WebSphere Commerce Version 5.5, Business Edition with SAP R/3 using WMQI as well as the definition of the terms used in this book, and references to other related documents.

Conventions and terminology used in this book

Listed below are the terms and their definitions used in this book:

Intermediate Documents (IDocs)

Intermediate Documents (IDocs) are used to exchange data between R/3, and non-SAP systems. It is the document format that the SAP R/3 system understands.

ESQL

Extended SQL (ESQL) is a language used to access and update database data. It is derived from SQL version 3 and is particularly suited to manipulating both database and message data. ESQL has a whole range of data types and each has its own way of writing literal values. ESQL also has a set of operators, pre-defined functions, statements, and nested statements.

WebSphere MQ Integrator

WebSphere MQ Integrator (WMQI, formerly known as MQSeries Integrator or MQSI) is the message broker used to extend the basic functionality of WebSphere MQSeries by providing transformation and routing capabilities.

Reference data

Reference data is a sample set of products and items included in the reference application that can be used for demonstrations. It is included as part of the sample store, and also as a delimited file that can be loaded onto the SAP R/3 system.

Note: In this document the following are used interchangeably:

- WebSphere Commerce and WebSphere Commerce Business Edition
- WMQI and WebSphere MQ Integrator
- SAP or R/3 and SAP R/3 Enterprise System

Overview

This reference application is designed to integrate SAP R/3 with WebSphere Commerce, providing the e-commerce functionality of WebSphere Commerce as the front-end, along with the ERP functionality of SAP R/3 as the supplier and fulfillment center. Typically, this reference application is an example of how to utilize WMQI to integrate WebSphere Commerce and other applications. The patterns and principals demonstrated in these examples for SAP R/3 can be applied to other systems.

In the scenario described below, the WebSphere Commerce server acts as an e-commerce front-end to the SAP R/3 system.

Business scenario - Integration with the seller's backend system

This reference application uses broker-based integration where information is exchanged between the WebSphere Commerce and SAP R/3 systems asynchronously.

In this scenario, the synchronization of data between WebSphere Commerce and SAP ensures that all the necessary information about inventory, prices, materials, and so on is present in both systems. This enables buyers to create orders in WebSphere Commerce, and SAP to process the buyer’s orders and other requests.

The customer, material, inventory, pricing, order status, and other information will be uploaded onto WebSphere Commerce from the SAP system on a regular basis. The nature of transactions in this scenario allows WebSphere Commerce to be notified of changes in the SAP data, and alternatively, SAP to be notified of the orders created in WebSphere Commerce.

WebSphere Commerce further extends the scope of enterprise applications by providing a reliable, scalable, and open-standards based e-commerce front-end. For example, an organization using SAP R/3 for enterprise functions, when integrated with WebSphere Commerce, can add the Internet as a new front-end sales channel for its products and services.

Business models enabled

This integration provides e-commerce functionality by adding an Internet sales channel to the SAP system. Any customer registered with SAP can browse and view products that are loaded from SAP catalogs onto the WebSphere Commerce site. From here, buyers can place orders and query for the status of their orders and other relevant information that is present in the SAP system. This involves the synchronization of material data, price, and inventory information between the two systems. This synchronization is possible by the initial upload of material data from SAP to WebSphere Commerce. Connectivity in the current implementation enables customer data, material data, and order statuses from SAP to be updated in WebSphere Commerce through a set of messages.

Features

- **Order creation:** Buyers can create orders in WebSphere Commerce and the details of the order are sent to SAP in IDoc format for further processing.

- **Order status:** WebSphere Commerce can query and retrieve order status information on behalf of the buyers. Whenever there is a change in the status of an order, a message conveying the same can be triggered from SAP and sent to WebSphere Commerce. The three order statuses supported are:
    - Order Confirmation

  - Order Delivery
  - Order Invoice

- **Customer creation**: When new customers are created in SAP they can be registered in WebSphere Commerce by sending the CustomerCreate message from SAP to WebSphere Commerce.

- **Customer update:** When existing customer information is updated in SAP the changes can be sent to WebSphere Commerce using the CustomerUpdate message.

- **Product price update:** Changes in product prices in SAP can be communicated to WebSphere Commerce using the ProductPriceUpdate message.

- **Product inventory update:** Changes in product inventory in SAP can be sent to WebSphere Commerce using the ProductInventoryUpdate message.

- **Load materials from SAP:** You can load material information from SAP onto WebSphere Commerce using the Loader Package utility provided by WebSphere Commerce. You can also update WebSphere Commerce with the changes in catalog prices, which is the list price in SAP.

Benefits

- Create channel specific business processes in WebSphere Commerce and change them rapidly without having to change the backend system.

- Easily and quickly add a new Internet sales channel to the enterprise backend system.

- Develop Web channel specific processes that augment the backend process in a flexible manner, adding and collecting data specific for the Web channel without changes in the backend system. This allows more dynamic changes in the Web channel process without any impacts on the backend system. Additional product information such as long or short descriptions, which is not included in SAP, can be added in WebSphere Commerce. This information may be required for specific uses in a Web channel.

- Use WebSphere Commerce as a Web channel to enable customers to integrate SAP and other backend systems providing a single Web channel integrated business process.

- Provide customers with access to Web site functions such as browsing catalogs, placing orders, and making online payments.

- Create online catalogs in WebSphere Commerce from SAP materials.

- Provide multicultural support and personalized content based on profile, history, demographics, and other factors.

- Synchronize product and customer information between the front end and back end systems.

- Create orders in WebSphere Commerce and send the orders to SAP for order processing and fulfillment. Check the status of the order with SAP from the WebSphere Commerce site

- Leverage the complex business processes supported by the SAP enterprise system.

- Create commerce sites leveraging WebSphere Commerce product management, customer management, personalization, and merchandising capabilities.

- Manage complex relationships between buyers and sellers, organizations and sub organizations, which helps strengthen relationships with customers, partners, and suppliers.

- Provide highly scalable commerce storefront capabilities utilizing information from a SAP backend system while limiting the load on the SAP system.

- Provide operational and business analytics based on site statistics, usage scenarios, campaign effectiveness, demographics, and other factors.

- The broker-based integration maximizes the isolation of business processes from the external organization, the flexibility to change the processes, and the applications that implement them.

- Enables customers to create Commerce sites leveraging WebSphere Commerce product management, customer management, personalization, and merchandizing capabilities

WMQI message broker

A message broker is built on a queue manager and routes messages to applications. A message broker can provide real-time, rules-based message routing and dynamic message-content transformation and formatting. At run time, the message broker allows multiple applications to implement a published service with the broker providing application integration.

A message broker acts as a hub for messages passing between MQ applications. Once the message broker receives the message, it can be processed depending on the contents of the message and how the message broker is configured. The individual functions within the message broker are assigned to a collection of interconnected nodes (message flow) where the processing and transformation activities can take place as required.

References

Apart from this guide, the following reference documents are available with their respective products:

- WebSphere Commerce Business Edition messaging system.
  Information can be found in the product documentation
  http://www.ibm.com/software/webservers/commerce/wc_be/

- SAP R/3 documentation
  http://help.sap.com

- WebSphere MQSeries documentation
  http://www.ibm.com/software/integration/wmq/

- MQSeries link for R/3 documentation
  http://www.ibm.com/software/integration/mqfamily/

- WebSphere MQ Integrator
  http://www.ibm.com/software/integration/mqfamily/integrator/broker/

Chapter 2. Prerequisites

This section covers the software components used in this reference application.

WebSphere Commerce 5.5, Business Edition

WebSphere Commerce 5.5, Business Edition is an e-commerce software that has various subsystems. The messaging system gives WebSphere Commerce the ability to communicate with an external environment. This communication includes sending and receiving messages to and from back-end systems. This is achieved through the following components:

- A listener for WebSphere MQ to process inbound requests.
- An adapter for WebSphere MQ for outbound requests to allow you to integrate with back-end and external systems.

Note: This reference application uses a DB2® database server. The instructions provided in Chapter 6. Installing and configuring assume that a DB2 server is used.

WebSphere MQ 5.3

WebSphere MQ is used as the transport middleware to communicate with various external systems, including WebSphere MQ Integrator. Refer to the IBM WebSphere Commerce *Additional Software Guide Version 5.5* for information on setting up WebSphere MQ for WebSphere Commerce.

WebSphere MQ Integrator (WMQI) Version 2.1

WMQI is the transformation engine. The various components required in WMQI are message sets, messages, message flows, assignments, and broker. Using parsers like XML, IDoc, and ESQLs, an input message is parsed and transformed, or reformatted into an output message in the required format.

The IDoc parser is a WMQI parser for representing incoming data to WMQI in a format that the WMQI compute node can manipulate. This parser plugin is a prerequisite to process the SAP IDoc messages.

WMQI allows data mapping for different formats into IDoc streams. Apply CSD3 (Corrective Service Diskette) to WMQI 2.1. For more information on CSD3 refer to:
http://www-3.ibm.com/software/integration/integratorbroker/

MQSeries link for R/3 Adapter 1.2

This adapter provides the WebSphere MQ messaging functionality to the SAP system. It consists of two servers, inbound and outbound. The inbound server receives IDocs from WMQI and sends them to the SAP system. The outbound server receives IDocs from the SAP system and sends them to

WMQI.

SAP R/3 4.6

This is an ERP system that contains the master data. WebSphere Commerce Business Edition provides the e-commerce functionality. The SAP system interacts with external applications by exchanging information in the form of messages. It generates IDocs that external applications will use and accepts IDocs from other applications that require processing by the R/3 system.

Chapter 3. Sample topology

Before you begin your installation you must decide on the hardware topology for this reference application. The following is the sample topology used for this integration:

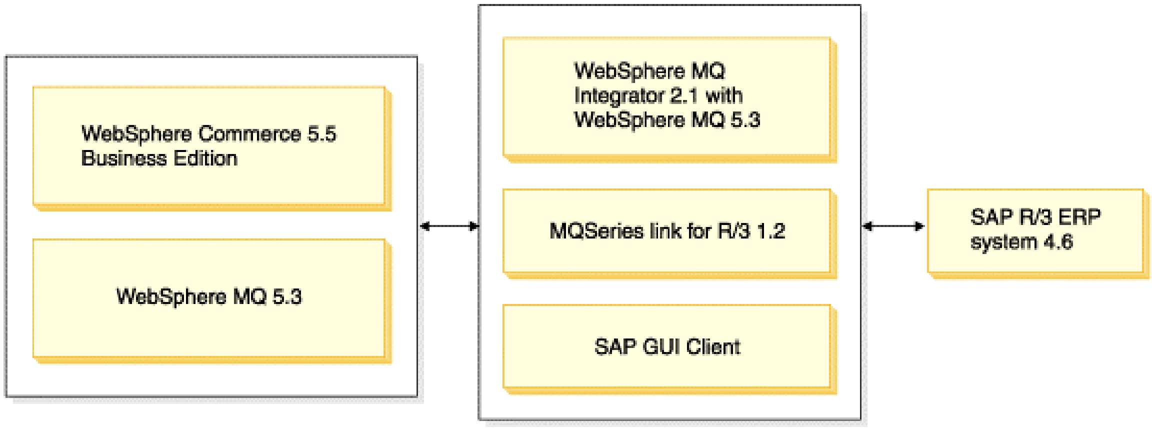

WebSphere Commerce 5.5 Business Edition
WebSphere MQ 5.3
WebSphere MQ Integrator 2.1 with WebSphere MQ 5.3
MQSeries link for R/3 1.2
SAP GUI Client
SAP R/3 ERP system 4.6

Figure 1: Sample topology

In this sample topology, WebSphere Commerce 5.5 Business Edition and WebSphere MQ 5.3 are installed on a Windows 2000 system. WebSphere MQ Integrator 2.1 with WebSphere MQ 5.3 and MQSeries link for R/3 1.2 are installed on a Windows NT® system.

Apart from the settings in Figure 1, the following alternate settings are also recommended*:*

- The MQSeries link for R/3 can be installed on the SAP server.
- The SAP GUI (graphical user interface) client is not a necessary component in the system.
- WebSphere Commerce Business Edition, WebSphere MQ, and WMQI can exist on different machines.

Chapter 4. Message flow

WebSphere Commerce and SAP interact through messages. These messages are passed through the messaging subsystem provided by IBM WebSphere MQ. Since message formats are different for WebSphere Commerce and SAP, WMQI translates messages before they are processed by other applications.

This reference application supports various messages that enable integration with SAP. The outbound and inbound messages supported and its flows from and to WebSphere Commerce Business Edition are described in detail.

Order create message (outbound from WebSphere Commerce)

WebSphere Commerce Business Edition generates this message when an order is submitted in the commerce server.

Order status message (inbound to WebSphere Commerce)

This is generated by the SAP system. These messages are of three types:

- Order confirm status: Generated when orders are confirmed by SAP.
- Order delivery status: Generated when delivery for the order is complete at the SAP end.
- Order invoice status: Generated when the order is invoiced in SAP.

Customer new message (inbound to WebSphere Commerce)

SAP generates this message when a new customer is registered in SAP.

Customer update message (inbound to WebSphere Commerce)

SAP generates this message when an existing customer's information is updated in SAP.

Product price update message (inbound to WebSphere Commerce)

SAP generates this message when the product price is updated in SAP.

Product quantity update message (inbound to WebSphere Commerce)

SAP generates this message when the product quantity is updated in SAP. This could occur when the inventory is reduced at the time of goods issue for an order, or updated manually

You can customize the message flows used for parsing messages to suit your circumstances. For the default mapping between WebSphere Commerce and IDoc fields, see Appendix B. Mapping information. If you modify the mapping of fields to suit your requirements, then change the ESQL (Extended SQL) code for the modified mappings.

Message flow from and to WebSphere Commerce

The following figure illustrates the message flow from and to WebSphere Commerce. It depicts how a message passes through various components and describes their roles.

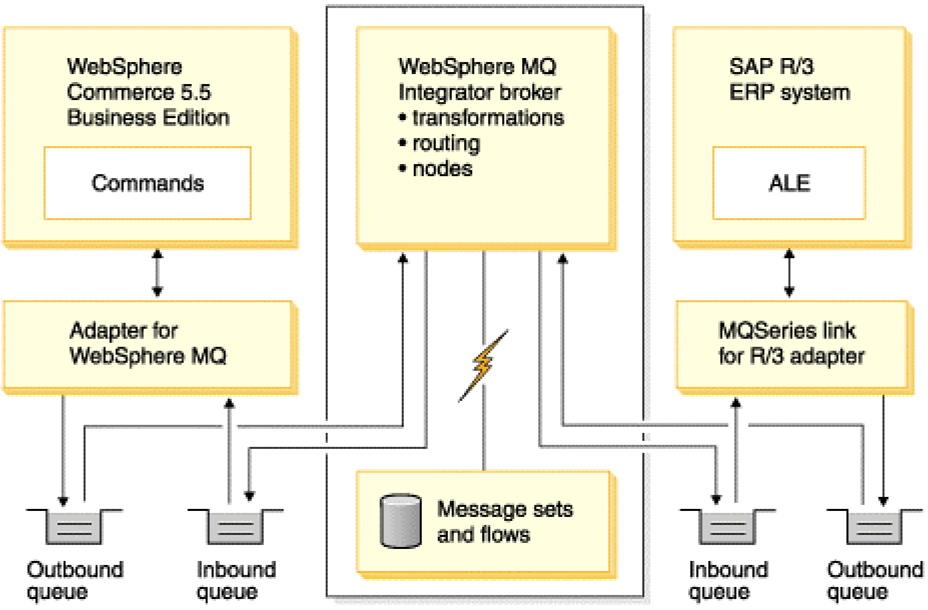


Figure 2: Message flow from and to WebSphere Commerce

Outbound message flow from WebSphere Commerce

The message is generated in WebSphere Commerce, which is configured to place the XML message in the WebSphere Commerce output queue. WMQI reads the messages from this queue and processes it according to the message flow and its nodes. The input node determines which parser will process the incoming message. The ESQLs in compute node are used to transform the input message into IDoc message format. The output node places this information in the queue configured to put the transformed message. The MQSeries link for R/3 inbound server reads the message to be sent to the SAP R/3 system. The reverse applies to the messages coming from SAP.

Inbound message flow to WebSphere Commerce

When an IDoc is generated at the SAP R/3 system, the message is sent to the outbound server of the R/3 link adapter that is configured with the SAP R/3 application name and host. The outbound server receives the message from the SAP R/3 system and puts it in the R/3 link output queue associated with the outbound server. WMQI or the message broker reads the message, parses and translates it by applying the appropriate ESQLs. The reformatted message after transformations from WMQI is an XML message to be used by a specific WebSphere Commerce command, for example OrderStatusUpdate in WebSphere Commerce. The output node in WMQI decides on the destination queue, which is the WebSphere Commerce input queue where the message must be put. The WebSphere Commerce server reads the message from this queue and invokes the corresponding command for further processing.

Chapter 5. Organization and process mapping

This section describes how the entities and organization hierarchies in WebSphere Commerce and SAP R/3 are mapped.

Customer

A customer that has a sold-to party partner function registered in SAP/R3, to whom products or services have been sold, is registered as a customer in WebSphere Commerce.

Item

An item in WebSphere Commerce is a product with defined values for its attributes. This is related to material in SAP R/3.

Product

A product in WebSphere Commerce represents an item with attributes. This is represented as a *Material Group* in SAP R/3. A material in SAP R/3 can be associated to a *Class* in SAP, which can have *Characteristics* to define the material. The WebSphere Commerce product attributes can be expressed as characteristics in SAP.

Store

A WebSphere Commerce store is mapped to a SAP *Sales Area*. The sales area is composed of the sales organization, the division, and the distribution channel. A company can have one or more sales areas and plants. A customer is registered and orders are created at the sales-area level. Customer information also contains the default plant where material should be picked up from to fill the customer’s orders. For this reference application, in SAP, a sales area and plant are dedicated to web sales. Materials for web sales must be associated with the dedicated plant. Web customers must be associated with a dedicated sales area. This may require extending the existing customers and materials to dedicated sales areas and plants respectively.

Fulfillment center

In SAP R/3 the inventory is kept in *plants and storage locations*. A plant contains storage locations.

SAP R/3 must have a dedicated plant and storage location for web sales. You can map this combination of plant and storage location to the default fulfillment center in WebSphere Commerce. In SAP R/3 the movement of goods is at the plant level and is not related to the sales organization.

Shipping carrier

This reference application uses one shipping carrier. The default shipping carrier information in the WMQISAPToolTech sample store archive is used for shipping. You must define the corresponding incoterms and conditions in SAP. Ensure that the three-character abbreviation for the carrier is stored in WebSphere Commerce. This abbreviation must be unique to the carriers, as SAP R/3 uses it as Incoterms1.

If new shipping carriers are introduced, decide on the unique three-character for them and create the corresponding incoterms in the SAP R/3 system.

In SAP R/3, shipping conditions are used to determine shipping costs. Shipping conditions are maintained based on the region of delivering plant, region and Incoterms 1 and Incoterms 2 and material weight.

Map the shipping carrier and shipping code in WebSphere Commerce with the Incoterms 1 and 2 in SAP respectively.

Price

In SAP R/3, pricing information can be at the customer, customer group level or, even at the material level based on the pricing conditions defined**.** In WebSphere Commerce, an item can have a default offer and multiple offers. A TradingPositionContainer contains the offers and is associated with the member groups. The material prices extracted from SAP are mapped to the default offer and default trading position container in WebSphere commerce. SAP allows you to create additional pricing conditions, for example, customer group or price list based pricing. Map these additional pricing conditions to an associated TradingPositionContainer in WebSphere Commerce.

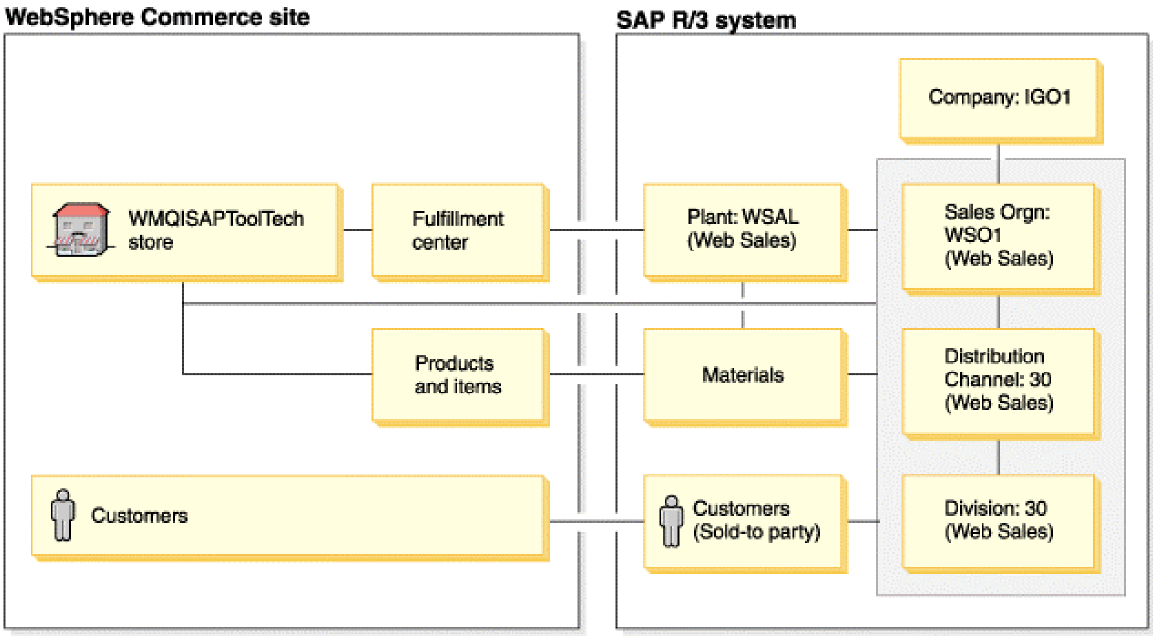


Figure 3: Organization mapping

Note: This mapping is easily understood in a single-supplier scenario. It also implies the following:

- Pricing information in SAP is based on the sales area. In WebSphere Commerce, the list price and offer price are independent of the store. Consequently, to exchange prices the store should represent the member. This implies that a member will have one store for the purposes of pricing.

- Inventory information is at the plant level in SAP, whereas, in WebSphere Commerce it is at the fulfillment center level. The command for updating the inventory in WebSphere Commerce requires the store identifier. This restricts one fulfillment center to a store, so that the mapping is one-to-one.

Assumptions

When developing the maps in WMQI, the following assumptions are made between the data models of WebSphere Commerce and SAP:

- The Password Expired is set to 1. This is a default value provided by the WMQI ESQLs during reformatting, for the customer create message.

- The Address Type is set to "SB" (shipto, billto) in ESQL during transformation for the customer create and update messages.

- The Profile Type is set to "B" (customer) in ESQL during transformation for customer create and update messages.

- Preferred Language values (DEBMAS05) in WMQI are mapped based on the values provided in the SAP server.

- In the inventory update message (INVCON01), the combination of plant and storage location is mapped to the fulfillment center. The default configuration maps WSAL and WSL2 to 10001, which is the FulFillmentCenterID. For your installation, change the values in the CUSTDATA table.

- In product price update message (COND_A02), map the sales organization to the member ID. The default substitution is WS01 (sales organization) with –2000. For your installation, change the values in the CUSTDATA table.

Chapter 6. Installing and configuring

To use the WebSphere Commerce – SAP integration you must install and configure the following components.

- WebSphere Commerce 5.5, Business Edition and configure WebSphere Commerce for WebSphere MQ
- WMQI with CSD3
- MQSeries link for R/3
- Updating literals for WMQI message flows

Installing and configuring WebSphere Commerce

Install WebSphere Commerce 5.5, Business Edition. Refer to the product documentation for the installation steps and the post-install configuration. The WebSphere Commerce messaging system enables messages to interact with back-end systems. For details, see Chapter 4. Message flow.

You must update the CMDREG table, which is the command registry table in your WebSphere Commerce database to use the XML message format.

To enable the Report_NC_PurchaseOrder message, which is the OrderCreate XML message, update the CMDREG table using the following SQL statement:

```
update cmdreg set classname =
'com.ibm.commerce.messaging.commands.SendXMLOrderCmdImpl'
where interfacename =
'com.ibm.commerce.order.commands.OrderMessagingCmd'
```

Note: The above commands will take effect only after restarting the server or refreshing the registry using the WebSphere Commerce Administration Console.

Installing and configuring WebSphere MQ

Install WebSphere MQ 5.3 using the documentation provided with the WebSphere MQ product.

To configure WebSphere MQ to work with WebSphere Commerce, WebSphere Commerce requires a queue manager and a minimum of five queues for integration. The five queues include:

Queue	**Queue description**
Error	Default error queue. Collects erratic inbound messages.
Inbound	Used by SendReceiveImmediate mode of the adapter for WebSphere MQ.
Parallel inbound	Any message arrive at this queue will be processed in parallel manner.

Queue	Queue description
Serial inbound	Any message arrive at this queue will be processed in serial manner based on first-in-first-out.
Outbound	Used for WebSphere Commerce initiated outbound messages and reply messages from WebSphere Commerce.

This reference application uses the following:

- A queue manager
- Serial inbound queue
- Outbound queue

For instructions on creating a queue manager and queues, refer to the WebSphere MQ documentation.

For the sample topology provided in this guide, your outbound queue in WebSphere Commerce must be created as a remote queue to enable communication with the WMQI system. The serial inbound queue in WebSphere Commerce must be created as a local queue to receive messages from the WMQI system.

Similarly, you must create a queue manager and queues for the WMQI system. The queues required to communicate with the WebSphere Commerce messaging system components are:

- A local queue definition, for processing the messages coming from the WebSphere Commerce system.
- A remote queue definition that will be mapped to the WebSphere Commerce serial inbound queue. This queue will send messages to the WebSphere Commerce system.
- The following table shows how the queues must be mapped between WebSphere Commerce and WMQI:

WebSphere Commerce	WMQI
Outbound queue (remote queue)	Inbound queue (local queue)
Serial inbound queue (local queue)	Outbound queue (remote queue)

Additionally, you need to create MQ channels for communication between the two WebSphere MQ servers. For more information refer to WebSphere MQ documentation on inter-communication.

After you create the queues, follow the instructions given in the WebSphere MQ section from the *IBM WebSphere Commerce Additional Software Guide Version 5.5*. The instructions include information on how to use WebSphere Commerce and WebSphere Application Server with WebSphere MQ.

Configuring the WebSphere Commerce messaging system

To configure the WebSphere Commerce messaging system, start the WebSphere Commerce Site Administration Console. Log in as a Site Administrator, go to the **Configuration** section and choose the **Transport** option. Select **WebSphere MQ** as your transport and change the status to **activate**.

Log out of the Administration Console.

Note: When you complete publishing the store as described in "Publishing the store", log into the WebSphere Commerce Administration Console as a Store Administrator and select the WMQISAPToolTech store from the **Configuration** section add **MQ Transport** to the store.

Installing and configuring WMQI Version 2.1

The WMQI install shield wizard installs both the WMQI and WebSphere MQ. If WebSphere MQ is already installed, only the WMQI components will be installed. The WMQI installation and post-install configuration steps are available in the product documentation. The following section covers the message sets and message flows that you must set up for the WebSphere Commerce – SAP integration.

Note: In this document the message repository database for WMQI is referred to as MQSIMRDB.

Setting up the WMQI message sets

1. Use the import.bat batch file from the wmqi directory to import the message repositories into the message repository database, MQSIMRDB. Ensure the following before you begin the import:

- You have exited from the WMQI control center.
- You have stopped the configuration manager and broker.

Note: For information on the directory structure, refer to the readme.txt file available in the integration package that you have downloaded.

2. Open the import.bat file and modify the message repository database name, user ID, password, reference application directory, and MQSIBinDir parameters in the batch file. This batch file creates the CUSTDATA table that contains the name and values of the literals. Modify the values of the variables in the insert statements to suit your installation. For details see, "Literals".

3. Save the changes and run the import.bat batch file from a DB2 command window.

4. After the import is complete start the configuration manager and broker services.

5. Start the WMQI control center.

6. To create a workspace, from the **File** menu select **New Workspace**. To save this new workspace, from the **File** menu select **Save Workspace**, for example refapp.xml.

7. From the **Message Sets** tab, right-click **Message Sets** in the left frame. Select **Add to workspace** and select **Message Set**.

8. In the next window select all the message sets that you imported. The following is a list of message sets that you must select. Click **Finish** to add these message sets to your workspace.
   a. OrderIDOC
   b. Orders05DTD
   c. OrderConfirmation
   d. OrderDeliveryDTD
   e. OrderDelveryIDOC
   f. OrderInvoice
   g. CustomerCreateDTD
   h. CustomerCreateIDOC
   i. PPUPD01
   j. PQUPDDTD
   k. PQUPDIDOC
9. From the **File** menu select **Save workspace**.

Setting up WMQI message flows

1. From the WMQI control center go to **File** and then **Import to Workspace**.

2. From the Import Resources window select the **Message Flows** check box. Note that this is selected by default.

3. Click **Browse** and select the message flow XML files from the wmqi directory. The two XML files are TotalFlow02.xml and OrderFlow01.xml.This adds the message flows to the workspace. Save the workspace.

4. To change the input queue name to suit your set up, right-click the input node for each message flow and select **Properties**. Move to the **Basic** tab and make the change.

5. To change the queue manager and output queue names to suit your setup, right-click the input node for each message flow and select **Properties**. Move to the **Basic** tab and make the changes.

6. Right-click the Trace node and select **Properties**. Change the **File path** for each Trace node.

Deploying WMQI message sets and message flows

1. Move to the **Topology** tab in the control center.
   a. Select **Topology** from the left pane, right-click and select **Check out**.
   b. Once again right-click **Topology** and select **Create** and then **Broker**. Type the name of the broker and the queue manager.

   Note: The name of the broker must be identical to the one you assigned when creating the broker, using the mqsicreatebroker command as part of post-installation steps. Ensure that the queue manager name is identical to the one that you have created previously in this section.

   c. Select **Topology** from the left frame and right-click. Select **Check in**.
2. Move to the **Assignments** tab in the control center.

    a. From the left frame check out MQSI_SAMPLE_BROKER and the default execution group.
    b. Drag and drop the message flows from the center frame to the default section in the right frame.
    c. Drag and drop all message sets associated with the message flows into the broker node, for example MQSI_SAMPLE_BROKER.

3. Check in MQSI_SAMPLE_BROKER and the default execution group.

4. From the **File** menu click **Save workspace**.

5. From the **File** menu go to **Deploy**, **Complete Configuration (all types)** and select **Forced**. This deploys the entire configuration made in WMQI. If the deployment is successful, then a message confirms the same.

IDoc parser

The samples in this solution use the WMQI support pack called the Category 2 SupportPac™. Alternatively, a Category 1 SupportPac is also available as a service offering that provides SAP IDoc processing capability with WMQI. The Category 1 SupportPac also enables loading SAP IDoc type metadata into the WMQI message repository and processing of basic and extended IDocs at run time. For more details about the Category 1 SupporPac, refer to http://www.ibm.com/software/ts/mqseries/txppacs/ia0p.html

The IDoc parser is a WMQI parser for representing incoming data in a format that the WMQI compute node can manipulate. It also allows data of a different format to be mapped to an IDoc stream. Download the IDoc parser for WMQI 2.1 with CSD3 from the following URL as a Category 2 SupportPac. For more details on how to use the parser, refer to the IDoc parser documentation. Refer to the readme.txt file for information on how to use the parser for WMQI 2.1. For more information refer to http://www.ibm.com/software/integration/support/supportpacs/individual/ia0f.html

Literals

The component that requires updates or changes is a literal. A literal carries a fixed value. Literals store values that the WMQI broker will use at runtime for the following:

- To evaluate conditions while parsing and transforming messages.
- To set substitution values.
- To set the data that is required by the target application and not available in the source application.

The default installation of the integration of WebSphere Commerce with SAP assumes a set of parameters that are given as default values to literals. When there is a change in any of these parameters, there will be a change in the corresponding literal. These literals are stored in the CUSTDATA table in MQSIMRDB message repository database. Modify the SQL provided according to your set up. The following table lists the literals to be changed in accordance with the installation procedure:

Literal to be changed	**Description of the value**	**Sample value**
SAP_ORDERS05_SNDPRN	Partner number of sender	ZWSIB100

Literal to be changed	**Description of the value**	**Sample value**
SAP_ORDERS05_SNDPOR	Sender port (SAP system, external subsystem)	ARFC
SAP_ORDERS05_SNDPRT	Partner type of sender	LS
SAP_ORDERS05_RCVPRN	Partner number of recipient	ZWSOB100
SAP_ORDERS05_RCVPRT	Partner type of recipient	LS
SAP_ORDERS05_MANDT	Client code	100
SAP_ORDERS05_DocRel	Document Release	46C
WC_SAP_Def_DistributionChannel1	SAP distribution channel that is mapped to the WebSphere Commerce Business Edition store	30
WC_SAP_Def_Division1	SAP division that is mapped to the WebSphere Commerce Business Edition store	30
WC_SAP_Def_SalesOrganization1	SAP sales organization that is mapped to WebSphere Commerce Business Edition Store	WS01
WC_SAP_Def_StoreID1	WebSphere Commerce Business Edition store ID that is to be mapped to the above sales organization, division and distribution channel	10001
WC_SAP_OrderType	The order type that is expected by the SAP system	ZOR1
WC_FFM	The WebSphere Commerce Business Edition fulfillment center that is mapped to the SAP plant and storage location dedicated to web sales	10001
WC_MemberID	WebSphere Commerce Business Edition member ID that owns the store and products	7000000000000000001
TradingPositionContainerID	WebSphere Commerce Business Edition trading position container ID	10001
SAP_Plant	Plant name in SAP along with storage location is mapped to fulfillment center in WebSphere Commerce Business Edition	WSAL
SAP_Storage_Location	Storage location in SAP along with plant is mapped to fulfillment center in WebSphere Commerce Business Edition	WSL2

Installing and configuring MQSeries link for R/3

To install the MQSeries link for R/3 refer to the user's guide provided with the MQSeries link for R/3 product.

To configure the MQSeries link for the R/3 adapter do the following:

1. Define the TCP/IP ports for use with the operating system, if needed.

   If you already have the SAP GUI installed on the machine where the adapter is installed, you do not have to define the TCP/IP ports. To define them, follow the instructions outlined in the *MQSeries Link for R/3 User's Guide*.

2. Define the RFC destinations in SAP R/3. Refer to the SAP help for more information.

   Note: A user exit can be used in the outbound server to add and remove headers and perform any other logic that may be required. For this reference application, the standard R/3 link adapter exits are used and no user exits are added.

3. Configure destinations for the outbound servers.

   Configure the destinations for the outbound server in the smqDestConf file located in the samples directory. Refer to the *WebSphere MQSeries Link for R/3 User's Guide* for an explanation of each key in the file. Configure the smqDestConf file in the MQSeries link for R/3 bin directory.

4. Specify the server configuration in the initialization (.ini) files.

   The outbound server uses the out.ini file, and the inbound server uses the in.ini file for their startup configuration. Refer to the *MQSeries Link for R/3 User's Guide* for more information. The sample in.ini and out.ini files are provided in the samples directory.

5. Define the queue manager and queues.

   The queue manager and queues used for the inbound server must be the same as the outbound queue of WMQI. Similarly, the queue manager and queue for the outbound server must be the same as the inbound queue of WMQI. In addition there are few more queues required for the R/3 link to be functioning.

Note: The queue names must be identical to those provided in the sample in.ini and out.ini files.

Publishing the sample store

To showcase this integration, you can use the sample store provided with this reference application to create a new store.

This section covers the following:

1. Instructions required before publishing the sample store.

2. Publishing the sample store

Before publishing the store

Do the following before publishing the store:

Populating the units of measure
This reference application provides a massloadable XML file to upload all the unit of measures defined by SAP into WebSphere Commerce. However, this massloadable XML file does not include the descriptions for the unit of measures. To populate the descriptions, extract them from SAP and update UnitOfMeasure.xml present in the store\uom directory. This enables you to use the unit of measures in WebSphere Commerce Accelerator. Do the following to populate the unit of measures:

1. Edit the ImportUOM.bat file from the store\uom directory in a text editor and change the class paths and database information in accordance with your WebSphere Commerce installation. Save this file

2. From a DB2 command window, go to the store\uom directory and run the following command:

   ```
   ImportUOM –infile UnitOfMeasure.xml –method sqlimport
   ```

   This populates the unit of measures in the WebSphere Commerce tables according to the standard SAP installation. The UnitOfMeasure.xml file is present in the store\uom directory.

Populating the state codes
This reference application provides a massloadable XML file to upload all of the state codes defined in SAP for United States of America, Canada, and Japan to WebSphere Commerce. To populate the state codes, do the following:

1. Open the file StateCodes.xml file from the store\statecode directory, in an editor and populate it with the appropriate state codes used in your SAP installation. Save the changes.

   Note: The following step is required only if you want to add new state codes for other countries or modify the state codes provided in the XML file.

2. Open the file ImportStateCode.bat file present in the store\statecode directory in an editor and change the class paths and database information as per your WebSphere Commerce installation. Save this file and execute it from a DB2 command window as shown below:

   ```
   ImportStateCode -infile StateCodes.xml –method sqlimport
   ```

   This uploads the state codes to WebSphere Commerce according to your SAP installation.

Publishing the store

The sample store model provided with this solution uses IBM WebSphere Commerce Payments. For details about installing and configuring IBM WebSphere Commerce Payments refer to the *WebSphere Commerce Installation Guide*.

Creating a new store
The WMQISAPToolTech.sar file for this reference application is present in the store directory. This store archive file is built on top of the business direct store model that comes as part of the WebSphere Commerce Business Edition standard installation.

To enable the store model do the following:

1. Copy the WMQISAPToolTech.sar and Feature_saptooltech_en_US.html files from the store directory, to *WC_installdir/*samplestores/BusinessDirect folder.

2. Edit *WC_installdir/*xml/tools/devtools/SARRegistry.xml file by adding the following lines before </SAR-properties> and save the file.

```
<SampleSAR fileName="WMQISAPToolTech.sar"
relativePath="BusinessDirect">
<html locale="en_US"
featureFile="BusinessDirect/Feature_saptooltech_en_US.html"
sampleSite="BusinessDirect/preview/en_US/index.html"/>
</SampleSAR>
```

3. Launch the WebSphere Commerce Administration Console and publish this store. In the Store Archives page select the .sar file created in the previous step and click **Publish.** It may take a few minutes to complete publishing the .sar file. When complete, the status changes from **Publishing** to **Publishing completed successfully**.

WebSphere Commerce program adapter security

The samples provided with this reference application use a default messaging security that WebSphere Commerce provides. The default security is dependent on the WebSphere MQ security of the queues. Alternatively, you can enable additional security on messages in WebSphere Commerce. For more information on how to change Program Adapter Security for MQSeries, refer to the WebSphere Commerce documentation available with the product.

Modifying WMQI ESQLs for credentials

To include information about credentials for messages inbound to WebSphere Commerce, modify the WMQI ESQLs manually.

1. CustomerNew message - No changes are required.

2. CustomerUpdate message - Include the following two lines of ESQL in the control area section:

```
SET
"OutputRoot"."XML".(XML.Element)"Create_WCS_Customer"."Control
Area"."Credentials"."LogonID" = 'logonId';

SET
"OutputRoot"."XML".(XML.Element)"Create_WCS_Customer"."Control
Area"."Credentials"."Password" = 'password';
```

   Ensure that the logon ID and password are identical to the ones created by the user.

3. ProductPriceUpdate message - Include the following two lines of ESQL in the control area section:

```
SET
"OutputRoot"."XML".(XML.Element)"Create_WCS_Customer"."Control
Area"."Credentials"."LogonID" = ‘logonId’;

SET
"OutputRoot"."XML".(XML.Element)"Create_WCS_Customer"."Control
Area"."Credentials"."Password" = ‘password’;
```

The logon ID and password must belong to the UserGroup that has the authority to update prices.

4. ProductQuantityUpdate message

   Include the following two lines of ESQL in the control area section:

```
SET
"OutputRoot"."XML".(XML.Element)"Create_WCS_Customer"."Control
Area"."Credentials"."LogonID" = ‘logonId’;

SET
"OutputRoot"."XML".(XML.Element)"Create_WCS_Customer"."Control
Area"."Credentials"."Password" = ‘password’;
```

   The logon ID and password must belong to the UserGroup that has the authority to update quantity.

5. OrderStatus messages

   Include the following two lines of ESQL in the control area section:

```
SET
"OutputRoot"."XML".(XML.Element)"Create_WCS_Customer"."Control
Area"."Credentials"."LogonID" = ‘logonId’;

SET
"OutputRoot"."XML".(XML.Element)"Create_WCS_Customer"."Control
Area"."Credentials"."Password" = ‘password’;
```

   The logon ID and password must belong to the UserGroup that has the authority to update the order status.

6. OrderCreate message - No changes are required.

Configuring the SAP R/3 system

This section covers how to configure the SAP R/3 system for this reference application. Typically, a SAP consultant configures the various settings. The information provided here is a quick reference to the different configurations required in the SAP R/3 system.

For more information on how to configure each of the following, refer to the SAP documentation or the SAP online help:

- Defining an organization structure: For this integration, you can use your existing organization structure or create a new one. Some of the required entities are

Company code, Controlling Area, Plant, Storage location, sales area, and shipping point,

- Customizing master data: Create a new account group and make the REGION and TERMS OF PAYMENT fields mandatory. In addition, assign the sold-to party, ship-to party, bill-to party, and payer partner functions to the new account group.
- Configuring price, freight, and tax: Create the necessary condition tables, condition records, and access sequences for price, freight, and tax calculation.
- Customizing order numbers – You must customize the corresponding field in SAP to store and send the WebSphere Commerce order number in the order status messages, for example, SISCSO01, SISDEL01, and SISINV01 IDocs.

Configuring partner profile communication: Configure the partner profile communication to enable the SAP ALE layer to communicate with the external system; send and receive requests from the SAP connector. The configurations include defining logical systems, RFC destination, maintaining partner profiles, creating and assigning message types, and enabling change pointers.

Chapter 7. Loading master data

This chapter describes how to load master data and reference data. It includes:

- Loading reference data onto SAP
- Loading SAP materials and customer data onto WebSphere Commerce
- Store customizations

Loading SAP materials and customer data onto WebSphere Commerce

The following process can be used to synchronize data between systems. For incremental updates, messages can be exchanged and transformed using WMQI.

To synchronize material data between WebSphere Commerce and SAP, use the Loader Package utilities like TextTransformer, XMLTransformer, IdResolver and Massloader. For more information refer to the WebSphere Commerce online help.

Extracting data from SAP

To extract data from SAP, which includes material and customer master data**,** you need to write report programs in the ABAP editor (transaction code se38). For this, go to Basis Components, ABAP programming, Runtime Components and select BC-ABAP programming. Identify the fields to be extracted as required by the pre-determined flat file, and select the fields from the transparent tables using **Open SQL** select statements. Specify a selection criterion for the program to extract records, for example, material number(s). After this you can execute the report program and download the records from the internal tables, into a file in ASCII format by calling the function module “Download”.

Loading material data

To load material data onto WebSphere Commerce using the massload scripts provided, ensure that you extract the material data from the SAP system in the following sequence and delimited by “~”:

Note: The delimiter expected by the massload script is “~”. If you are using a different delimiter, you must change the delimiter in the import schema files provided with this reference application.

Sequence no.	Data at position	Data description	Corresponding fields in SAP
1	MaterialPartNumber	Part number of the material	MARA-MATNR (Mandatory)
2	MaterialGroupPartNumber	Part number of the material group	MARA-MATKL (Mandatory)
3	Language	Language specification in SAP. An example of language specification in WCS is en US for English	T002T-SPTXT (Mandatory)

Sequence no.	Data at position	Data description	Corresponding fields in SAP
		is en_US, for English.	
4	MaterialName	Name of the material	MAKT-MAKTX (Optional)
5	MaterialShortDescription	Short description of the material.	(Optional)
6	MaterialLongDescription	Long description of the material.	(Optional)
7	MaterialImageName	Filename of the image/picture of the material.	(Optional)
8	MaterialLastUpdatedOn	Indicates the last time the material was updated.	MARA-LAEDA (Optional)
9	MaterialPrice	Amount of the material price.	MBEW-STPRS (Mandatory)
10	Currency	Currency of the material price.	T001-WAERS (Mandatory)
11	MaterialWeightMeasure	The unit of measurement for weight.	MARA-GEWEI (Optional)
12	MaterialSizeMeasure	The unit of measurement for length, width and height.	MARA-MEABM (Optional)
13	MaterialQuantityMeasure	The unit of measure for nominal quantity.	MARA-MEINS (Mandatory)
14	MaterialWeight	The nominal weight associated with the material	MARA-BRGEW (Optional)
15	MaterialLength	The nominal length associated with the material.	MARA-LAENG (Optional)
16	MaterialWidth	The nominal width associated with the material.	MARA-BREIT (Optional)
17	MaterialHeight	The nominal height associated with the material.	MARA-HOEHE (Optional)
18	MaterialNominalQuantity	Nominal quantity for a material, used for pricing. For example, if a material is priced as "3 for a dollar", then the nominal quantity of the material is 3, and the price of the material is one dollar	MVKE-AUMNG (Mandatory)
19	MaterialDataIndicator	An indicator that specifies whether the data for that material is for CREATE or UPDATE.	CDHDR-CHANGE_IND (Mandatory)
20	ManufacturerName	The name of the manufacturer of this material	(Optional)

Sequence no.	Data at position	Data description	Corresponding fields in SAP
21	ManufacturerPartNumber	The part number used by the manufacturer to identify this material	(Optional)
22	Material group name	Name of the material group to which this material is associated	T023T-WGBEZ
23	Material group description	Description of the material group to which this material is associated	T023T-WGBEZ60

The attributes information for items in WebSphere Commerce is optional. You can load the materials without attributes. If you are loading the attributes for materials, then extract the characteristics information for these materials in the following sequence and delimited by “~”:

Note: The delimiter expected by the massload script is “~”. If you are using a different delimiter, you must change the delimiter in the import schema files provided with this reference application.

Sequence No.	Data at position	Data description	Corresponding fields in SAP
1	MaterialPartNumber	Part number of the material	AUSP-OBJEK (Mandatory)
2	MaterialGroupPartNumber	Part number of the material group	AUSP-KLART (Mandatory)
3	MaterialCharacteristicName	Name given to the material characteristic.	CABN-ATNAM (Mandatory)
4	MaterialCharacteristicValue	Value of the respective material characteristic name	AUSP-ATWRT (Mandatory)
5	Language	Language specification in SAP. An example of language specification in WCS is en_US, for English.	T002T-SPTXT (Mandatory)

To upload material data in the specified format do the following:

1. Use SAP transaction SE38 to execute the ABAP programs written, to extract the material information and material characteristics (optional) from SAP. Move the extracted files into the store\dataload\material directory.

2. Open the ManifestFile.txt file from the store\dataload\material directory. Replace the itabmara.txt file with the output file name derived from the material information extract program. Replace the itab.txt file with the output file name derived from the material characteristics extract program. If the program for material characteristics is not executed then delete the following lines.

```
itab.txt,attribute_schema.xml,Output.xml,Append

itab.txt,attrvalue_schema.xml,Output.xml,Append
```

Note: For information on the directory structure, refer to the readme.txt file available in the integration package that you have downloaded.

3. Save and close this file.

4. Open the MaterialUpload.bat batch file from the store\dataload\material directory in an editor and change the following parameters according to your installation:

   DB_NAME - The type of database, which is DB2 in this case.

   WCS_DBNAME – WebSphere Commerce instance database name for example, **mall**.

   WCS_DBUSER - Database user ID.

   WCS_DBPWD - Database user password.

5. Edit the following line to include your WebSphere Commerce installation path:

   WC_installdir\bin\setenv.bat

6. Change the following literal as per your installation.

   ```
   set DB2_HOME=D:\Websphere\sqllib
   ```

7. Change the parameter values passed to the XMLTransformer according to your installation. You can find the values for these parameters in the WebSphere Commerce database tables.

   MemberIdValue - Identifier of the storeowner (ORGENTITY.ORGENTITY_ID).

   TradingPositionName - Trading position name associated with the store (TRADEPOSCN.NAME).

   CatalogName - Catalog identifier of the store (CATALOG.IDENTIFIER).

   ImportLocation=*WC_installdir*\schema\xml\wcs.dtd. This is the location of the wcs.dtd file in WebSphere Commerce.

   StoreIdentifier – Identifier of the store published to showcase this reference application (STORE.STORE_ID).

   FulfillmentCenterName – The fulfillment center name that is associated with the store (FFMCENTER.NAME).

8. Save the changes made in the previous step and run the MaterialUpload.bat batch file from a DB2 command window.

9. Launch the store and check for the products and items under **SAP Products** -> **SAP Category 1** hierarchy.

Note: Any item that is not specifically grouped under a product in SAP can be found under the product SAP10001 in the same hierarchy.

When manufacturer details are not present for a product, by default the product takes "SAP Tools" as the manufacturer name and product part number as the manufacturer part number.

When the short description details are not available for a product, by default it takes the material name as the short description.

When massloading the material data, only names of the product images are loaded. To view the product images on the corresponding page of the store you must manually copy the image files into the following directory:

WAS_installdir\installedApps\WC_Enterprise_App_i*nstance_name*.ear\wcstores.war\ s*tore_name*\images.

Where, *instance_name*.ear is the name of the commerce instance in your installation and s*tore_name* is the name of the store to which the materials were uploaded.

Loading customer data

You can load the customer data using the massload scripts provided with this reference application. Ensure that the customer data is in the following sequence and delimited by "~":

Note: The delimiter expected by the massload script is "~". If you are using a different delimiter, you must change the delimiter in the import schema files provided with this reference application.

Sequence. No.	Data field	Data description	Corresponding fields in SAP
1	LogonId	Independent User: LogonID of the user	KUNNR (mandatory)
2	Password	Independent User: Password with the LogonId for authentication.	SORTL (mandatory) This field has been mapped to SORTL, which is a mandatory field in SAP). You can use any other field for this purpose. Instead of mapping to an existing field you can generate a password before loading the customer data.
3	Title	Title of the person to which this address applies.	ANRED (optional)
4	LastName	Independent user: Last name of the Customer	NAME1 (mandatory)
5	MiddleName	Independent user: Middle name of the Customer	NAME3 (optional)

Sequence. No.	Data field	Data description	Corresponding fields in SAP
6	FirstName	Independent user: First name of the Customer.	NAME2 (optional)
7	Address1		STRAS (mandatory)
8	City		ORT01 (mandatory)
9	State		REGIO (mandatory)
10	ZipCode		PSTLZ (mandatory)
11	Country		LAND1 (mandatory)
12	Phone1		TELF1 (optional)
13	Phone2		TELF2 (optional)
14	Fax 1		TELFX (optional)
15	Profile	Z2 (Independent user)	GFORM (optional)
16	Preferred Currency		UWAER (optional)
17	Preferred Language		SPTXT (optional)
18	Preferred Delivery		INCO2 (optional)

Note: The data field Profile is required to massload the customer data. If this information is not present in SAP, then edit the file extracted from SAP that contains the customer information to include the Profile field. The value expected in the Profile field is Z2.

When you create or update customer data in the SAP system, the data entered in the SORTL field is converted to uppercase.

To upload customer data do the following:

1. Use the SAP transaction SE38 to execute the ABAP programs written to extract customer information from SAP. Move these extracted files to the store\dataload\customer directory.

2. Open ManifestFile.txt file from the store\dataload\customer directory, and replace itabkna2.txt with the output file name derived from the customer information extract program.

3. Open the CustomerUpload.bat batch file from the store\dataload\customer directory in an editor and change the following parameters according to your installation:

DB_NAME - The type of database, which is DB2 in this case.

WCS_DBNAME - WebSphere Commerce instance database name for example, **mall**.

WCS_DBUSER - Database user ID.

WCS_DBPWD - Database user password.

4. Edit the following line to include your WebSphere Commerce installation path:

   WC_installdir\bin\setenv.bat

5. Change the following literal as per your installation.

   ```
   set DB2_HOME=D:\Websphere\sqllib
   ```

6. Change the following parameter values passed to the XMLTransformer according to your installation.

   ImportLocation=*WC_installdir*\schema\xml\wcs.dtd. This is the location of the wcs.dtd file in WebSphere Commerce.

7. Open and edit password.bat file present in the store\dataload\customer directory to include your WebSphere Commerce installation path:

   WC_installdir\bin\setenv.bat

8. Save all the changes made and run the CustomerUpload.bat batch file from a DB2 command window as:
   `CustomerUpload.bat <MerchantKey>`
   where, *MerchantKey* is the unique merchant key provided when creating the Commerce instance.

9. To verify the upload of customer data launch the store and login to check the validity of the user that was uploaded.

Note: During customer data load, USERREG.PASSWORDEXPIRED is set to 1. These users will have to change the password when they login for the first time.

Possible errors when loading initial data
If the initial data loading fails, then refer to the following list of error messages and corresponding solutions. This applies to both material and customer data.

Redirect the output of the MaterialUpload.bat or CustomerUpload.bat batch fles to a log file. Check for the following error(s) in the log file:

1. The system cannot find the path specified.

   This error could be due to some problem when specifying the directory locations for loaderdir, libdir, configdir, and sqllibDir directories. Ensure that the path settings specified in MaterialUpload.bat or CustomerUpload.bat batch files are pointing to the correct directory in the target machine.

2. "<Database_Name>" is not a valid database name

Ensure that you have specified a correct value for `WCS_DBNAME` in MaterialUpload.bat or CustomerUpload.bat batch files.

3. Error: <FileName> (The system cannot find the file specified)

   Ensure that the paths provided for the input files in ManifestFile.txt file are correct.

4. The username and password supplied are incorrect.

   Ensure that the user ID and password supplied have access to your database.

5. Resolution control file not found, going with unique indexes

   If this is the error, then the IDKEYS.properties file is not found or the file name specified for ID_PROPERTIES_File in MaterialUpload.bat or CustomerUpload.bat files does not match with the one in the host machine.

   If there are any other errors, then it could be due to the invalid values for some of the parameters in the batch file or the input data file is incorrect.

Loading reference data onto SAP

To demonstrate this integration you can also use the reference data provided with this reference application. A portion of the sample store model items is packaged as reference data. You must import this data into the SAP R/3 system. This allows the synchronization of WebSphere Commerce and SAP R/3 material data.

BDC_MaterialInput.txt file in the store\dataload\referencedata directory contains the material reference data in the following format that you must load onto SAP using a Batch Data Conversion (BDC) program.

The prerequisite to load material data is to define the material groups in the SAP. Use “SAP Customizing” to do this.

Serial number	**Description**	**Data element as in SAP**	**Length**	**Depends on existent SAP configurations and data**
1	Material Number	MATNR	18	No
2	Industry Sector	MBRSH	1	Yes
3	Material Type	MTART	4	Yes
4	Plant	WERKS	4	Yes
5	Sales Organization	VKORG	4	Yes
6	Distribution Channel	VTWEG	2	Yes
7	Material Description	MAKTX	40	No
8	Unit of Measure	MEINS	3	Yes
9	Material Group	MATKL	9	Yes, you need to define this using SAP

Serial number	Description	Data element as in SAP	Length	Depends on existent SAP configurations and data
				customizing
10	General Item Category Group	MTPOS_MARA	4	Yes
11	Gross Weight	BRGEW	17	No
12	Weight Unit	GEWEI	3	Yes
13	Net Weight	NTGEW	17	No
14	Size/Dimensions	GROES	32	No
15	Description Language	DESC_LANGU_GDTXT	16	Yes
16	Document Number	ZEINR	22	No
17	Class Type	KLART	3	Yes
18	Class Number	CLASS	18	Yes
19	Cash Discount Indicator	SKTOF	1	No
20	First Entry Displayed	PAGPOS	3	No
21	Delivering Plant	DWERK	4	Yes
22	Item Category Group from Mat master	MTPOS	4	Yes
23	Checking Group for availability Check	MTVFP	2	Yes
24	Transportation Group	TRAGR	4	Yes
25	Loading Group	LADGR	4	Yes
26	MRP Type	DISMM	2	Yes
27	MRP Controller	DISPO	3	Yes
28	Lot Size	DISLS	2	No
29	Procurement Type	BESKZ	1	Yes
30	In-house Production Line	DZEIT	3	No
31	Scheduling Margin Key for Floats	FHORI	3	Yes

Serial number	Description	Data element as in SAP	Length	Depends on existent SAP configurations and data
32	Period Indicator	PERKZ	1	No
33	Planning Strategy group	STRGR	2	No
34	Total Replenishment Lead Time	WZEIT	3	No
35	Valuation Class	BKLAS	4	Yes
36	Price Control Indicator	VPRSV	1	No
37	Price Unit	PEINH	6	No
38	Standard Price	STPRS	15	No

BDC_SellingPriceInput.txt file in the store\dataload\referencedata directory contains the standard price for material reference data in the following format that you must load onto SAP using a BDC program.

Defining the price condition in SAP is a prerequisite to load this data.

Serial number	Description	Data element as in SAP	Length	Depends on existent SAP configurations
1	Pricing Condition Type	KSCHL	4	Yes
2	Material Number	MATNR	18	No
3	Material Selling Price	KBETR	16	No
4	Currency	KONWA	5	No

To import reference data into SAP as a batch process from the input file, you need a Batch Data Conversion (BDC) program. This program requires you to record the transaction to load the data. To do this, use transaction SHDB and do the following:

1. Using the SAP client, from the SAP Easy Access screen go to SHDB transaction and enter the record name. Follow the onscreen instructions to record the MM01 transaction and then record the VK12 transaction.

2. Based on the values in your SAP installation, change the data in the reference data files before importing.

3. Use transaction SE38 to execute the recorded programs.

4. After loading the reference data, create inventory records before placing any orders for these materials.

Store customizations

This reference application is built on top of the business direct store model. The reference store contains catalog data and Web assets in the English language only. For more information, refer to the *Store Development Guide*. The following changes are made to the business direct store in order to achieve the SAP integration functionality:

Reference Data: The catalog related XMLs are modified to populate the reference data. The list of XMLs includes catalog.xml, en_US/catalog.xml, offering.xml, and others.

Disable ATP: The Store.xml file is modified to change the ALLOCATIONGOODFOR to 0 because if ALLOCATIONGOODFOR is zero, ATP inventory allocation is not used by the store **Inventory:** The Storefulfill.xml file is modified to populate the inventory details for the reference catalog items.

Store Language: The store.xml file is updated to support the English language only. Only en_US locale specific properties are provided for this reference application.

Address page: Modifications to the AddressForm.jsp are made provide a selection box to select the country and state codes. This information is mandatory in SAP for customer registration.

Shipping page: The BillingShippingDisplay.jsp file is modified to use OrderDisplay as the redirection URL instead of AllocationCheck.

Order Display Pending and Order Confirmation pages: The OrderSubmitForm.jsp file and the OrderConfirmation.jsp files are modified not to include subtotals, tax, and shipping details.

Track Order Status: The OrderStatusDisplay.jsp file is modified to allow you to retrieve the list of orders confirmed, shipped or invoiced, based on details available in ORDSTAT and ORDISTAT tables. The possible status values of the orders are: C (Confirmed), S (Shipped), and I (Invoiced). A link is provided for each of the orders in the list to view the detailed order status.

Order Status Details: The OrderDetailDisplay.jsp file is modified to display the detailed order status available in the order status tables.

Shipping Modes: Shipping.xml is modified to replace the shipping codes A1, A2, A3 with BYTRUCK, BYRAIL, BYAIR respectively, and shipping carrier 'XYZ Carrier' with CFR. en_US/Shipping.xml is modified to set the description for the preceding shipping modes to 'CFR-TRUCK', 'CFR-RAIL', and 'CFR-AIR' respectively. The changes to the Shipping.xml are made to match the shipping modes provided by SAP. These shipping modes need be used when placing the order using the default contract. If any other contracts are created, then you can use these shipping modes while defining terms and conditions, otherwise create new shipping modes before they are used in the terms and conditions of the contract. For more details, refer to the WebSphere Commerce documentation.

Chapter 8. Verification procedure

Use the following steps to determine whether your WebSphere Commerce Business Edition – integration with SAP setup is working properly. Ensure that the necessary configuration in the SAP R/3 system has been performed and that the following services are running before testing individual messages:

- WebSphere Commerce Business Edition instance
- WMQI services
- MQSeries link for R/3 inbound server
- MQSeries link for R/3 outbound server
- IBM WebSphere MQSeries service

Running the servers

The following are the servers you must run in the WebSphere Commerce – SAP integration.

IBM WebSphere MQSeries service

The IBM WebSphere MQSeries service can be run from the services in the control panel in Windows NT or Windows 2000.

WMQI services

WMQI services, Configuration Manger and Broker, can be started from the **Services** in the control panel in Windows NT or Windows 2000.

WebSphere Commerce instance

You can start the WebSphere Commerce instance from the Administration Console of the WebSphere Application Server. Ensure that the WebSphere Administrator Server is running and the instance has been created. To start the WebSphere Commerce instance, expand the tree for the host name where WebSphere Commerce is installed. Select the instance name and click the **Run** icon.

Inbound server

1. Copy the in.ini file from the samples directory to *MQlink_installdir*\bin and make the necessary changes to the file.
2. Go to *MQlink_installdir* \bin and run the following command:

```
smqsi –iin.ini
```

Outbound server

1. Copy the out.ini file from the samples directory to *MQlink_installdir* \bin and make the necessary changes to the file.

2. Go to *MQlink_installdir*\bin and run the following command:

```
smqso –iout.ini
```

Testing individual messages

Order create message (ORDERS05)

To create an order and check if the order create message is generated and processed correctly, do the following:

1. Ensure that the customer, product, pricing, and inventory data in WebSphere Commerce is consistent with that in SAP.

2. Log into the WebSphere Commerce store as a customer, using a valid user ID. You must use the logon ID of the customer created in SAP. After creating the customer in the WebSphere Commerce from SAP, assign the role of Registered Customer to the new user. Complete this step before logging into the store.

3. Add items to the shopping cart and submit an order. This must generate the order create XML message. It may take some time to deliver the message, as determined by the scheduler configuration in WebSphere Commerce. By default, the scheduler process for sending messages into WebSphere MQ runs at an interval of five minutes.

If the order create XML was successfully parsed, formatted, and sent to the SAP system, the SAP transaction we02 shows the status of the incoming ORDERS05 IDoc. The R/3 link inbound server shows the receipt of IDocs in the console.

If the message passes through WMQI but encounters an error in the R/3 link inbound server processing, then details of the error will be placed in the error file associated with the inbound server.

Order confirmation status message (SISCSO01)

To check if the order confirmation status message is generated and processed correctly do the following:

1. In response to a successful order creation in SAP, SAP sends the order confirmation status message. This is in the form of SISCSO01 IDoc and can be verified using SAP transaction we02.

2. If the message is processed successfully, then the status of the corresponding order in WebSphere Commerce is set to 'G' in the ORDERS table. The respective order confirmation entries are made in the ORDSTAT and ORDISTAT tables.

3. Log into the WMQISAPToolTech store. Select **Order Status** and look for the order in the Orders Confirmed section.

Errors can be detected in the error file for the outbound server, system application logs, or WebSphere Commerce Business Edition logs. In WebSphere Commerce, the logging for the messaging component must be enabled.

Order delivery status message (SISDEL01)

To check if the order delivery status message is generated and processed correctly do the following:

1. To create the SISDEL01 IDoc, run SAP transaction va02.

2. Type the order number that SAP generates. This can be ascertained from the SISCSO01 IDoc generated previously. If the order creation was unsuccessful, then you will not be able to do this.

3. When the page showing the details of the order appears, select **Sales Document - Deliver** from the menu.

4. This process creates the delivery message unless an error occurs. The SAP client in the error log shows the errors.

5. Log into the WMQISAPToolTech store. Select **Order Status** and look for the order in the Orders Shipped section.

   To verify the status, check the ORDSTAT and ORDISTAT tables in WebSphere Commerce. The status of the corresponding order items should be 'S'.

Note: WebSphere Commerce allows versioning of order status messages. Depending on the option selected, either the existing status record is updated or a new record is added to the tables. By default, the order status header and the order status item are not versioned.

Product inventory update message (INVCON01)

To check if the product inventory update message is generated and processed correctly do the following:

1. To generate INVCON01 IDoc, run transaction vl02. This displays the number of the delivery document created previously. Details of the order display.

2. Type the storage location (WSL2) and the “picking” quantity in the table for the line item.

3. Click **Post Goods Issue**. This creates the INVCON01 IDoc.

4. To verify in WebSphere Commerce, check the inventory of the ordered material in the INVENTORY table.

Order invoice status message (SISINV01)

To check if the order invoice status message is generated and processed correctly do the following:

1. To create SISINV01 IDoc, run the transaction vf01. This creates the SISINV01 IDoc. When vf01 runs you are asked for the billing type.

2. Select **Invoice (F1)** from the menu. This automatically retrieves the document number for the delivery created above.

3. If the document number does not display, then select the document number from the menu.

4. Login to the WMQISAPToolTech store. Select **Order Status** and look for the order in the Orders Invoiced section.

5. To verify if the IDoc is created, check the ORDSTAT and ORDISTAT tables in WebSphere Commerce. The status of the corresponding order should be set to ‘I'.

Product price update message (COND_A02)

To check if the product price update message is generated and processed correctly do the following:

1. To generate a COND_A02 message, run SAP transaction vk12. When prompted for the condition type, type the appropriate pricing condition you have created.

2. When the transaction runs, you are asked for the key combination. Based on your preference, select the option. Enter the required fields on the next screen, and run the transaction.

3. In the next screen, change the rate for a ‘WCBE-known’ material and save the document.

4. Run SAP transaction se38 to generate the IDocs.

5. Type `RBDMIDOC` as the program name and execute.

6. In the next screen type the message type as `COND_A`. and run the program. This creates the IDoc but the IDoc will not be sent.

7. To send the IDoc, again go to SAP transaction se38 and run the RBDOUTPU program.

8. Select **dispatch** and run the program.

9. Type `COND_A02` as the basic type and run the program. This should dispatch the COND_A02 IDOC.

10. Log into the CWSAPToolTech store. Check the price of the order item after adding it to the shopping cart.

11. Verify the change in price by checking the OFFERPRICE and OFFER table in WebSphere Commerce

Customer create or update message (DEBMAS05)

To check if the customer create or update message is generated and processed correctly do the following:

1. Generate a DEBMAS05 message by creating a new customer (SAP transaction xd01) or by changing the details of an existing customer (SAP transaction xd02).

2. Execute SAP transaction se38 to generate the IDoc. Select the program RBDMIDOC and run the program.

3. Type `DEBMAS` as the message and run the program. This creates the IDoc but does not dispatch it.

4. To send the IDoc, go to SAP transaction se38 and run the RBDOUTPU program.

5. Select **dispatch** and run the program.

6. Type `DEBMAS05` as the basic type and run the program. This sends the DEBMAS05 IDoc.

7. To verify, check for corresponding values in the USERREG, ADDRESS, and USERS tables in WebSphere Commerce.

8. Log into the WMQISAPToolTech store. Select the **Account** and click **Change Personal Information**. Note the change in the address.

Verifying the master data upload

Loading material data

1. Extract the data from SAP into a delimited file in accordance with the format defined in Chapter 7. Loading master data.

2. Modify the ManifestFile.txt file and MaterialUpload.bat batch file as mentioned in Chapter 7. Loading master data.

3. Run the `MaterialUpload.bat batch file` from a DB2 command window. This populates the CATGROUP, STORECGRP, CATTOGRP, CATGRPREL, QTYUNIT, BASEITEM, ITEMSPC, OFFERPRICE, CATGPENREL, STORECENT, CATENTDESC, LISTPRICE, CATENTSHIP, CATENTREL, ATTRIBUTE, ATTRVALUE, OFFER, and INVENTORY tables.

4. You should be able to view the products and items in the catalog, under the top category **SAP Products**.

Loading customer data

1. Extract the data from SAP into a delimited file in accordance with the format defined in Chapter 7. Loading master data.

2. In the extracted file for each customer record, the **Profile** field must have the value Z2 for the customer data of an independent user. If this information is not available in SAP ensure that you edit the extracted file to include this before loading the data into WebSphere Commerce.

3. Modify the ManifestFile.txt file and CustomerUpload.bat batch file as mentioned in Chapter 7. Loading master data and run the CustomerUpload.bat from a DB2 command window.

4. To verify, login to the WMQISAPToolTech store with the logon ID, which is the customer number registered in SAP and the Password, which is the value that is

entered in the "Search term 1" text field when registering the customer in SAP. If the login is successful, you will be prompted to change your password.

The MEMBER, USERS, USERREG, ADDRESSBOOK, ADDRESS, BUSPROF, USERPROF, and USERDEMO tables are populated.

Chapter 9. Adding new messages to WebSphere Commerce

The WebSphere Commerce integration with SAP R/3 currently supports all the messages mentioned in Chapter 4. Message flow. This reference application provides interfaces to extend the current support to new inbound and outbound messages that you may add.

Inbound message into WebSphere Commerce

Use the user_template.xml file inbound message template definition file to add a new inbound XML message. For more information on how to process a new inbound message, refer to the WebSphere Commerce Business Edition online documentation. Complete the following steps to support new inbound messages to WebSphere Commerce. Here, we have used material master as an example:

1. Identify the corresponding IDoc in the SAP R/3 system for the new message. For example, MATMAS in SAP R/3 is for Materials Master.

2. Configure SAP to generate and send the IDoc message. This must happen whenever a new material or product is added, or a current material or product attribute changes.

3. Identify the support required in WebSphere Commerce to process this new message. Add a new command if there is no existing command to process the inbound message.

4. WebSphere Commerce must contain the XML message definition to support the new inbound message. If not, create a new definition (DTD). Enter the information about the new DTD in the instance properties. Open the Configuration Manager, expand InstanceList node and select the relevant *instance_name*. Select the **Messaging** node under *instance_name* \Instance Properties node. In the right frame add the details of the new DTD under the Inbound Message DTD files parameter. For information on how to work with the Configuration Manager refer to the WebSphere Commerce documentation.

5. Provide information in the user_template.xml file on how to parse the new message added and the command that must process this message. In this file, indicate the controller command that the new message invokes, define the elements of the message, and indicate the command parameters to which the element corresponds. Refer to the WebSphere Commerce Development online help for information on how to create user_template entries.

6. Before writing the ESQLs in WMQI, complete the mapping required between the SAP IDoc segments or fields and WebSphere Commerce XML elements. See Appendix B. Mapping information for the messages supported in this integration.

7. Import the XML DTD and the IDoc message definition into WMQI. You need to import the IDoc C-header file extracted from the SAP system. For more information refer to the IDoc parser documentation.

8. Create a message flow and write the ESQLs in WMQI to transform the IDoc message into an XML message.

9. Using the WMQI MQOutput node, route the message to the appropriate target queue, which is the message queue configured for WebSphere Commerce.

Outbound message from WebSphere Commerce

To add a new outbound message, write a new task command to build the new message. Send this message to the SAP system using the send services of the outbound messaging system. For more information on how to create a new outbound message, refer to the WebSphere Commerce Development online help. The following is a sample for the customer or user registration message; use the same steps to create a new message:

1. Add a new task command to generate the message. The command can use the composition services provided by the messaging system to compose the content or, it can build its content for the message.
2. If you are using the message composition service, define the DTD and create the JSP for the new message.
3. Identify the corresponding IDoc for the new message that SAP supports.
4. Import the DTD and IDoc C-header into WMQI for message sets.
5. Before writing the transformation and message flows in WMQI, complete the necessary mapping between the SAP IDoc segments or fields and WebSphere Commerce XML elements.
6. Write the ESQLs in WMQI to transform IDoc messages to XML messages.
7. Using the WMQI MQOutput node route the message to the appropriate target queue, which is the message queue configured for SAP.

Note: A user exit is not required for outbound messages since the ESQLs add the SAP header information.

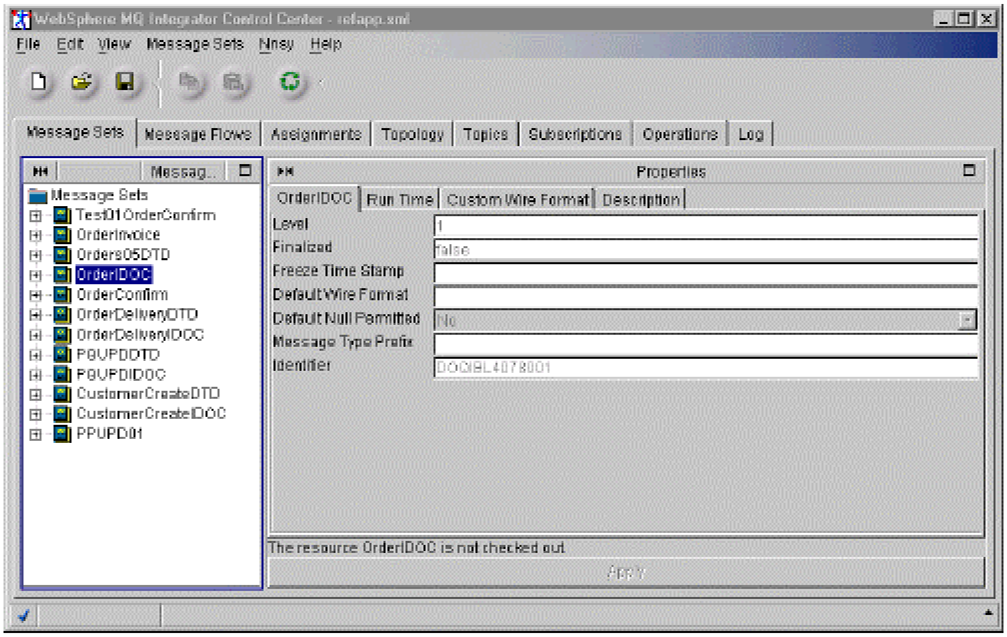
Appendix A. WMQI message sets and message flows

This reference application uses WMQI message sets and message flows to exchange information between WebSphere Commerce and SAP.

WMQI message sets

A message set is a collection of messages. It is a central repository or dictionary of message definitions associated with a business project. The MRM (Message Repository Manager) maintains these messages in the message repository, which comprises a set of tables in a database.

In the following figure, the Control Center lists the message sets for all the messages supported in this integration. Messages are defined or imported using the control center. The configuration manager stores and manages the definitions in the message repository.



Message Sets
- Test01OrderConfirm
- OrderInvoice
- Orders05DTD
- OrderIDOC
- OrderConfirm
- OrderDeliveryDTD
- OrderDeliveryIDOC
- PGUPDDTD
- PGUPDIDOC
- CustomerCreateDTD
- CustomerCreateIDOC
- PPUPD01

Level: 1
Finalized: false
Freeze Time Stamp:
Default Wire Format:
Default Null Permitted: No
Message Type Prefix:
Identifier: DOCIBL4078001

The resource OrderIDOC is not checked out

Figure 4: WMQI message sets

WMQI message flows

Each action or subset of actions is implemented as a message-processing node, and these are wired together in a sequence to form a message flow. You can create message flows using the control center. For more information refer to the WMQI documentation.

In this integration two message flows are used, one for inbound messages to WebSphere Commerce, and the other for outbound messages from WebSphere Commerce. For inbound message flows the aggregatecontrol node, the resetcontentdescriptor nodes and the filter nodes are used to route the message to the appropriate compute nodes.

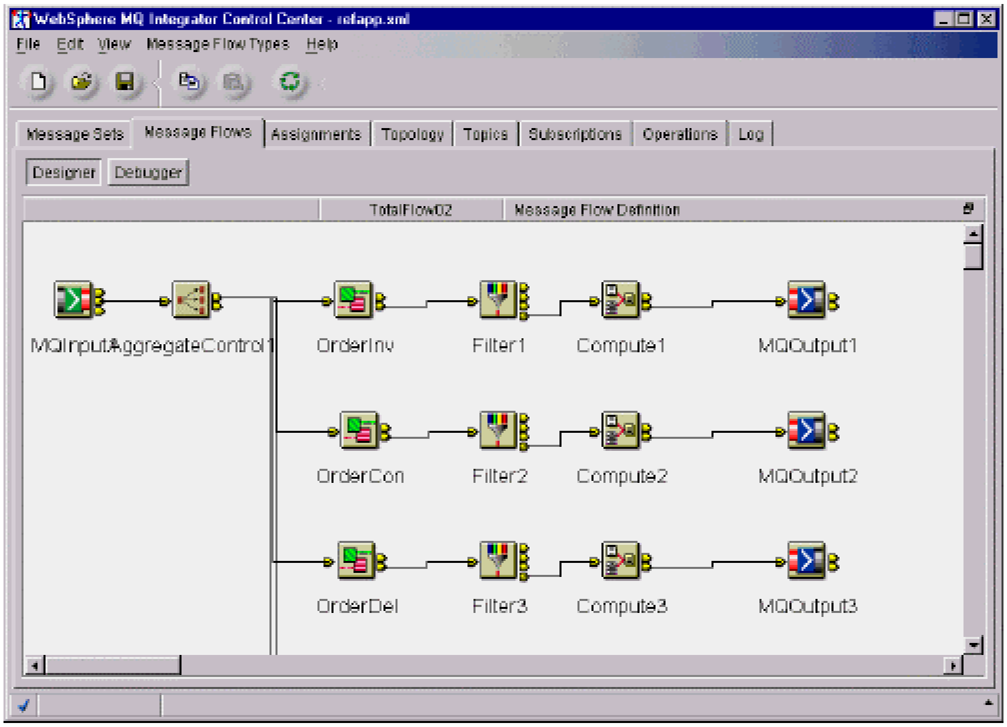


Figure 5: WMQI message flows

Appendix B. Mapping information

This section includes the DTDs used for messages supported by IBM WebSphere Commerce - SAP integration, and the necessary mapping information for these messages.

Table 1: Messages, IDocs and DTDs used

Messages	**IDocs**	**DTD**
Customer Create	DEBMAS05	Create_WCS_Customer_20.dtd
Customer Update	DEBMAS05	Update_WCS_Customer_20.dtd
Order Create Message	ORDERS05	Report_NC_PO_10.dtd
Product Price Update	COND_A02	Update_WCS_ProductPrice_20.dtd
Product Inventory Update	INVCON1	Update_WCS_ProductInventpry_20.dtd
Order Confirm Status Message	SISCSO01	Update_WCS_OrderStatus_20.dtd
Order Delivery Message	SISDEL01	Update_WCS_OrderStatus_20.dtd
Order Invoice Message	SISINV01	Update_WCS_OrderStatus_20.dtd

Customer Create/Update Message (DEBMAS05)

IDOC segment names begin with E1 and E2 and are used interchangeably in this document and the ESQLs.

XML element	WebSphere Commerce table name	Database column	Length	Description	IDoc segment	IDoc field	Length	Field description
LogonID	USERREG	LOGONID	254	LogonID of the user with password for authentication.	E1KNA1M	KUNNR	10	Customer Number
Password	USERREG	LOGONPASSWORD	128	Password with the LogonID for authentication.	E1KNA1M	SORTL	10	Sort field (The password must contain a numeric character)
VerifyPassword	N/A	N/A	128	Password confirmation	E1KNA1M	SORTL	10	Sort field
CustomerStatus	USERREG	STATUS	INT	This allows a user's logon ID to be disabled without being removed from the system. 1 enabled, 0 disabled.	E1KNA1M	AUFSD	2	Central blocking for customer
PasswordExpired	USERREG	PASSWORDEXPIRED	INT	0 - not expired 1- expired		Always set to 1		
AddressType	ADDRESS	ADDRESSTYPE	5	S - Shipto B- Billto SB - default		Defaulted to SB		

XML element	WebSphere Commerce table name	Database column	Length	Description	IDoc segment	IDoc field	Length	Field description
				(shipto,billto)				
Title	ADDRESS	PERSONTITLE	50	Title of the person to which this address applies.	E1KNA1M	ANRED	15	Form of address for contact person
LastName	ADDRESS	LASTNAME	128		E1KNA1M	NAME1	35	Name 1
FirstName	ADDRESS	FIRSTNAME	128		E1KNA1M	NAME2	35	First name
AddressLine	ADDRESS	ADDRESS1	50		E1KNA1M	STRAS	35	Street and house number
City	ADDRESS	CITY	128		E1KNA1M	ORT01	35	City
State	ADDRESS	STATE	128		E1KNA1M	REGIO	3	Region (State, Province, Country)
ZipCode	ADDRESS	ZIPCODE	40		E1KNA1M	PSTLZ	10	Postal code
Country	ADDRESS	COUNTRY	128		E1KNA1M	LAND1	3	Country key
Telephone	ADDRESS	PHONE1	32	First occurrence	E1KNAIM	TELF1	16	First telephone number
Telephone	ADDRESS	PHONE2	32	Second occurrence	E1KNAIM	TELF2	16	Second telephone number
Fax	ADDRESS	FAX1	32	First occurrence	E1KNA1M	TELFX	31	Fax number
Profile	USERS	PROFILETYPE	2	C (base profile data), B (business profile data)		Defaulted to “C”		

XML element	WebSphere Commerce table name	Database column	Length	Description	IDoc segment	IDoc field	Length	Field description
				data)				
PreferredCurrency	USERS	SETCURR	3	3-character alphabetic code as per ISO 4217	E1KNA1M	UWAER	5	Currency of sales figure
PreferredLanguage	USERS	LNAGUAGE_ID	INT	Preferred language	E1KNA1M	SPRAS	1	Language key
PreferredDelivery	USERPROF	PREFERREDDELIVERY	1,000	preferred delivery method	E1KNVVM	INCO2	28	INCOterms
ComapnyName	USERDEMO	COMPANYNAME	128	The company for which the user works	E1KNA1M	NAME1	35	Name 1

Order Confirmation Status Message (SISCSO01)

XML element	WebSphere Commerce table name	Database column	Length	Field description	IDoc segment	IDoc field	Length	Field description
OrderNumber	ORDSTAT	ORDER_ID	BIGINT	WebSphere Commerce order reference number.	E1CVBAK	BSTNK	20	Customer purchase order number
OrderNumber	ORDSTAT	OSMORDER	BIGINT	Order number generated by backend system.	E1CVBAK	VBELN	10	Sales document
Placed Date	ORDSTAT	OSPLTIME	TIMESTAMP	Order placed timestamp.	E1CVBAK	AUDAT	8	Document date

XML element	WebSphere Commerce table name	Database column	Length	Field description	IDoc segment	IDoc field	Length	Field description
TotalPriceInfo (currency attribute)	ORDSTAT	OSCPCUR	10	Currency in which the price is expressed	E1CVBAK	WAERK	5	Document currency
TotalNetPrice	ORDSTAT	OSPRTOT	20	Total product price for the order.	E1CVBAK	NETWR	8	Net value of order in document currency (Sum of KZWI1 of E1CVBAP segments)
TotalTaxPrice	ORDSTAT	OSTXTOT	20	Total tax for the order item	E1CVBAP		7	Subtotal 5 from pricing procedure for condition (Sum of KZWI5 of E1CVBAP segments)
TotalShippingPrice	ORDSTAT	OSSHTOT	20	Total shipping charge for the order item	E1CVBAP		7	Subtotal 4 from pricing procedure for condition Sum of KZWI4 of E1CVBAP segments)
TotalTaxOnShippingPrice	ORDISTAT	OSSHTXTOT	20	Total tax on shipping charges for the order item	E1CVBAP		7	Subtotal 6 from pricing procedure for condition Sum of KZWI6 of E1CVBAP segments)
Status	ORDSTAT	OSSTATUS	32	Default value of ‘C’ is set by the command				

XML element	WebSphere Commerce table name	Database column	Length	Field description	IDoc segment	IDoc field	Length	Field description
ShippingInfo (Ship Condition attribute)	ORDSTAT	OSSCOND	2	SC =ShipComplete SP = ShipPartial	E1CVBAK	AUTLF	1	Complete delivery indicator for each sales order
RequestShipDate	ORDSTAT	OSRSTIME	TIMESTAMP	Requested shipping timestamp.	E1CVBAK	VDATU	8	Proposed schedule line date
ItemNumber	ORDISTAT	ORDERITMES_ID	BIGINT	WebSphere Commerce order item reference number.	E1CVBAP	POSEX	6	Item number of the customer purchase order
ItemNumber	ORDISTAT	OIMTEM	30	Order item number generated by backend system	E1CVBAP	POSNR	6	Sales document item
ProductNumberByMerchant	ORDISTAT	PARTNUMBER	64	Part number or SKU	E1CVBAP	MATNR	18	Material
RequestedQuantity	ORDISTAT	OIQTREQUEST	INT	Quantity of items requested	E1CVBAP	KWMENG	8	Cumulative order quantity in sales units
ConfirmedQuantity	ORDISTAT	OIQTCONFIRM	INT	Quantity of items confirmed	E1CVBAP	KBMENG	8	Cumulative confirmed quantity in sales unit
ItemUnitPrice	ORDISTAT	OIUNPRC	20	Unit price of the order item	E1CVBAP	NETPR	6	Net price
TotalPriceInfo (currency attribute)	ORDISTAT	OICPCUR	10	Currency in which the price is expressed	E1CVBAP	WAERK	5	Document currency

XML element	WebSphere Commerce table name	Database column	Length	Field description	IDoc segment	IDoc field	Length	Field description
TotalNetPrice	ORDISTAT	OIPRTOT	20	Total product price for the order item	E1CVBAP	KZWI1	8	Net value of the order item in document currency
TotalTaxPrice	ORDISTAT	OITXTOT	20	Total tax for the order item	E1CVBAP	KZWI5	7	Subtotal 5 from pricing procedure for condition
TotalShippingPrice	ORDISTAT	OISHTOT	20	Total shipping charge for the order item	E1CVBAP	KZWI4	7	Subtotal 4 from pricing procedure for condition
TotalTaxOnShippingPrice	ORDISTAT	OISHTXTOT	20	Total tax on shipping charges for the order item	E1CVBAP	KZWI6	7	Subtotal 6 from pricing procedure for condition
Status	ORDISTAT	OISTATUS	32	Default 'C' set by the command				
PlacedDate	ORDISTAT	OIPLTIME	TIMESTAMP	Order item placed timestamp	E1CVBAK	AUDAT	8	Document date
ShippingInfo ShipCondition (attr)	ORDISTAT	OISCOND	2	SC = ShipComplete SP = ShipPartial	E1CVBAP	KZTLF	1	Partial delivery at item level
ScheduledShipDate	ORDISTAT	OISSTIME	TIMESTAMP	Order item scheduled timestamp	E1CVBEP	EDATU	8	Delivery date for the schedule line

Order Delivery Status Message (SISDEL01)

Output format: SAP2WCS_OrdShpStatusMsg					**Input format: SAP.IC.SISDEL01**			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc segment**	**IDoc field**	**Length**	**Field description**
OrderNumberByWCBE	ORDSTAT	ORDERS_ID	19	WCBE order reference number	E2CLIKP	LIFEX	35	Order Number
OrderNumberByBackend	ORDSTAT	OSMORDER	30	Order number generated by the backend system	E2CLIPS	VGBEL	10	The sales order to which the delivery refers.
TotalNetPrice	ORDSTAT	OSPRTOT	21	Total product price for the order				
TotalTaxPrice	ORDSTAT	OSTXTOT	21	Total tax for the order				
TotalShippingPrice	ORDSTAT	OSSHTOT	21	Total shipping charges for the order				
TotalTaxOnShippingPrice	ORDSTAT	OSSHTXTOT	21	Total tax on shipping charges for the order				
Status	ORDSTAT	OSSTATUS	32	Status of the order. Defaults to S =Shipped				
PlacedDate	ORDSTAT	OSPLTIME	25	OrderPlaced Date				

Output format: SAP2WCS_OrdShpStatusMsg					**Input format: SAP.IC.SISDEL01**			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc segment**	**IDoc field**	**Length**	**Field description**
RequestedShipDate	ORDSTAT	OSRSTIME	25	Requested shipping time stamp				
ScheduledShipDate	ORDSTAT	OSSSTIME	25	Scheduled shipping time stamp				
ActualShipDate	ORDSTAT	OSASTIME	25	Actual shipping time stamp	E2CLIKP	wadat_ist		Actual goods movement date
ItemNumberByWCBE								
ItemNumberByBackend	ORDISTAT	OIMITEM	30	Order item number generated by the backend system	E2CLIPS	VGPOS	6	The number of the sales order item to which the delivery item refers
ProductNumberByMerchant	ORDISTAT	PARTNUMBER	19	Part number or SKU	E2CLIPS	MATNR	18	Key uniquely identifying the material
RequestedQunatity	ORDISTAT	OIQTREQUEST	10	Quantity of items requested	E2CLIP2	APKWMENG	18	Cumulative order quantity in sales unit
ConfirmedQuantity	ORDISTAT	OIQTCONFIRM	10	Quantity of items confirmed	E2CLIP2	APKBMENG	18	Cumulative confirmed quantity in sales

Output format: SAP2WCS_OrdShpStatusMsg					**Input format: SAP.IC.SISDEL01**			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc segment**	**IDoc field**	**Length**	**Field description**
								unit
ShippedQuantity	ORDISTAT	OIQTSHIP	10	Quantity of items shipped	E2CLIPS	LFIMG	15	Actual quantity delivered in sales unit
TotalPriceInfo attribute Currency	ORDISTAT	OICPCUR	10	Currency in which the price is expressed	E2CLIP2	APWAERK	5	Currency that applies to the document
TotalNetPrice	ORDISTAT	OIPRTOT	21	Total product price for the order item	E2CLIP2	APNETWR	18	Net value in document currency
TotalTaxPrice	ORDISTAT	OITXTOT	21	Total tax for the order item				
TotalShippingPrice	ORDISTAT	OISHTOT	21	Total shipping charges for the order item				
TotalTaxOnShippingPrice	ORDISTAT	OISHTXTOT	21	Total tax on shipping charges for the order item				
Status	ORDSTAT	OSSTATUS	32	Defaulted to I = Invoiced				
PlacedDate	ORDSTAT	OIPLTIME	25					
Shipping Info Attribute Shipping condition	ORDISTAT	OISCOND	2	Code indicating if partial shipment of the order is	E2CLIPS	KZTLF	1	Partial delivery at item level

Output format: SAP2WCS_OrdShpStatusMsg					Input format: SAP.IC.SISDEL01			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc segment**	**IDoc field**	**Length**	**Field description**
				the order is accepted				
RequestShipDate	ORDSTAT	OSRSTIME	25	Requested shipping time stamp				
ScheduledShipDate	ORDSTAT	OSSSTIME	25	Scheduled shipping time stamp	E2CVBEL	EDATU	8	Delivery date for the schedule line
ActualShipDate	ORDSTAT	OSASTIME	25	Actual shipping time stamp	E2CLIKP	WADAT_IST	8	Actual goods issue date

Order Invoice Status Messages (SISINV01)

Output format: SAP2WCS_OrdInvStatusMsg					Input format: SAP.IC.SISINV01			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc Segment**	**IDoc Field**	**Length**	**Field description**
OrderNumberByWCBE	ORDSTAT	ORDERS_ID	30	WebSphere Commerce order reference number	E2CVBRK	BSTNK_VF	35	Customer purchase order number
OrderNumberByBackend	ORDSTAT	OSMORDER	19	Order number generated by the backend system	E2CVBRP	AUBEL	10	The number that uniquely identifies the sales document
TotalNetPrice	ORDSTAT	OSPRTOT		Total product price for the order				
TotalTaxPrice	ORDSTAT	OSTXTOT		Total tax for the order				
TotalShippingPrice	ORDSTAT	OSSHTOT		Total shipping charges for the order				
TotalTaxOnShippingPrice	ORDSTAT	OSSHTXTOT		Total tax on shipping charges for the order				
Status	ORDSTAT	OSSTATUS		Defaulted to				

Output format: SAP2WCS_OrdInvStatusMsg					Input format: SAP.IC.SISINV01			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc Segment**	**IDoc Field**	**Length**	**Field description**
				I = Invoiced				
PlacedDate	ORDSTAT	OSPLTIME		Order placed date				
RequestShipDate	ORDSTAT	OSRSTIME		Requested shipping time stamp				
ScheduledShipDate	ORDSTAT	OSSSTIME		Scheduled shipping time stamp				
ActualShipDate	ORDSTAT	OSASTIME		Actual shipping time stamp				
InvoiceDate	ORDSTAT	OSINVTIME	25	Invoice time stamp for order item	E2CVBRK	FKDAT	8	The date on which the billing is processed and booked for accounting
InvoiceValue	ORDSTAT	OSINVVAL	21	Net value of the invoice for an order item	E2CVBRK	NETWR	17	Net value of the document item
ItemNumberByWCS								
ItemNumberByBackend	ORDISTAT	OIMITEM	30	Order item number generated by the backend system	E2CVBRP	POSNR	6	The number that uniquely identifies the item in the

Output format: SAP2WCS_OrdInvStatusMsg					**Input format: SAP.IC.SISINV01**			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc Segment**	**IDoc Field**	**Length**	**Field description**
				backend system				billing document.
ProductNumberByMerchant	ORDISTAT	PARTNUMBER	64	Part number or SKU	E2CVBRP	MATNR	18	Alphanumeric key uniquely identifying the material
RequestedQunatity	ORDISTAT	OIQTREQUEST						
ConfirmedQuantity	ORDISTAT	OIQTCONFIRM						
ShippedQuantity	ORDISTAT	OIQTSHIP						
TotalPriceInfo Attribute Currency	ORDISTAT	OICPCUR	10	Currency in which the price is expressed	E2CVBRK	WAERK	5	Document Currency
TotalNetPrice	ORDISTAT	OIPRTOT	21	Total product price for the order item	E2CVBRP	NETWR	18	Net value of the billing item in the document currency
TotalTaxPrice	ORDISTAT	OITXTOT	21	Total tax for the order item	E2CVBRP	KZWI5	15	Subtotal for the pricing condition
TotalShippingPrice	ORDISTAT	OISHTOT	21	Total shipping charges for the order item	E2CVBRP	KZWI4	15	Subtotal for the pricing condition
TotalTaxOnShippingPrice	ORDISTAT	OISHTXTOT	21	Total tax on	E2CVBRP	KZWI6	15	Subtotal for the

Output format: SAP2WCS_OrdInvStatusMsg					Input format: SAP.IC.SISINV01			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc Segment**	**IDoc Field**	**Length**	**Field description**
				shipping charges for the order item				pricing condition
Status		OISTATUS		Defaulted to I=Invoiced				
PlacedDate	ORDISTAT	OIPLTIME						
RequestShipDate	ORDISTAT	OSRSTIME						
ScheduledShipDate	ORDISTAT	OSSSTIME						
ActualShipDate	ORDISTAT	OSASTIME						
InvoiceDate	ORDISTAT	OIINVTIME	25	Invoice time stamp for order item.	E2CVBRK	FKDAT	8	The date on which the billing is processed and booked for accounting
InvoiceValue	ORDISTAT	OIINVVAL	21	Net value of the invoice for an order item	E2CVBRP	NETWR	18	The net value of the billing item

Order Create Message (ORDERS05)

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
OrderNumberByBuyer							
OrderNumberByMerchant	ORDERS	ORMORDER	Unique order reference number generated by the merchant				
OrderNumberByNC	ORDERS	ORDERS_ID	Unique order reference number, internally generated. This is a primary key	E1EDK02	BELNR QUALF=001 (constant to indicate purchase order number) QUALF=018 (constant to indicate customer order number)	35	IDoc document Number
DateTimeReference							
PlacedDate	ORDERS	TIMEPLACED	Date the order was placed, in the format YYYYMMDD.	E1EDK03	DATUM	8	IDoc: Date

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
PlacedTime	ORDERS	TIMEPLACED	Time the order was placed, in the format HHMMSS.	E1EDK03	UZEIT	6	IDoc: Time
LastUpdateDate	ORDERS	LASTUPDATE	The time this Order was most recently updated				
TotalPriceInfo	ORDERS	TOTALPRODUCT	The sum of ORDERITEMS.TOTALPRODUCT for the OrderItems in the Order.				
Currency (Attribute)	ORDERS	CURRENCY	Currency in which the price is expressed. The format of the currency must adhere to ISO 4217 standards	E1EDK01	CURCY, HWAER	3	Currency
TotalNetPrice	ORDERS	TOTALPRODUCT	Total product price for the order			15	Price (Net)
TaxInfo							

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
MonetaryAmount	ORDERS	TOTALTAX	Total sales tax for the order	E1EDK04	MWSBT	18	Value Added tax amount
Currency(Attribute)	ORDERS	CURRENCY	Currency				
TaxType							
Percentage							
TaxExemptionStatusType							
TaxExemptionNumber							
TaxJurisdictionCode							
TaxJurisdictionCodeType							
TotalShippingPrice	ORDERS	TOTALSHIPPING	Total shipping charges for the order				
TotalTaxOnShippingPrice	ORDERS	TOTALTAXSHIPPING	Total tax on shipping charges for the order				
Instruction	NA						
ShipStatus	ORDERS	STATUS					

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
BillToInfo BuyOrgInfo (Optional) ShiptoInfo (Optional) RequisitionerInfo	NA			E1EDKA1	PARVW='RE' PARVW='EK' PARVW='WE' PARVW='AG'	3	Partner Function
OrgName	NA						
AddressLine	ADDRESS	ADDRESS1	Bill to person's address line 1	E1EDKA1	STRAS	35	Street and house number 1
AddressLine	ADDRESS	ADDRESS2	Bill to person's address line 2	E1EDKA1	STRS2	35	Street and house number 2
AddressLine	ADDRESS	ADDRESS3	Bill to person's address line 3	E1EDKA1			
City	ADDRESS	CITY	Bill to person's city name.	E1EDKA1	ORT01	35	City
State	ADDRESS	STATE	Bill to person's state, province, or equivalent, abbreviated.	E1EDKA1	REGIO	3	Region
Zip	ADDRESS	ZIPCODE	Bill to person's zip code or equivalent.	E1EDKA1	PSTLZ	9	Postal code
Country	ADDRESS	COUNTRY	Bill to person's country.	E1EDKA1	LAND1	3	Country key

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
ContactPersonName	NA	NA					
Title	NA			E1EDKA1	TITLE	15	Title
FullName	NA						
LastName	ADDRESS	LASTNAME	Bill to person's last name	E1EDKA1	NAME1	35	Name 1
FirstName	ADDRESS	FIRSTNAME	Bill to person's first name	E1EDKA1	NAME3	35	Name 3
MiddleName	ADDRESS	MIDDLENAME	Bill to person's middle name	E1EDKA1	NAME2	35	Name 2
AlternateName	ADDRESS	NICKNAME					
ContactInfo	NA						
Telephone	ADDRESS	PHONE1	Bill to person's primary phone number	E1EDKA1	TELF1	25	Telephone number 1 of contact person
Telephone	ADDRESS	PHONE2	Bill to person's secondary phone number	E1EDKA1	TELF2	25	Telephone number 2 of contact person
Email	ADDRESS	EMAIL1	Bill to person's primary e-mail or URL address				
Email	ADDRESS	EMAIL2	Bill to person's secondary e-mail				

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
			or URL address				
Fax	ADDRESS	FAX1	Bill to person's fax number	E1EDKA1	TELFX	25	Fax number
MerchantInfo	NA						
OrgName	STOREENDTS	DISPLAYNAME	Merchant's company name.				
OrgID	NA						
Type(attribute)	ORDERS	STOREENT_ID	Merchant reference number.	E1EDK14	QUALF=006 (DIVISION INFORMATION) ORGID QUALF=007 (DISTRIBUTION INFORMATION) ORGID QUALF=008 (SALES ORGANISATION)	3	IDoc qualifier organization

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
					ORGID QUALF=012 (Order Type) ORGID		
Address	NA						
AddressLine	STADDRESS	ADDRESS1	Merchant's company address line 1				
AddressLine	STADDRESS	ADDRESS2	Merchant's company address line 2				
AddressLine	STADDRESS	ADDRESS3	Merchant's company address line 3				
City	STADDRESS	CITY	Merchant's company city name				
State	STADDRESS	STATE	Merchant's company state, province, or equivalent, abbreviated				
Zip	STADDRESS	ZIPCODE	Merchant's company zip				

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
			code or equivalent				
Country	STADDRESS	COUNTRY	Merchant's company country				
URL	NA						
Telephone	STADDRESS	PHONE1	Merchant's company phone number				
ContactPersonName	NA						
Title	STADDRESS	PERSONTITLE	Merchant contact's title				
FullName	NA						
LastName	STADDRESS	LASTNAME	Merchant contact's last name				
FirstName	STADDRESS	FIRSTNAME	Merchant contact's first name				
MiddleName	STADDRESS	MIDDLENAME	Merchant contact's middle name				
AlternateName	NA						
ContactInfo	NA						

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
Telephone	STADDRESS	PHONE1	Merchant contact's primary phone number				
Telephone	STADDRESS	PHONE2	Merchant contact's secondary phone number				
Email	STADDRESS	EMAIL1	Merchant contact's primary e-mail or URL address				
Email	STADDRESS	EMAIL2	Merchant contact's secondary e-mail or URL address				
Fax	STADDRESS	FAX1					
BuyOrgInfo	Na			E1EDKA1	PARVW='RE' (BUYER)	3	Partner function
OrgName	ADDRESS	ORGNAME					
OrgID						17	Partner number
Type (attribute)							
Address							
AddressLine				E1EDKA1	STRAS	35	Street and house number 1

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
							number 1
AddressLine				E1EDKA1	STRS2	35	Street and house number 2
AddressLine							
City				E1EDKA1	ORT01	35	City
State				E1EDKA1	REGIO	3	Region
Zip				E1EDKA1	PSTLZ	9	Postal code
Country				E1EDKA1	LAND1	3	Country key
URL							
ContactInfo							
Telephone				E1EDKA1	TELF1	25	Telephone number 1 of contact person
Telephone				E1EDKA1	TELF2	25	Telephone number 2 of contact person
Email							
Email							
Fax				E1EDKA1	TELFX	25	Fax number
RequisitionerInfo				E1EDPA1	PARVW='AG'	3	Partner function

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
					(Sold-To Party)		
RequisitionerID	ORDERS	MEMBER_ID					
Type(Attribute)							
RequisitionerID	USERREG	LOGONID		E1EDKA1	PARTN	17	Partner number
Type(attribute)							
RequisitionerGroup							
Address							
AddressLine	ADDRESS	ADDRESS1		E1EDKA1	STRAS		Street and house number 1
AddressLine	ADDRESS	ADDRESS2		E1EDKA1	STRS2		Street and house number 2
AddressLine	ADDRESS	ADDRESS3					
City	ADDRESS	CITY		E1EDKA1	ORT01		City
State	ADDRESS	STATE		E1EDKA1	REGIO		Region
Zip	ADDRESS	ZIPCODE		E1EDKA1	PSTLZ		Postal code
Country	ADDRESS	COUNTRY		E1EDKA1	LAND1		Country key
ContactPersonName	ADDRESS						

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
Title	ADDRESS	PERSONTITLE		E1EDKA1	TITLE		Title
FullName	ADDRESS						
LastName	ADDRESS	LASTNAME		E1EDKA1	NAME1		Name 1
FirstName	ADDRESS	FIRSTNAME		E1EDKA1	NAME3		Name 2
MiddleName	ADDRESS	MIDDLENAME		E1EDKA1	NAME2		Name 3
AlternateName							
ContactInfo							
Telephone	ADDRESS	PHONE1		E1EDKA1	TELF1		Telephone number 1 of contact person
Telephone	ADDRESS	PHONE2		E1EDKA1	TELF2		Telephone number 2 of contact person
Email	ADDRESS	EMAIL1					
Email	ADDRESS	EMAIL2					
Fax	ADDRESS	FAX1		E1EDKA1	TELFX		Fax number
ShipDateReference	Na						
RequestedShipDate	Na						
ScheduledShipDate	Na						

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
ActualShipDate	Na						
PcardInfo	Na						
MonetaryAmount	ORDPAYMTHD	MAXAMOUNT					
Currency(attribute)	ORDERS	CURRENCY					
CardType	ORDPAYMTHD	PAYMETHOD					
CardNumber	ORDPAYMTHD	PAYDEVICE					
ExpirationDate	ORDPAYMTHD	ENDDATE					
IssueDate	ORDPAYMTHD	STARTDATE					
CreditAuthorizationNumber							
CustomerReferenceNumber							
ShippingCarrierInfo							
Carrier	SHIPMODE	CARRIER	Carrier identifier	E2EDK17	LKOND when INCOTERMS QUALF = ‘001’	3	Delivery condition code
Method	SHIPMODE	CODE	A merchant assigned code	E2EDK17	LKTEXT when INCOTERMS QUALF = ‘002’	70	Delivery condition text

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
					QUALF = ‘002’		
BuyOrgAccountingDetail							
Percentage							
MonetaryAmount							
Currency(Attribute)							
BudgetCode							
Description							
CalculationCode							
OrderCustomerField	ORDERS	Field1					
OrderCustomerField	ORDERS	Field2					
OrderCustomerField	ORDERS	Field3					
UserData							
UserDataField name Attribute							
UserDataField							
ReportPOItem							
ItemLineNumber	Na						

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
ItemNumberByNC	ORDERITEMS	ORDERITEMS_ID					
ProductNumberByBuyer	Na						
ProductNumberByMechant	CATENTRY	PARTNUMBER		E1EDP19	IDTNR	35	IDoc material ID
ManufactureName	CATENTRY	MFNAME	The name of the manufacturer of this CatalogEntry	E1EDP19	MFRNR	10	Manufacturer number
ManufactureURL							
ManufacturePartNumber	CATENTRY	MFPARTNUMBER	The part number used by the manufacturer to identify this CatalogEntry	E1EDP19	MFRPN	42	Manufacturer part number
ItemUnitPrice	ORDERITEMS	PRICE		E1EDP01	NETWR	18	Item value (Net)
Currency (attribute)	ORDERITEMS	CURRENCY		E1EDP01	CURCY	3	Currency
TaxInfo				E1EDP04			
MonetaryAmount	ORDERITEMS	TAXAMOUNT	The total sales taxes associated with this OrderItem	E1EDP04	MWSBT	18	Value Added tax amount
Currency(attribute)							

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
TaxType							
Percentage							
TaxExemptionStatusType							
TaxExemptionNumber							
TaxJurisdictionCode							
TaxJurisdictionCodeType							
ItemProductQuantity	ORDERITEMS	QUANTITY		E1EDP01	MENGE , BMNG2	15	Quantity, quantity in price unit
UnitOfMeasure							
Classification				E1EDP01	PSTYP		
ItemProductShortDescription	CATENTDESC	SHORTDESCRIPTION		E1EDP01	ABGRT	40	Description
Instruction							
ShipToInfo				E1EDPA1	PARVW='WE' (Ship-To party)	3	Partner function
OfficeAddressLine							
ContactPersonName							
Title							

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
FullName							
LastName	ADDRESS	LASTNAME		E1EDPA1	NAME1	35	Name 1
FirstName	ADDRESS	FIRSTNAME		E1EDPA1	NAME3	35	Name 3
MiddleName	ADDRESS	MIDDLENAME		E1EDPA1	NAME2	35	Name 2
AlternateName	ADDRESS	NICKNAME					
Address							
AddressLine	ADDRESS	ADDRESS1		E1EDPA1	STRAS	35	
AddressLine	ADDRESS	ADDRESS2		E1EDPA1	STRS2	35	
AddressLine	ADDRESS	ADDRESS3		E1EDPA1			
City	ADDRESS	CITY		E1EDPA1	ORT01	35	City
State	ADDRESS	STATE		E1EDPA1	REGIO	3	Region
Zip	ADDRESS	ZIPCODE		E1EDPA1	PSTLZ	9	Postal code
Country	ADDRESS	COUNTRY		E1EDPA1	LAND1	3	Country
ContactInfo							
Telephone	ADDRESS	PHONE1		E1EDPA1	TELF1	25	Telephone number 1 of contact person
Telephone	ADDRESS	PHONE2		E1EDPA1	TELF2	25	Telephone number 2 of contact person

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
							2 of contact person
Email	ADDRESS	EMAIL1					
Email	ADDRESS	EMAIL2					
Fax	ADDRESS	FAX1		E1EDPA1	TELFX		Fax number
Comment	ORDERITEMS	COMMENTS					
ShippingCarrierInfo				E1EDPA1 Also in E1EDP17	PARVW='SP' (carrier information)	3	Partner function
Carrier	SHIPMODE	CARRIER		E1EDPA1 Also in E1EDP17	PARTN LKOND for INCOTERMS QUALF = '001'	3	Partner number Delivery condition code
Method	SHIPMODE	CODE		E1EDP17	LKTEXT for INCOTERMS QUALF = '002'	70	Delivery condition text
ShipStatus	ORDERITEMS	STATUS					
DateTimeReference				E1EDP03	IDDAT='022' (PURCHASE ORDER DATE)		

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
PlacedDate	ORDERITEMS	LASTCREATE		E1EDP03	DATUM	8	Date
PlacedTime	ORDERITEMS	LASTCREATE		E1EDP03	UZEIT	6	Time
LastUpdateDate	ORDERITEMS	LASTUPDATE					
LastUpdateTime	ORDERITEMS	LASTUPDATE					
ProductMeasurement							
ProductWeight							
UnitOfMeasure(attribute)							
ProductDimension							
UnitOfMeasure(attribute)							
ProductLength							
ProductWidth							
ProductHeight							
BuyOrgAccountingDetail							
Percentage							
MonetaryAmount							

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
Currency (Attribute)							
BudgetCode							
Description							
CalculationCode							
ServiceAllowanceCharge							
AllowanceChargeCode							
Percentage							
MonetaryAmount							
Currency							
Description							
CalculationCode							
ItemShippingSchedule							
Quantity							
ShipDateReference							
RequestedShipDate							
ScheduledShipDate							

XML element	**WebSphere Commerce table name**	**Database column**	**Description**	**IDoc segment**	**IDoc field**	**Length**	**Field description**
ActualShipDate							
ItemCustomerField	ORDERITEMS	FIELD1					
ItemCustomerField	ORDERITEMS	FIELD2					
UserDataField name							
UserDataField							

Product Price Update Message (COND_A02)

Output format: SAP2WCS_ProPriceUpdt Msg					Input format: SAP.IC.COND_A01			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc segment**	**IDoc field**	**Length**	**Field description**
ProductNumberByMerchant	CATENTRY	PARTNUMBER	64	Reference number to identify the part number for the catalog entry	E2KNOP	MATNR	18	Alphanumeric key uniquely identifying the material
MerchantID	CATENTRY	MEMBER_ID	64	Reference number to identify the owner of the catalog entry item	E2KNOP	Substitute Based on the value VKORG. WS01 with 10001	4	An organizational unit responsible for the sale of certain products
Precedence	OFFER	PRECEDENCE	8	When more than one Offer is effective at a particular time, the one with the highest precedence is used		Default the value to 0 so that the existing price gets updated		
TradingPositionContainerID	TRADEPOSCN	TRADEPOSCN_ID	8	TradingPosition ContainerID		Get the value of TradingPositionCon		

Output format: SAP2WCS_ProPriceUpdt Msg					Input format: SAP.IC.COND_A01			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc segment**	**IDoc field**	**Length**	**Field description**
ainerID		ID		ContainerID		tainerID for the store and member group		
Currency	LISTPRICE/OFFERPRICE	CURRENCY	3	The currency in which the price is expressed	E2KNOP	KONWA	5	Rate unit
ItemUnitPrice	LISTPRICE/OFFERPRICE	LISTPRICE/PRICE	21	The amount of the listprice	E2KONP	KEBTR	16	Rate
Published	OFFER	PUBLISHED	4	Whether or not offer is published		Default to 1, which means published		

Product Quantity Update Message (INVCON01)

Output format: SAP2WCS_ProInvUpdt Msg					**Input format: SAP.IC.INVCON01**			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc segment**	**IDoc field**	**Length**	**Field description**
ProductNumberBy Merchant								
ProductSKU	INVENTORY	CATENTRY_ID	64	Internal reference value as assigned by WebSphere Commerce system	E2ICSL0	MATNR	18	Alphanumeric key uniquely identifying the material
MerchantID	INVENTORY	STORE_ID	10	A unique number to identify the merchant's store				Substitute with STORE_ID
Quantity	INVENTORY	QUANTITY	16	The quantity as available by the merchant	E2ICSL0	LABST	18	Unrestricted-use valuated stock
FulfillmentCenterID	INVENTORY	FFMCENTER_ID	10	The fulfillment center	E2ICSL0	Substitute for the value of WERKS and LGORT. E.g. WERKS = WSAL	WERKS (4),LGORT(4)	WERKS Key uniquely identifying the plant, LGORT Storage location

Output format: SAP2WCS_ProInvUpdt Msg					Input format: SAP.IC.INVCON01			
XML element	**WebSphere Commerce table name**	**Database column**	**Length**	**Field description**	**IDoc segment**	**IDoc field**	**Length**	**Field description**
						LGORT = WSL2 Substitute with (10001)		

Notices

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

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

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

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

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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

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

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

IBM Canada Ltd.
Office of the Lab Director

8200 Warden Avenue
Markham, Ontario
L6G 1C7
Canada

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

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

This document may contain information about other companies' products, including references to such companies' Internet sites. IBM has no responsibility for the accuracy, completeness, or use of such information.

Trademarks

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

IBM

DB2

MQSeries

WebSphere

DB2 Universal Database™

Microsoft®, Windows, and Windows NT are trademarks or registered trademarks of Microsoft Corporation.

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

SAP and R/3 are trademarks or registered trademarks of SAP AG.

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

Printed in U.S.A