Integration Guide

for SAP® R/3® using WebSphere MQ Integrator

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

Second Edition (December 2002)

IBM welcomes your comments. You can send your comments by any one of the following methods:

1. Electronically to the E-mail address listed below. Be sure to include your entire network address if you wish a reply.

   Internet:torrcf@ca.ibm.com

2. By regular mail to the following address:

   IBM Canada Ltd. Laboratory
   B3/KB7/8200/MKM
   8200 Warden Avenue
   Markham, Ontario, Canada L6G 1C7

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.

This edition applies to version 5.4 of IBM WebSphere Commerce (Program 5724 - A18) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Introduction

About this book

This document contains information about the integration of WebSphere Commerce Version 5.4 with the SAP R/3 4.6 Enterprise Resource Planning (ERP) system, using WebSphere MQSeries® and WebSphere MQ Integrator.

Who should read this book?

This book is for anyone who wishes to integrate WebSphere Commerce with the SAP R/3 backend system. It will assist developers and engagement teams in installing this reference application. Additionally, demonstrators or marketing personnel who want to demonstrate this functionality can use this book.

The samples provided are for WebSphere Commerce 5.4, Business Edition. However, the same principal and pattern can apply to WebSphere Commerce 5.4, Professional Edition.

Note: Knowledge of WebSphere Commerce and SAP R/3 Enterprise system is assumed.

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, R/2, 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 sell-side solution, 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.

In the scenario described below, the WebSphere Commerce server sell-side 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 of 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 the 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 Create**: 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 Massloader 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

- Provides customers with access to Web site functions such as browsing the catalog, placing orders, and making online payments.
- Synchronizes product and customer profile information.

- Creates orders in WebSphere Commerce and sends the orders to SAP for order processing and fulfilment.
- Customers can check the status of orders, online.
- The broker-based integration maximizes both the isolation of business processes from the external organization, and the flexibility to change the processes, and the applications that implement them.
- You can create online catalogs in WebSphere Commerce for SAP materials.

WebSphere MQ Integrator - 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 runtime, 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 configuration of the message broker and the contents of the message. 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-4.ibm.com/software/webservers/commerce/wc_be/
- SAP R/3 documentation http:/help.sap.com
- WebSphere MQSeries documentation http://www-3.ibm.com/software/ts/mqseries/
- MQSeries link for R/3 documentation http://www-3.ibm.com/software/ts/mqseries/
- WebSphere MQ Integrator documentation http://www-3.ibm.com/software/ts/mqseries/integrator/

Chapter 2. Pre-requisites

This section covers the software components used in this application. All the software components are installed and configured on Windows NT and Windows 2000 operating environment.

WebSphere Commerce version 5.4, Business Edition

WebSphere Commerce version 5.4, 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:

- The WebSphere MQSeries Adapter enables integration by processing inbound messages coming from back-end systems
- The outbound messaging system allows you to send outbound messages to back-end systems

Note: The database server used in this reference application is DB2. The instructions provided in the Installation and Configuration chapter assume that DB2 server is used.

WebSphere MQSeries version 5.2

This component is used as the transport middleware to communicate with various back-end systems, including SAP. The WebSphere Commerce version 5.4, Business Edition requires the middleware to support Java Messaging Service (JMS) APIs. The WebSphere Commerce messaging system uses the Common Connector Framework (CCF JMS connector) to connect to transport systems. WebSphere MQSeries allows you to define the queue manager and various queues that WebSphere Commerce will utilize to read inbound messages and send outbound messages.

The MA88 Product Extension Pack

This pack contains the IBM MQSeries classes for Java and Java Messaging Service. It consists of the JMS API that the WebSphere MQSeries Adapter uses to communicate with WebSphere MQSeries. You must create the JMS QueueConnectionFactory and JMS queues that map to the corresponding physical WebSphere MQSeries objects. This allows the WebSphere MQSeries Adapter to access WebSphere MQSeries entities through JMS.

WebSphere MQ Integrator (WMQI) version 2.1

The WebSphere MQ Integrator is the transformation engine. The various components required in WebSphere MQ Integrator 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 WebSphere MQ Integrator parser for representing incoming data to WebSphere MQ Integrator in a format that the WebSphere MQ Integrator compute node can manipulate. This parser plugin is a prerequisite to process the SAP IDoc messages.

WebSphere MQ Integrator allows data mapping for different formats into IDoc streams. Refer to the `readme.txt` and then apply the appropriate CSD (Corrective Service Diskette) to WMQI. For more information on CSD refer to http://www.ibm.com/software/ts/mqseries/

MQSeries link for R/3 Adapter 1.2

This adapter provides the WebSphere MQSeries 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 to be used by external applications and accepts IDocs from other applications to be processed by the R/3 system.

Software versions supported

The following are the versions of the software supported:

- WebSphere Commerce version 5.4, Business Edition
- SAP R/3 version 4.6C
- WebSphere MQSeries version 5.2 with MA88 extension
- WebSphere MQ Integrator version 2.1 with IDoc parser
- MQSeries link for R/3 version 1.2

Chapter 3. Sample topology

This diagram shows a possible topology recommended for the components:

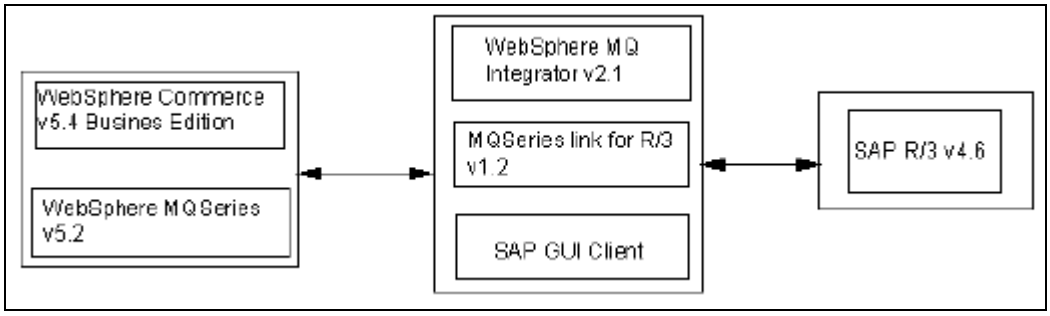

Figure 1: Sample topology

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 client is not a necessary component in the system.
- WebSphere Commerce Business Edition, WebSphere MQSeries, and WMQI can exist on different machines.

Note: The default configuration discussed in the following sections has WebSphere Commerce Business Edition, WebSphere MQ Integrator, WebSphere MQSeries and the MQSeries link for R/3 on one server and SAP R/3 on another server. The SAP GUI client is used to configure the R/3 system.

Chapter 4. Message flow

WebSphere Commerce and SAP interact through messages. These messages are passed through the messaging subsystem provided by IBM WebSphere MQSeries. Since message formats are different for WebSphere Commerce and SAP, the WebSphere MQ Integrator 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 is done for the order 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 C: Mapping information. If you modify the mapping of fields to suit your requirements, then change the ESQL code for the modified mappings.

Assumptions

The following assumptions are made when using this reference application:

- 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 "C" (customer) in ESQL during transformation for customer create and update messages.
- Preferred Language values (DEBMAS05) in WebSphere MQ Integrator 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.

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.

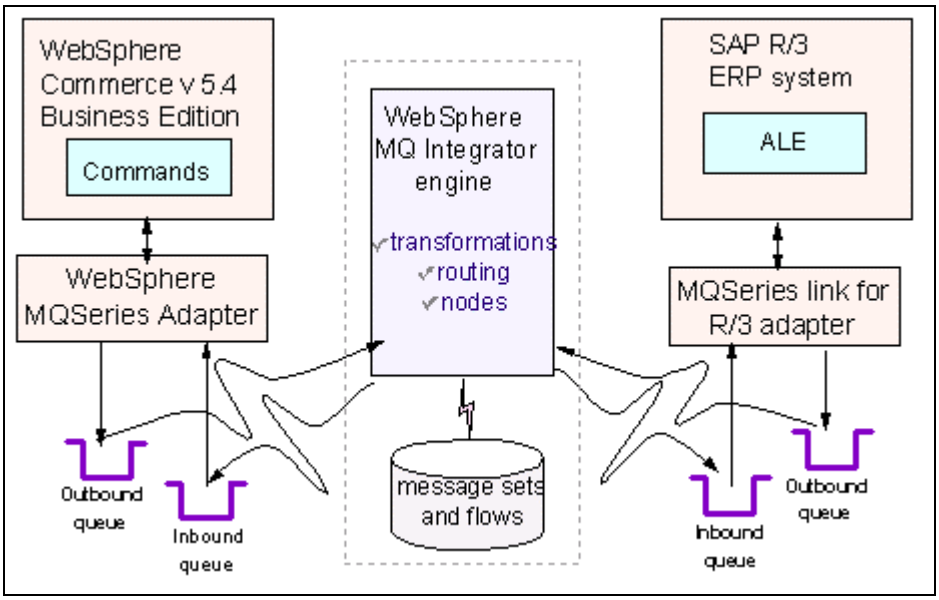

Figure 2: Message flow from and to WebSphere Commerce

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 an 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 R/3 link adapter's outbound server 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. The WMQI or 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 WebSphere MQ Integrator decides on the destination queue, which is the WebSphere Commerce input queue where

the message must be put. The 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 Business Edition and SAP R/3 are mapped.

User or shopper

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 user or shopper 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 R/3.

Store

A WebSphere Commerce store is mapped to a SAP R/3 **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. In SAP R/3, for this reference application, 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 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 SAPToolTech sample store archive is used for shipping. You must define the corresponding incoterms and conditions in SAP R/3. 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.

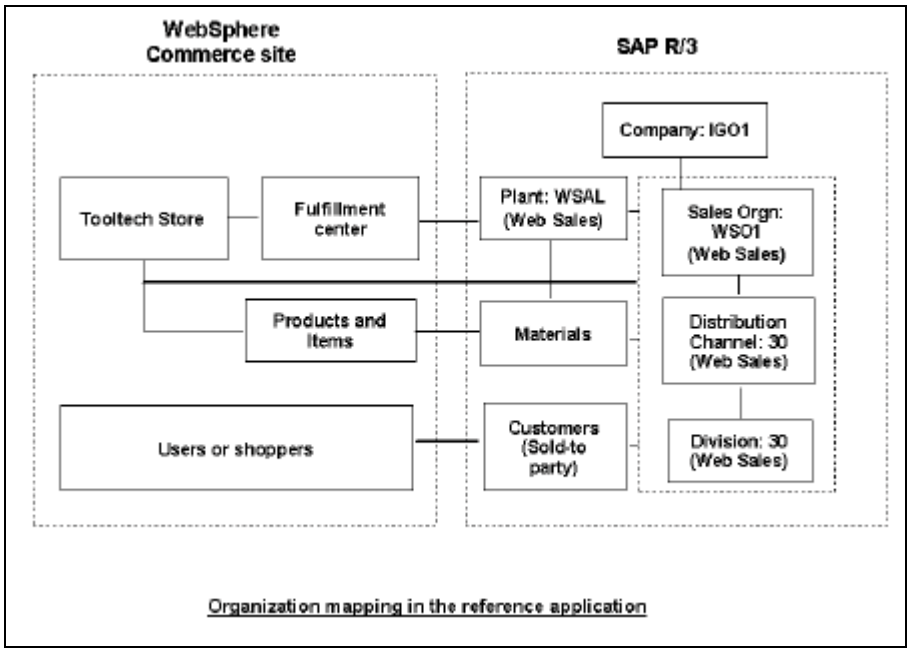

Figure 3: Organization mapping

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.

Chapter 6. Installation and configuration

To use the WebSphere Commerce – SAP integration you must install and configure the following relevant components. The configuration instructions described in this section apply to Windows NT and Windows 2000 operating environments only.

- WebSphere Commerce version 5.4, Business Edition and WebSphere Commerce MQSeries Adapter
- WebSphere MQ Integrator and MQSeries link for R/3
- Updating literals for WebSphere MQ Integrator message flows

Configuring WebSphere Commerce and MQSeries to enable WebSphere Commerce MQSeries adapter

Install WebSphere Commerce version 5.4, Business Edition. Refer to the product documentation for the installation steps and the post-install configuration. The WebSphere Commerce messaging system is equipped to handle the messages described in Chapter 4. Message flow, to interact with back-end systems.

The outbound messaging system can generate, for example, an `OrderCreate` message in XML format.

1. To enable the Report_NC_PurchaseOrder XML message, update the WebSphere Commerce database table CMDREG using the following SQL statement:

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

Note: Refresh the command registry for the changes to take effect.

2. Install WebSphere MQSeries version 5.2. Refer to WebSphere MQSeries documentation on how to set up either the WebSphere MQSeries binding mode, or the WebSphere MQSeries client mode configuration. You must have WebSphere MQSeries client mode configuration if your WebSphere Commerce is installed on a different machine. You can install WebSphere MQSeries in <*drive*>:\WebSphere directory or in any other location. A default queue manager is created when installing MQSeries called *QM_hostname,* where *hostname* is where you have installed MQSeries.

3. If you do not want to use the default queue manager, you can create a new one using WebSphere MQSeries Explorer. If you are unable to use the WebSphere MQSeries Explorer, then use the following commands from the command line to create your queue manager:

```
crtmqm Queue_Manager_Name
strmqm Queue_Manager_Name
```

Where, *Queue_Manager_Name* is the name you have chosen for the queue manager in this setup that creates the queue. `crtmqm` creates the queue manager. `strmqm` starts the queue manager.

4. To set the coded character set identifier of the queue manager to 1208 (UTF8), type the following WebSphere MQSeries commands from the command line:

```
strmqm Queue_Manager_Name

runmqsc Queue_Manager_Name


alter qmgr ccsid(1208)
end
```

Where,

`runmqsc` starts the MQ command line and `end` ends the MQ command line processor.

5. Define the three inbound queues, (Inbound queue, Serial inbound queue and Parallel inbound queue), Outbound Queue and Error Queue for the queue manager. Use WebSphere MQSeries Explorer or WebSphere MQSeries commands to do this. The commands for creating a queue are:

```
runmqsc Queue_Manager_Name

define qlocal(Queue_Name)
end
```

Where, *Queue_Name* is the name of the local queue that you must create.

Note: The queue names are case sensitive.

6. If you want to verify the queue manager and confirm that the queues created are working, use the **WebSphere MQSeries First Steps - Post Card** utility.

7. Install MA88 product extension in the *<drive>:\MQ_install_path* `\java` directory.

Where, *MQ_install_path* is the path where you have installed WebSphere MQSeries.

8. To configure the WebSphere Commerce messaging system to work with the JMS (Java Messaging Service), perform the following steps:

    a. Open a MS-DOS command window on your WebSphere Commerce machine and do the following:

    b. Update your classpath variable by typing the following command on one line:

```
set classpath=
%classpath%;MQ_install_path\java\lib\com.ibm.mqjms.
jar;MQ_install_path\java\lib\com.ibm.mq.jar;
WAS_install_path\lib\ns.jar
```

Where, *MQ_install_path* is the path in which you installed WebSphere MQSeries and *WAS_install_path* is the path in which you installed the WebSphere Application Server.

c. Add a new environment variable named *MQ_JAVA_INSTALL_PATH* by typing the following command:

```
set MQ_JAVA_INSTALL_PATH=MQ_install_path\java
```

.

d. Update the environment to use the JDK (java development kit) that comes with WebSphere Application Server by typing the following command.

```
set PATH = WAS_Intall_Path\Java\bin;%PATH%
```

Where, *WAS_install_path* is the path in which you installed the WebSphere Application Server.

e. Run the JMS Admin tool. Ensure that the WebSphere Application Server is running and the correct classpath and environment variable defined previously in this section are added.

i) Change to the *MQ_install_path*\java\bin directory. Open the JMSAdmin.config file and set the following values.

INITIAL_CONTEXT_FACTORY=com.ibm.ejs.ns.jndi.CNInitialContextFactory
PROVIDER_URL=iiop://localhost:900 (WebSphere Commerce and WebSphere MQSeries are installed on same machine)
SECURITY_AUTHENTICATION=none

ii) Run the JMSAdmin program by providing the JMSAdmin.config file as a command line input:

```
CommandPrompt:> JMSAdmin -cfg JMSAdmin.config -t -v
```

By executing this command you should be able to lookup the JNDI (Java Naming and Directory Interface) service provided by WebSphere Administration Server. You will see a **InitCtx>** prompt that you can use to run the JMS administration commands.

iii) Register the QueueConnectionFactory and set the coded character set identifier by typing the following commands:

- define qcf*(JMS_QueueConnection_Factory)* qmanager*(Your_QueueManager_Name)*

- `alter qcf(JMS_QueueConnection_Factory) ccsid(1208)`

  Where,

  JMS_QueueConnection_Factory is the name of the MQQueueConnectionFactory JMS object.

  On executing the previous set of commands, an entry for MQQueueConnectionFactory JMS object can be found in the WebSphere Application Server database under the BINDINGBEANTBL table. These objects are registered in the WebSphere Application Server database.

iv) Define the following JMSQueues. These queues are created for this reference application and you can change the queue names to suit your requirements.

- JMSSerialInboundQueue
- JMSParallelInboundQueue
- JMSInboundQueue
- JMSOutboundQueue
- JMSErrorQueue

Note: The JMSQueue names and JMS Connection Factory name must be same as the values entered in the connectionSpec section, of the instance xml file. You can find the details under the Transports section in the WebSphere Commerce Configuration Manager.

Syntax:

Type the following command on a single line:

`define q(JMS_Serial_Inbound_Queue)qmanager(Your_QueueManager_Name) queue(Your_Serial_Inbound_Queue)`

Where,

- *Your_Serial_Inbound_*Queue is the WebSphere MQSeries queue created for the serial inbound queue.
- *JMS_Serial_Inbound_Queue* is the name of the JMSQueue object.
- *Your_Queue_Manager_Name* is the name of the Queue Manager created.

v) After creating the queues set the following property to the outbound and error queues. This indicates that JMS is dealing with a native WebSphere MQSeries application.

```
alter q(JMS_Outbound_Queue) targclient(MQ)
alter q(JMS_Error_Queue) targclient(MQ)
```

f. Type `end` to exit the JMSAdmin tool. This finishes configuring the Java Messaging Service with WebSphere Commerce.

9. **Start** the WebSphere Commerce Administration Console. Connect as a Site Administrator, go to **Configuration** and choose the **Transport** option. Select WebSphere MQSeries as your transport and change the status to **activate**.

10. Log out from the Administration Console.

11. After you finish publishing the store, log into the Administration Console, this time as a Store Administrator and select the store used by this reference application. From **Configuration** add **MQ Transport** to the store. An entry for this is made in the STORETRANS table.

12. Enable the messaging system transport adapter. Open the **WebSphere Commerce Configuration Manager**, select the instance, and choose **TransportAdapter** under the Components node. To enable this component select it and **apply** the changes.

13. Update the WebSphere Application Server class path for the instance, to add the additional `jar` file entries. Open the WebSphere Application Server Advanced Administrative Console and complete the following:

    a. Select the host on which, you are running your WebSphere Commerce instance.

    b. Select the **WebSphere Administrative Domain**.

    c. Select **Nodes**.

    d. Select your **host name**.

    e. Select **Application Servers**.

    f. Select the WebSphere Commerce Server *instance_name*, where *instance_name* is the name of your WebSphere Commerce instance.

    g. Go to the JVM settings of the instance.

        i) Select **Add** a new system property.

        ii) Type in the following system property:

        `name= ws.ext.dirs`

        `value=`*`MQ_INSTALL_PATH`*`/java/lib`

        Where, *MQ_Install_Path* is the path where you installed WebSphere MQSeries.

14. Restart the WebSphere Application Server service for all the changes to take affect.

Installing and configuring WebSphere MQ Integrator version 2.1

The WebSphere MQ Integrator (WMQI) install shield wizard installs both the WebSphere MQ Integrator and WebSphere MQSeries. If WebSphere MQSeries is already installed, only the WebSphere MQ Integrator components will be installed. The WebSphere MQ Integrator 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 WebSphere MQ Integrator is referred to as MQSIMRDB.

Setting up the WMQI message sets

1. Use the `import.bat` batch file present in 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 WMQI control center.
   - You have stopped the configuration manager and broker.

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

2. Open the `import.bat` file and modify the message repository db name, user ID, password, RefAppDir 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 the 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 pane. Choose **Add to workspace** and select **Message Set**.

8. From 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 panel select the **Message Flows** check box. Note that this is selected by default.

3. Click **Browse** and select the message flow XML files present in 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, from **Properties** of the input node for each message flow move to the **Basic** tab and make the change.

5. To change the queue manager and output queue names to suit your set-up, from **Properties** of the output node for each message flow, move to the **Basic** tab and make the changes.

6. Change the **File path** from the **Properties** of 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 pane and right click. Select **Check in**.

2. Move to the **Assignments** tab in control center.

   a. From the left pane check out MQSI_SAMPLE_BROKER and the default execution group.

   b. Drag and drop the message flows from the center pane to the default box in the right pane.

   c. Drag and drop all message sets associated with the message flows into broker box, 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 IDoc parser is a WebSphere MQ Integrator 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 version 2.1 at 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` for information on how to use the parser for WMQI version 2.1. For more information refer to http://www.ibm.com/software/ts/mqseries/txppacs/ia0f.html

A Category1 SupportPac is also available as a service offering that provides SAP IDoc processing capability with WebSphere MQ Integrator. It also enables loading of SAP IDoc type metadata into the WMQI message repository and enables processing of basic and extended IDocs at runtime. For more details about the Category 1 SupporPac, refer to http://www.ibm.com/software/ts/mqseries/txppacs/ia0p.html

Literals

The component that requires updates or changes is a literal. A literal carries a fixed value. Literals store values that will be used by the broker 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 Business Edition with SAP platforms 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. You must 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
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	10001

Literal to be changed	Description of the value	Sample value
	sales	
WC_MemberID	WebSphere Commerce Business Edition member ID that owns the store and products	10001
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 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 these 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. See, Chapter 7. Configuring SAP.

   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. Configure destinations for the outbound servers.

   The destinations for the outbound server can be configured in the file `smqDestConf` 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.

3. 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.

4. 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. The queue names given in the following section belong to both WebSphere Commerce and MQSeries link for R/3.

Defining queues

The queue names must be identical to those provided in the sample `in.ini` and `out.ini` files. You must create the following queues for this reference application:

- OUT - The outbound queue for WebSphere Commerce
- IN - The inbound queue for WebSphere Commerce
- ERR – The error queue for WebSphere Commerce.
- SIN – The serial inbound queue for WebSphere Commerce.
- PIN – The parallel inbound queue for WebSphere Commerce.
- MQ_LINKIN - The inbound queue to the WebSphere MQSeries link for R/3. WMQI must place the messages intended for SAP R/3 in this queue.
- MQ_LINKOUT - The outbound queue from the WebSphere MQSeries link for SAP R/3. WMQI must read the messages received from SAP R/3, from this queue.
- MQLINKBADMSG - If any errors occur during the processing of messages by the WebSphere MQSeries link, the error message will be placed in this queue.
- TR_MQLINKIN – The transaction queue required by WebSphere MQSeries link for inbound transactions.
- TR_MQLINKIDOUT – The transaction queue required by WebSphere MQSeries link for outbound transactions.

Note: The queue names given previously are samples and can be changed according to your installation.

Assuming that you have all the components except SAP in a single system, you can create a sample WebSphere MQSeries configuration by doing the following:

1. Create a queue manager (if it hasn't already been created) named WCBESAPQM. To create this execute the following:

   ```
   crtmqm WCBESAPQM
   ```

2. Start the queue manager WCBESAPQM:

   ```
   strmqm WCBESAPQM
   ```

3. Define the local queues. You will need to run `runmqsc` from the command prompt.

4. After you run `runmqsc`, the MQSeries command prompt displays. Use the following command to create the queues:

   ```
   define qlocal('QueueName')
   ```

Populating the units of measure

This reference application provides a massloadable XML file to upload all the unit of measures defined in SAP to 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` present in the `store\uom` directory, in a text editor and change the class paths and database information as per 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 R/3 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` present in the `store\statecode` directory, in an editor and populate it with the appropriate state codes used in your SAP R/3 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` 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 R/3 installation.

Publishing the sample store

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

You must configure the IBM Payment Manager for this sample store model. For more details about installing and configuring the IBM Payment Manager refer to *WebSphere Commerce Installation Guide*.

Creating a new store

The `SAPToolTech.sar` file for this reference application is present in the `store` directory. This store archive file is built on top of the ToolTech 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 `SAPToolTech.sar` file and `Feature_saptooltech_en_US.html` present in the `store` directory, to *<Drive>:/Commerce_Install_Path/*`samplestores/ToolTech` folder, where *`Commerce_Install_Path`* is the WebSphere Commerce Business Edition install path.

2. Edit *<Drive>:/Commerce_Install_Path/*`xml/tools/devtools/SARRegistry.xml` to add the following lines before `</SAR-properties>` and save the file.

```
<SampleSAR fileName="SAPToolTech.sar"

relativePath="ToolTech">

<html locale="en_US"

featureFile="ToolTech/Feature_saptooltech_en_US.html"

sampleSite="ToolTech/preview/en_US/index.html"/>

    </SampleSAR>
```

3. Launch the store services and click **New** in the Store Archives page.

4. To create a new sar file, specify the Store Archive name, Store Directory name, select appropriate organization as store owner, select `SAPToolTech.sar` file and then click **OK.** This creates the sar file.

5. In the Store Archives page select the sar file created in the previous step and click **Publish** and then click **OK.** 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

WebSphere Commerce receives all inbound messages from the SAP R/3 system for processing. WebSphere Commerce allows two levels of security for inbound messages or for the requests being processed by the program adapter.

- Level 1 security - Limited security
- Level 2 security - User ID and Password (credentials) are required for every request.

Level 1 security is enabled by default during installation. By default, the WMQI ESQLs provided with this reference application are configured for Level 1 security. To change the level of security in WebSphere Commerce, you must change the value of the class attribute in the SessionContext node, in the MQSeries Adapter configuration. For more information on how to change Program Adapter Security for MQSeries, refer to the *WebSphere Commerce Version 5.4, Business Edition* 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"."Co
   ntrolArea"."Credentials"."LogonID" = 'logonId';

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

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

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

   The *`logonId`* 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"."Co
ntrolArea"."Credentials"."LogonID" = ‘logonId’;

SET
"OutputRoot"."XML".(XML.Element)"Create_WCS_Customer"."Co
ntrolArea"."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.

Chapter 7. Configuring SAP

This section details how to configure the SAP R/3 system for this reference application. It includes the transactions to be used, customizations to be made, and a description of creating master data. Assumptions that have been made are listed where appropriate.

- Organization structure
- Customizing master data
- Configuring price, freight, and tax
- Customizing order numbers
- Configuring partner profile communication
- Transactions – Master data

Organization structure

For this integration you must define an organization structure in SAP R/3. You can use your existing organization structure or create a new one. The following description of the organization structure includes the Company Code, Controlling Area, Sales Area, Plant, and so on.

Creating Company Code – transaction code EC01

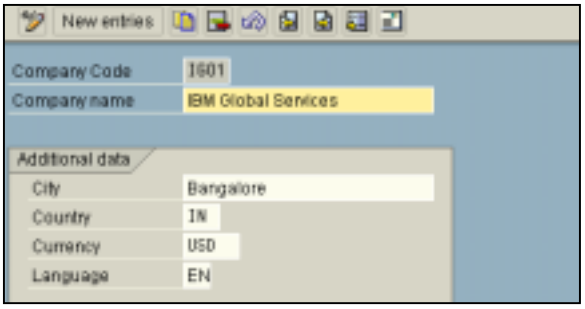

Figure 4: Creating a company code

Create a new company code, IGO1 with the currency as USD. Copy it from the 0001 company code.

Creating the Controlling Area – transaction code EC16

Create the new controlling area, IGO1. Copy it from the 0001 controlling area.

Figure 5: Creating the controlling area

Creating a Plant – transaction code EC02

Create a new plant, WSAL by copying it from the 0001 plant.

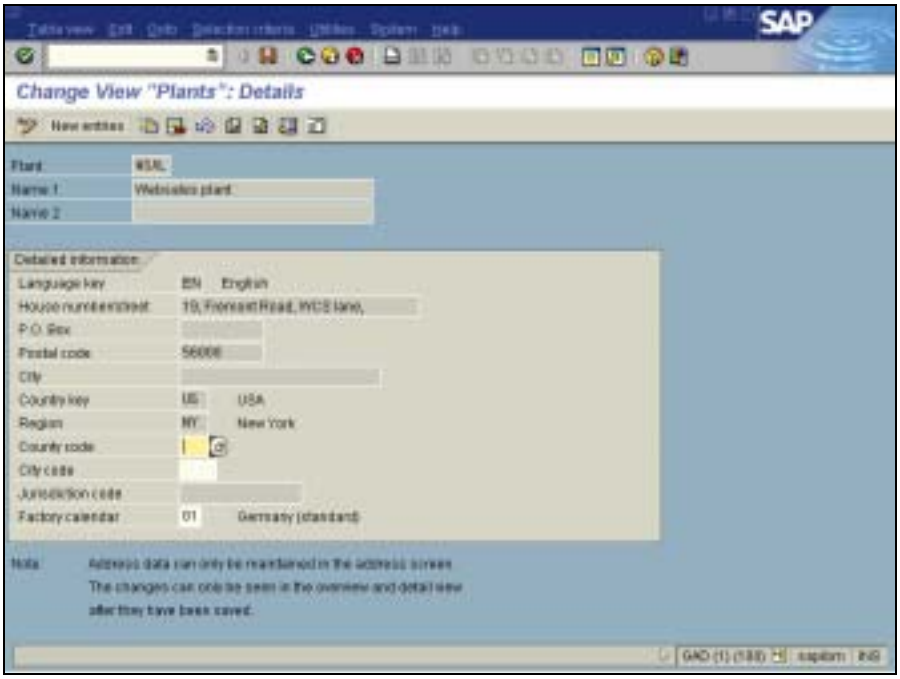

Figure 6: Creating a plant

Maintaining Storage location – transaction code OX09

Maintain storage location WSL2 for the WSAL plant.

Creating a Sales organization – transaction code EC04

Create a sales organization, WSO1 by copying it from the 0001 sales organization.

Figure 7: Creating a sales organization

Creating a Sales division – transaction code EC06

Create a sales division 30 by copying it from the 01 sales division.

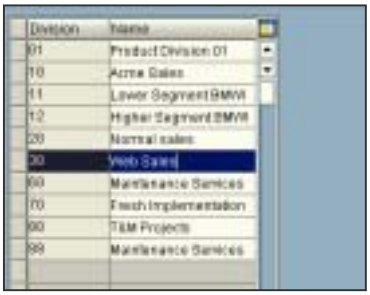

Figure 8: Creating a sales division

Creating a Distribution channel – transaction code EC05

Create a distribution channel 30, by copying it from the 01 distribution channel.

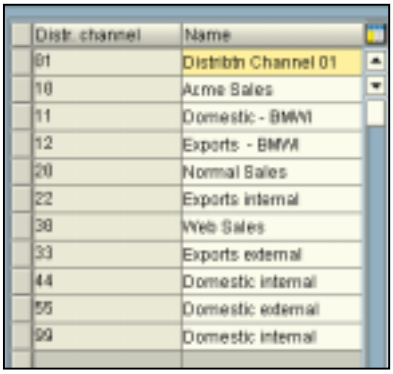

Figure 9: Creating a distribution channel

Creating shipping Point – transaction code EC07

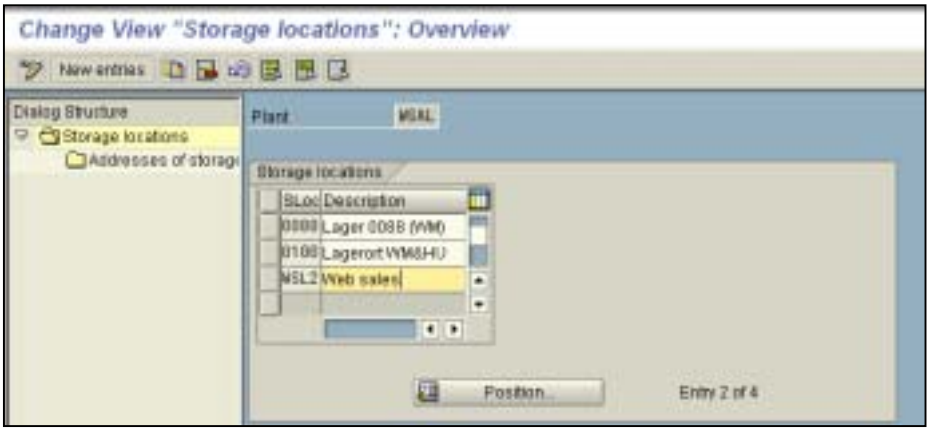

Figure 10: Maintaining storage location

Create the following assignments from transaction SPRO

From the **Enterprise Structure**, go to **Assignment** and select **Logistics, Sales and Distribution**.

1. Allocate plant to company code IG01.

Figure 11: Allocation of plants

2. Allocate the company code to the sales organization.

Figure 12: Allocation of the company code

3. Allocate the sales organization to a division.

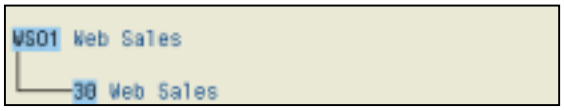

Figure 13: Allocating the sales organization to a division

4. Allocate the sales organization to a distribution channel.

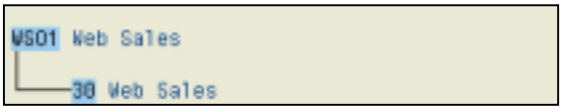

Figure 14: Allocating the sales organization to a distribution channel

5. Create a sales area (Sales Organization, Distribution channel, and Division).

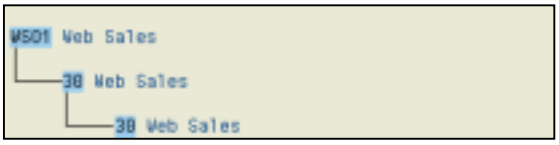

Figure 15: Creating a sales area

6. Allocate a plant to the sales organization and distribution channel.

Figure 16: Allocating a plant

7. Allocate the shipping point to a plant.

Figure 17: Allocating a shipping point

8. Allocate the shipping point to a plant, shipping conditions, and loading group.

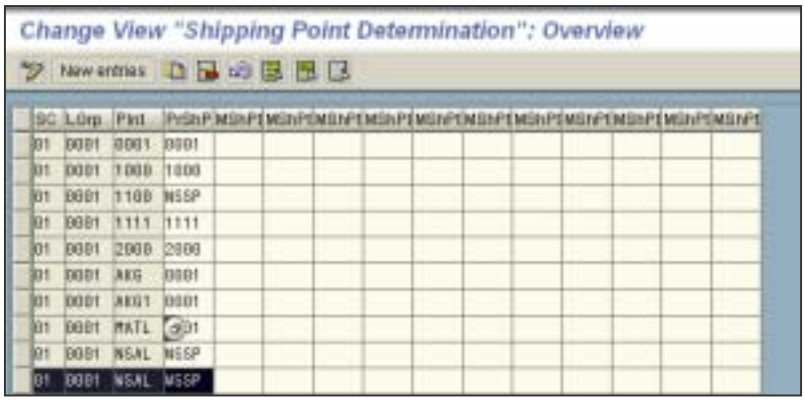

Figure 18: Allocating shipping point

Define a common distribution channel and division – transaction VOR1/VOR2

Maintain the following settings for the sales organization WSO1.

Figure 19: Distribution Channel settings

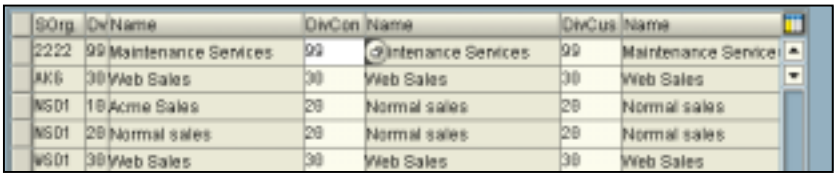

Figure 20: Division settings

Create a new order type – transaction code VOV8

A new order type ZOR1 (Web Sales) must be created. Create this by copying it from the standard order type OR.

Note: All of the related settings (copy control and so on) are also copied**.** You can continue working with the standard order type “OR” when called from WebSphere Commerce. However, it is recommended not to use the standard order type, instead copy from the standard order type.

Defining the MRP Controller

Every material that is relevant to the planning run must be assigned an MRP controller number in the material master record. The MRP Controller must be entered when creating materials for the WSAL plant.

1. To do this, go to transaction code SPRO, and click **SAP Reference IMG**.

2. From the tree structure, select **Implementation Guide for R/3 Customizing (IMG)**

3. Select **Production** > **Material Requirement Planning** > **Master Data** > **Define MRP Controllers**.

4. Add a new entry for the WSAL plant as a copy of 0001 as shown in the following figure.

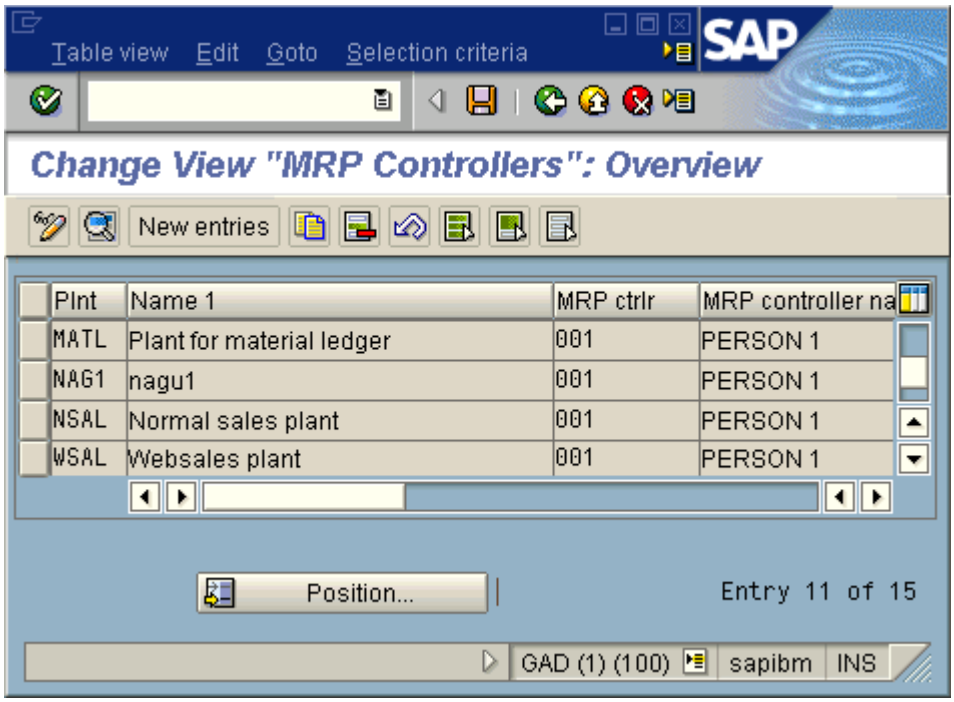

Figure 21: Defining the MRP Controller

Defining the Floats (Schedule Margin Key)

The floats are allocated to the material through the release period key in the material master record. The schedule margin key must be entered when creating materials for the WSAL plant.

1. To do this, go to transaction code SPRO, and click **SAP Reference IMG**.

2. From the tree structure, select **Implementation Guide for R/3 Customizing (IMG)**

3. Select **Production** > **Material Requirement Planning** > **Planning** > **Scheduling & Capacity Parameters** > **Define Floats (Schedule Margin Key)**.

4. Add new entries for the WSAL plant as a copy of 0001 as shown in the following figure.

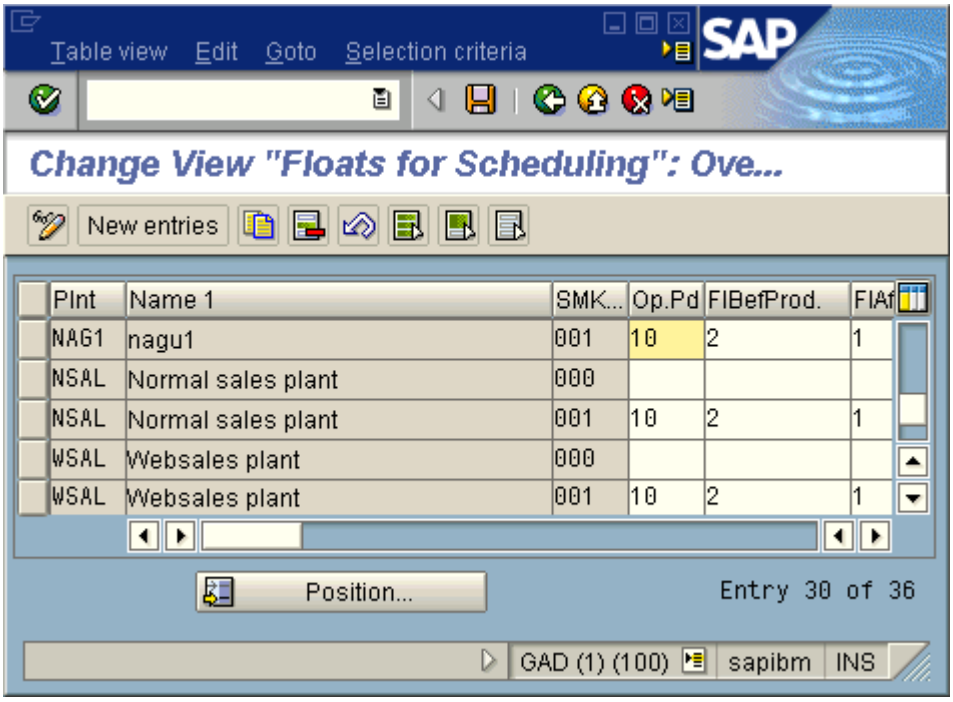

Figure 22: Defining the Floats (Schedule Margin Key)

Customizing the master data

This section describes how to create a customer account group, assign partner functions to the account group, setting up taxes, and enabling partial delivery.

Creating customer account group

Create a new customer account group called '**ZINC**'.

1. To do this, go to transaction code SPRO, and click **SAP Reference IMG**.

2. From the tree structure, select **Implementation Guide for R/3 Customizing (IMG)** as shown in Figure 23.

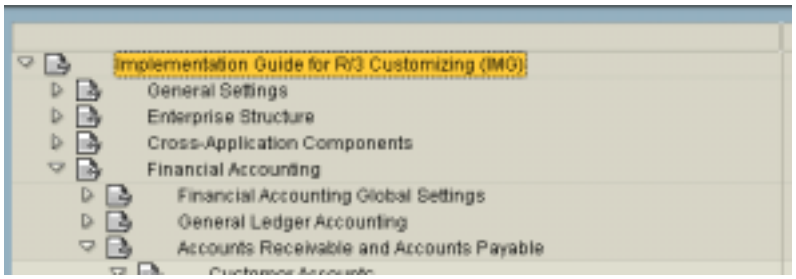

Figure 23: Customizing the master data

3. Select **Financial Accounting** > **Accounts Receivable & Payable** > **Customer Accounts** > **Master Records** > **Preparations for Creating Customer Master Records > Define Account Groups with screen layout (customers)**.

4. Copy account group **0001** to **ZINC**. Change the field group **General Data.** From the address group make **Region** mandatory. This field is used to calculate taxes and freight costs.

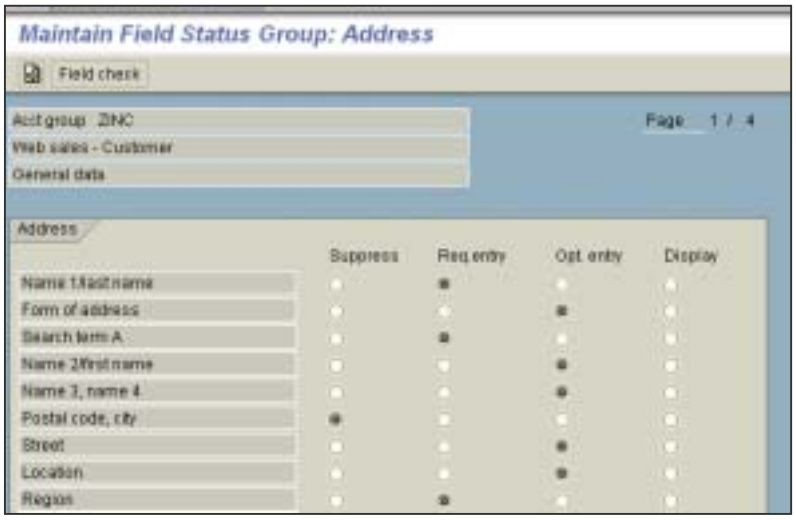

Figure 24: Maintaining the field status address group

5. From the field group **Sales Data**, go to the Billing group and make the **Terms of payment** field mandatory.

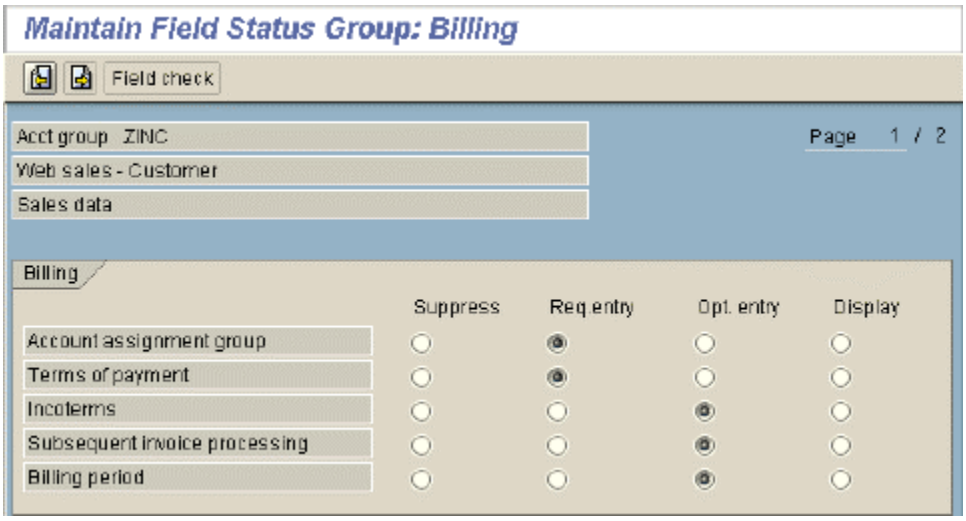

Figure 25: Maintaining the field status sales group

Assigning partner functions to the customer account group

For the customer account group, ZINC, you must assign the sold-to party, bill-to party, ship-to party, and payer partner functions.

1. To do this, go to transaction code SPRO, and click **SAP Reference IMG**.

2. From the tree structure, select **Implementation Guide for R/3 Customizing (IMG)**.

3. Select **Sales and Distribution** > **Basic Functions** > **Partner Determination** > **Define and Assign Partner Determination Procedures**.

4. Select **Customer master** and click **Partner Functions**.

5. From the menu, go to **Environment** and select **Account Group Assignment.**

6. Add **SP** (sold-to), **BP** (bill-to), **SH** (ship-to), and **PY** (payer) partner functions for the account group ZINC as shown in Figure 26.

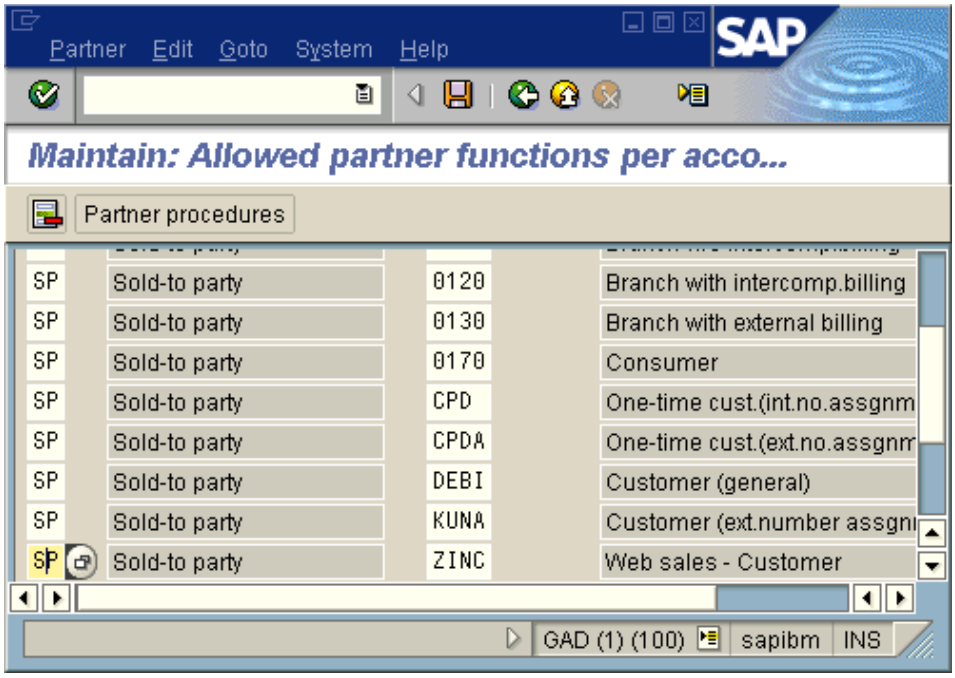

Figure 26: Assigning partner functions

Enabling partial or complete delivery

SAP allows for complete or partial delivery of order items. You can configure this in the customer master information. This information can be set according to the business requirement. By default, SAP sets the delivery option to 'Complete'. The following is the default configuration:

1. Go to SAP transaction xd01 (Create Customer).

2. Create a customer in the account group ZINC for the WebSphere Commerce sales area (sales organization, distribution channel, and division).

3. Complete all the mandatory fields (transportation zone, shipping conditions, reconciliation account, and so on).

4. On the Create Customer: Sales area data Shipping page, select the **complete delivery** check box.

5. In the **part deliveries/item** field enter "C".

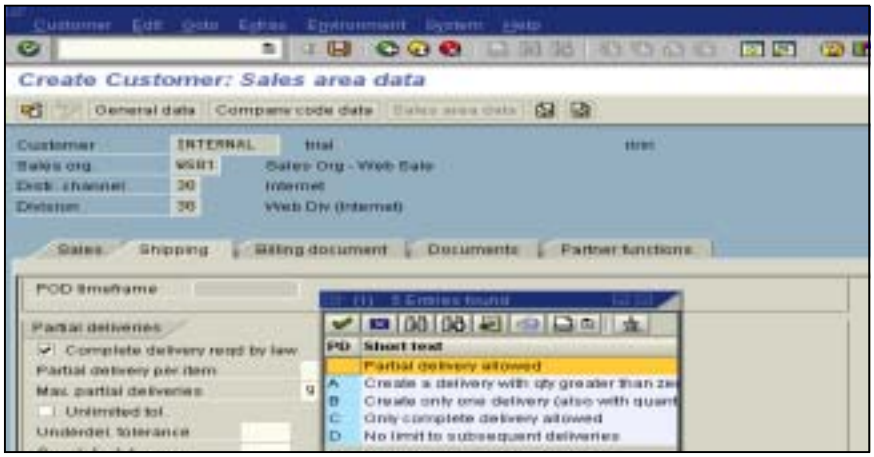

Figure 27: Selecting partial delivery in SAP

6. Enter the other mandatory fields such as tax classification and save.

Note: For existing customers, go to SAP transaction xd02 and create the entries as described in steps 4, 5 and 6.

Setting up taxes

To assign a country to a tax procedure, do the following:

1. Go to transaction code SPRO, and click **SAP Reference IMG**.

2. From the tree structure, select **Implementation Guide for R/3 Customizing (IMG)**.

3. Select **Financial Accounting** > **Financial Accounting Global settings** > **Tax on Sales/Purchases** > **Basic Settings** > **Assign Country to Calculation Procedure**.

4. Assign the procedure TAXUS to country US.

To assign a delivering plant for tax determination, do the following:

1. Go to transaction code SPRO, and click **SAP Reference IMG**.

2. From the tree structure, select **Implementation Guide for R/3 Customizing (IMG)**.

3. Click **Sales and Distribution** > **Basic Functions** > **Taxes** > **Assign Delivering Plants for Tax Determination**.

4. Maintain plant specific settings. The region of the delivering plant is attached to the plant. Attach region NY to plant WSAL.

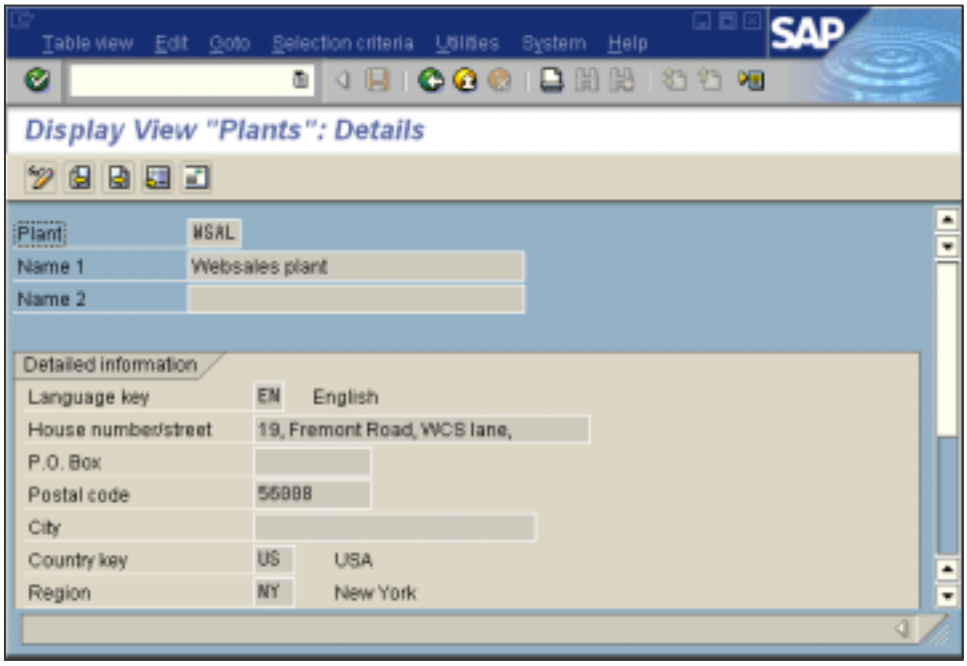

Figure 28: Maintaining plant specific settings

To maintain the tax relevancy of the master record, do the following:

1. Go to transaction code SPRO, and click **SAP Reference IMG**.

2. From the tree structure, select **Implementation Guide for R/3 Customizing (IMG)**.

3. Click **Sales and Distribution** > **Basic Functions** > **Taxes** > **Define tax relevancy for Master Records** > **Customer Taxes (or) Material Taxes.**

4. Maintain customer and material tax classification, for tax category UTXJ as shown in the following figures.

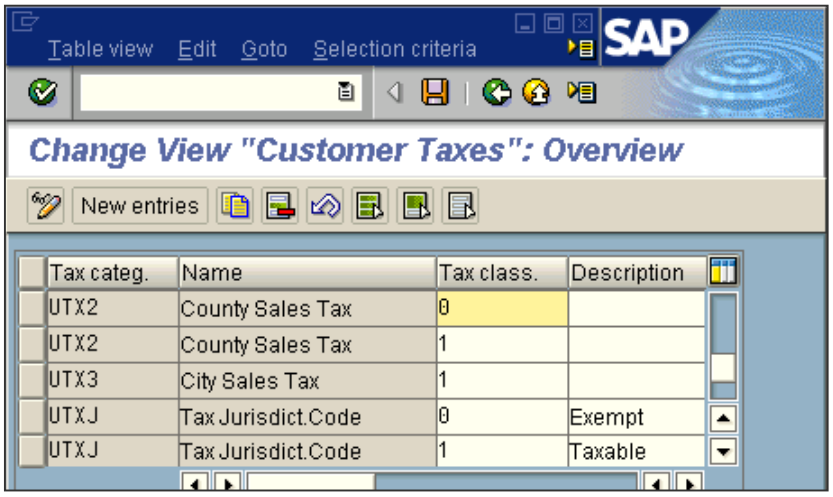

Figure 29: Maintaining tax relevancy for customers

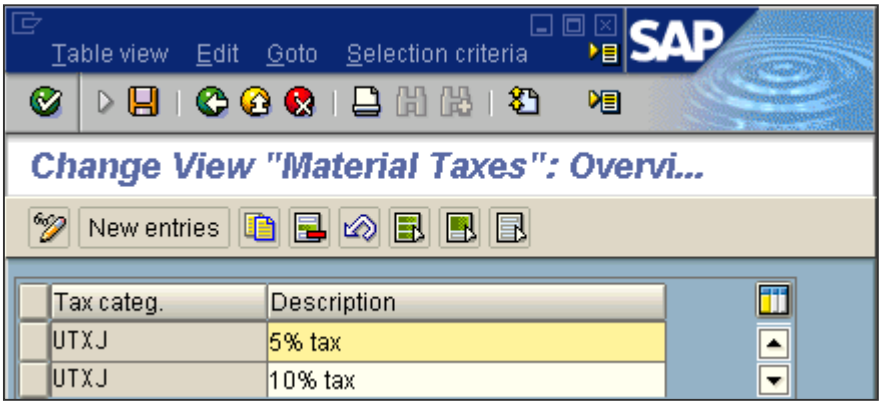

Figure 30: Maintaining tax relevance for materials

To maintain the tax codes using transaction code FTXP, do the following:

Maintain output tax codes A0 to A6 for various percentages of taxes.

- A0 -0%
- A1 – 5%
- A2-10%
- A3-4%
- A4-2%
- A5-1%

- A6-15%

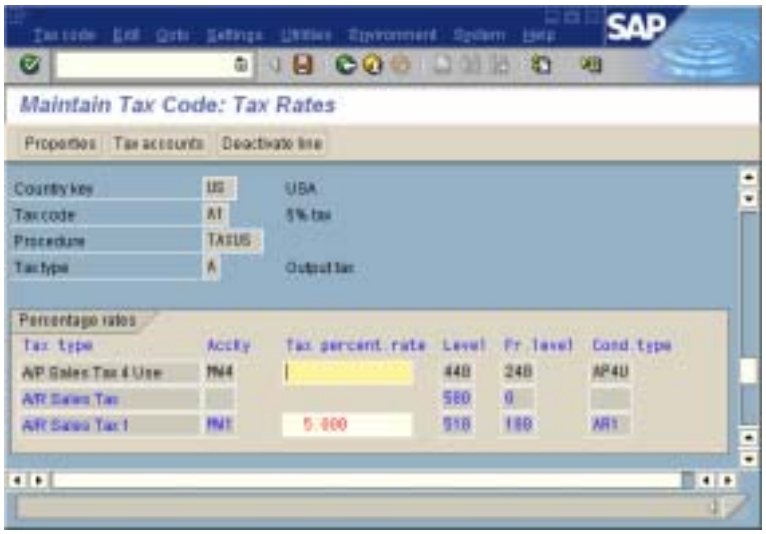

Figure 31: Maintaining tax codes for a country

Maintaining the material master related settings – transaction code OMSF

Maintain the material groups. For this reference application, two material groups 3010 and 3011 are maintained and the materials created for the plant must be associated with a material group.

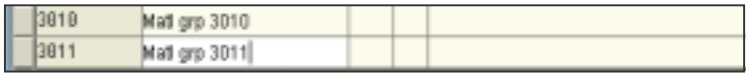

Figure 32: Material master settings

Maintaining the customer master related setting - transaction code OVS9

Maintain the new customer groups. The customer groups are used in pricing conditions for special prices.

Figure 33: Customer master settings

Configuring price, freight, and tax

To configure the price, freight and, tax do the following:

- Create the condition table

- Create the access sequence
- Create the condition type
- Create the pricing procedure
- Assign the pricing procedure
- Account determination procedure

Creating the condition table – transaction code V/03

SAP stores condition records in the form of tables. Create the following tables for pricing records, freight cost, and tax calculations:

- Table 502 for pricing

1. Create a new condition table 502, with the fields, **Sales organization**, **Distribution channel**, **Division**, **Customer group**, and **Material.**

2. To create tables:

    a. Go to **T.code V/05 > create**

    b. Select the fields that are required as shown in Figure 34 to create the table and save.

Figure 34: Table 502 for pricing

- Table 700 for calculating the freight

  This table contains the region of the delivering plant, the region of the customer, and the Incoterms 1 and 2 fields. Incoterms 2 is in the Text field.

Figure 35: Table 700 to calculate the freight

- Table 699 for calculating taxes

  This table contains the Country, Region of Delivering Plant, Material tax classification, Customer tax classification and Region of customer fields as shown in the following figure.

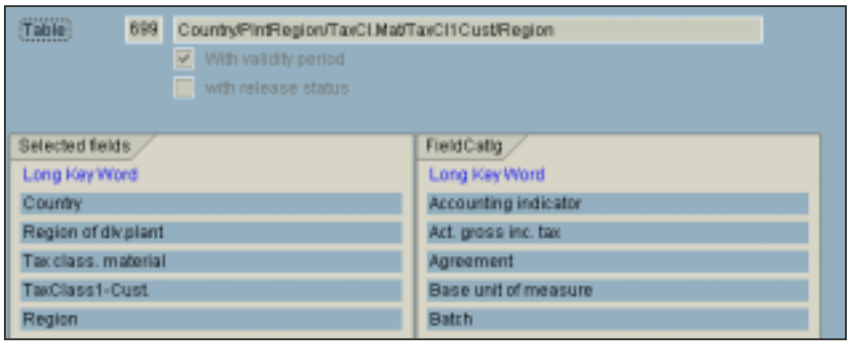

Figure 36: Table 699 to calculate taxes

Creating the access sequence – transaction code V/07

The access sequence searches for the valid condition record for a particular condition type, for example, it searches all the condition tables for the record until a valid record is found.

1. Create the access sequence ZPR0 for prices.

   This access sequence contains tables 502, 305 and 6 and 4. It searches these tables for a valid record and returns the value to the transaction.

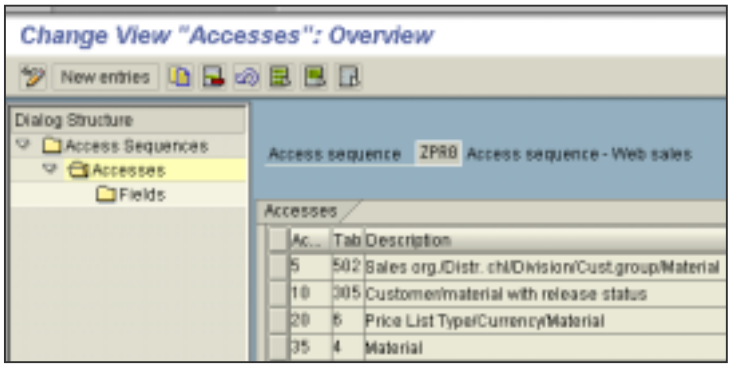

Figure 37: The access sequence that contains tables

2. Create the access sequence ZFR0 for freight (shipment costs).

   This access sequence contains table 700 as shown in the following figure. It searches this table for a valid record and returns the value to the transaction.

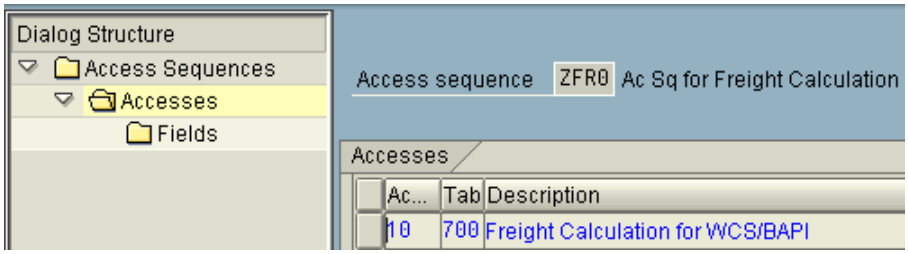

Figure 38: Access sequence ZFRO

3. Create the access sequence UTX1 for tax calculation.

   This access sequence contains tables 699, 78, 40, and 2. It searches these tables for a valid record and returns the value to the transaction.

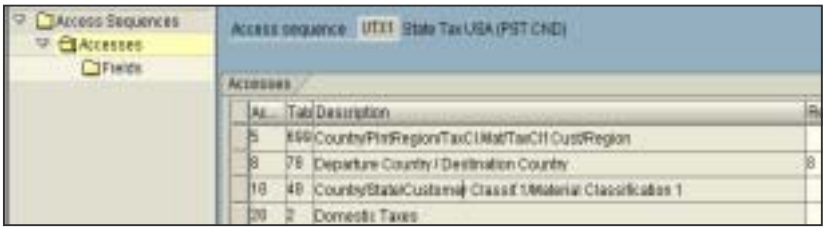

Figure 39: Access sequence UTX1

Creating the condition type – transaction code V/06

The various conditions that appear in a sales transaction such as prices, discounts, tax, and freight are recorded using the condition type. Create the following condition types:

1. Condition type ZPR0.

   Condition type ZPR0 is a copy of condition type PR02. Change PR02's access sequence to ZPR0.

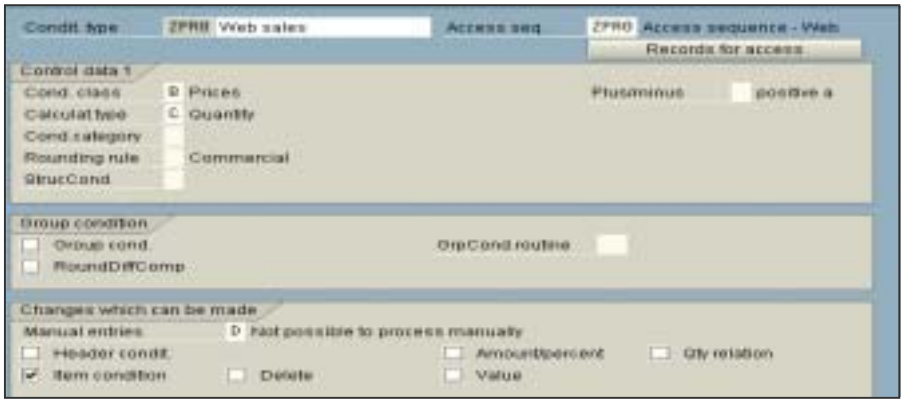

Figure 40: Creating the condition type ZPRO

2. Condition type ZF00.

To create condition type ZF00 copy it from any existing freight condition. Set the access sequence to ZFR0.

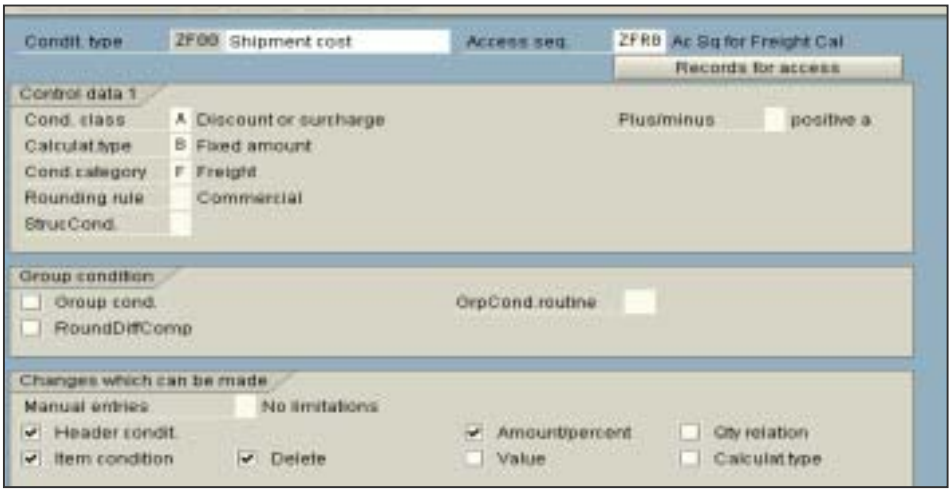

Figure 41: Creating the condition type ZF00

Creating a pricing procedure - transaction code V/08

Create a new pricing procedure called ZINSAL. This is a copy of RVA001 with the changes given below. The total price, tax, and shipping costs can be picked up from the fields KZWI1 - KZWI6.

Figure 42: Creating a pricing procedure

Make the following changes in the pricing procedure:

1. In step 11 of the pricing procedure, replace PR00 by **ZPR0**.

2. In step 815, enter the condition type **ZF00** and attach the act key ERF.

3. In step 915, replace MWST by **UTXJ**. In the subtotal field select "**5**" (carry over values to KZWI5).

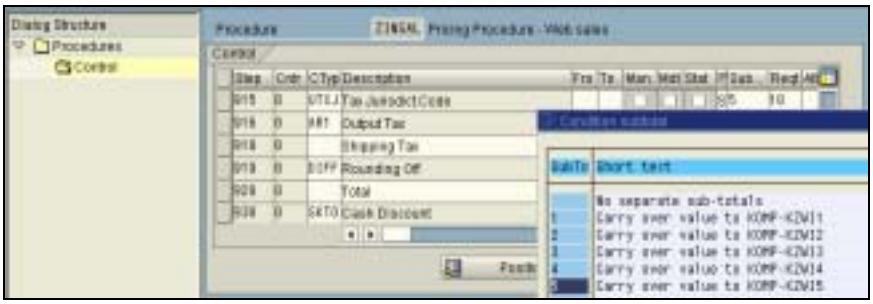

Figure 43: Step 915 for changes in pricing procedure

4. Create step 916 and enter condition type **AR1** (copy of MWST) and make all the settings in the pricing procedure as per step 915, except for the subtotal field, which you must leave blank.

5. Since the total tax shipping price is not a standard functionality in SAP, a new routine needs to be written in addition to certain pricing procedure modifications.

6. Create step 918 and enter the description as **Shipping tax**. Mark the **Stat** field. In the **Subtotal** field select "**6**" (carry over values to KZWI6). In the **AltCTy** field enter 900. (Before this step, ensure that you have created AltCTy 900). To create AltCTy 900 do the following:

    a. Go to T.code VOFM.

    b. From **Formulas** select **Maintain condition values.** You will need a developer ID and access key to modify the source code.

    c. In the **Routine number** field enter 900, with the description as Shipping tax.

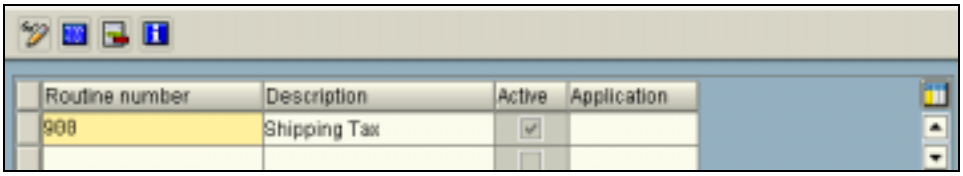

Figure 44: Entering a description in the routine 900 field

    d. In the Change request window create a request.

    e. In the ABAP4 program RV64A900 enter the following code:

```
FORM FRM_KONDI_WERT_900.

Xkwert = komp-kzwi4 * komp-kzwi5 / komp-kzwi3.

ENDFORM
```

    f. Save the routine after activation.

Figure 45: Saving the routine

Assigning a pricing procedure – transaction code OVKK

Assign a pricing procedure ZINSAL to Sales Organization WSO1, Distribution Channel 30, Division 30, Document Pricing Procedure A and Customer Pricing Procedure 1.

Figure 46: Assign pricing procedure

Account determination procedure - transaction code VKOA

Maintain the General Ledger accounts for Sales Organization WSO1 as per the standard for Chart of Accounts maintained for Sales Organization 0001.

Customizing order numbers

The purchase order field BSTNK is available in the sales order. This is taken from the WebSphere Commerce Business Edition order number and appears as an output in SISCSO01 IDoc message. The purchase order number must be made available in SISDEL01 as well. To enable this, you must make certain modifications.

In the copy control between sales order and delivery (T.code VTLA), introduce a new routine to enable copying **vbak-bstnk** to **likp-lifex**. To enable this new routine:

1. Go to T.code VOFM.

Figure 47: Data transports deliveries

2. Go to **Data Transfer** > **Deliveries**. The data transport deliveries window will display as seen in Figure 47.

3. Create routine 900 based on routine 1.

4. Comment the following lines:

```
* IF CVBAK-VBKLT EQ VBKLT_EDL_ENTN OR

* CVBAK-VBKLT EQ VBKLT_EDL_ENTN_KORR.
```

5. Go to transaction code **VTLA**.

6. Select the order type **ZOR1** and delivery **LF**.

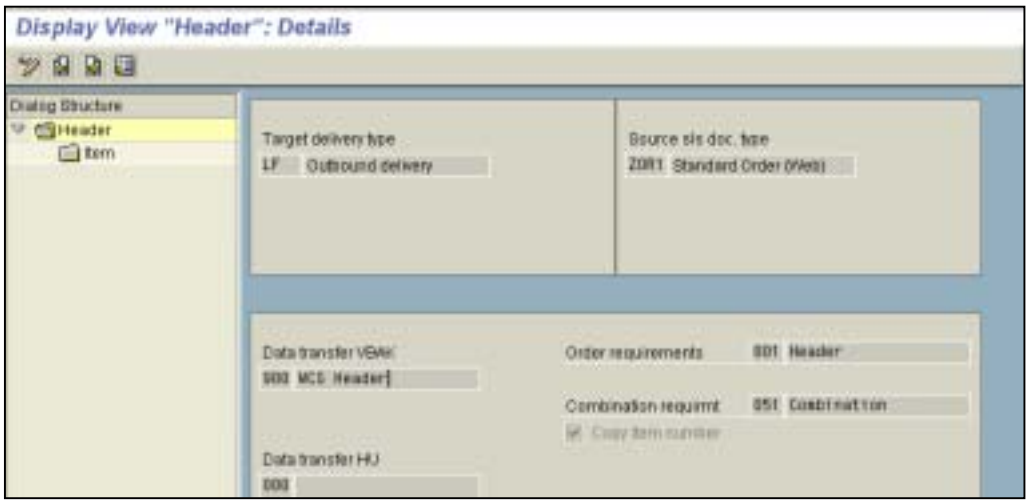

Figure 48: Display View “Header”: Details

7. Change the data transfer **VBAK** field from **1** to **900**.

Configuring partner profile communication

The following output condition records must be created for BA00 (Order Confirmation), LD00 (Delivery Output), and RD00 (Billing) for transmission medium 06 (EDI) and time 4 (immediately). To update these records do the following:

1. From the initial SAP screen, go to **Logistics** > **Sales & Distribution** > **Master Data** > **Output** > **Sales Document** > **Create**.

2. Select output type **BA00** (Order Confirmation).

3. For the selected customer, maintain the partner function, medium language, and time.

4. Create output records for delivery (LD00) and billing documents (RD00).

   To enable communication through EDI, partner profiles must be created for the customer with the relevant output type for the message types SISCS0, SISDEL, and SISINV, and for sales order, delivery, and billing confirmations.

Defining the logical system

To define the logical system, do the following:

1. Go to transaction ALE.
2. Expand the tree '**Sending and Receiving Systems**'.
3. Expand **Logical Systems**.
4. Click **Define Logical System**.
5. Click **OK**.
6. Click on **New entries**. Enter two logical systems, one for SAP and another for WebSphere Commerce, as shown in Figure 49. Save the entries.

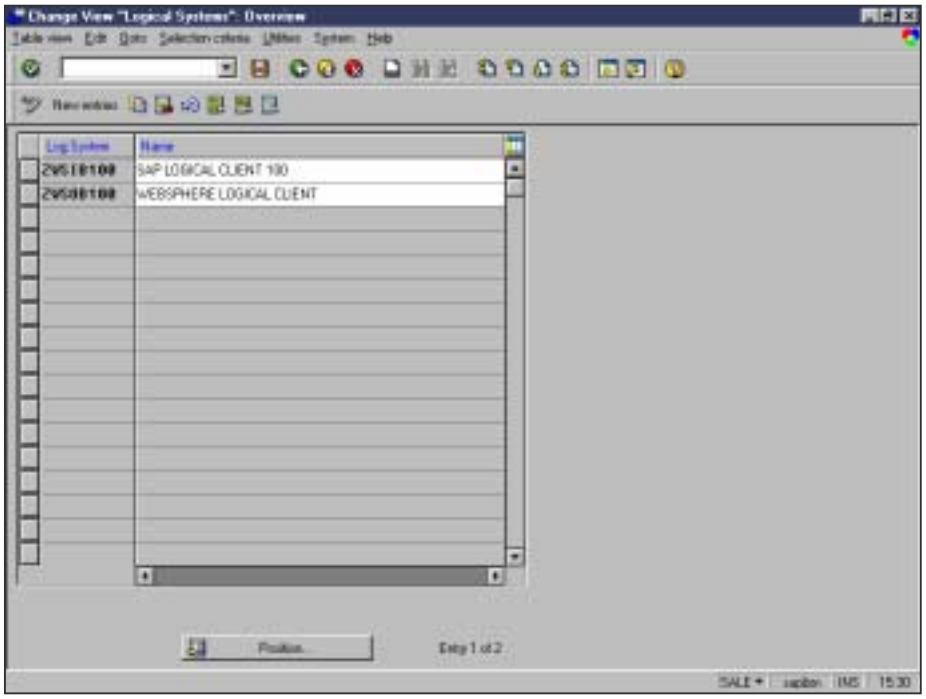

Figure 49: Defining logical systems for SAP and WebSphere Commerce

7. Attach the logical systems to the client by doing the following:
    a. Click **Assign Client to Logical System**.
    b. Click **OK** and proceed.
    c. Double-click on a client to select it.

d. Enter the logical system and save.

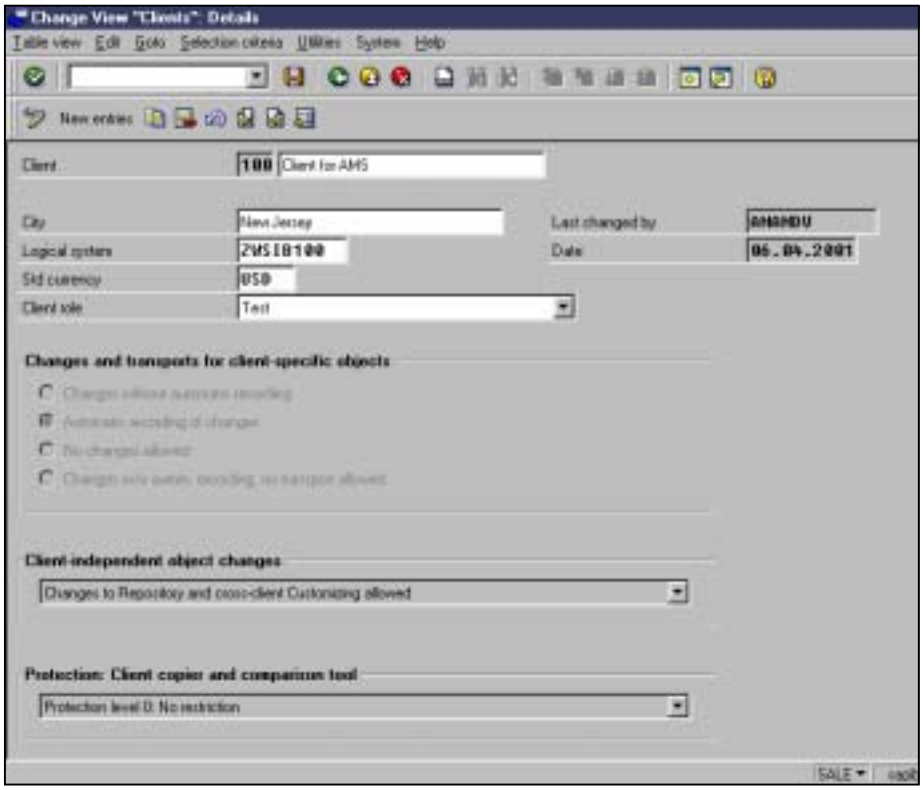

Figure 50: Assigning a client to a logical system

Defining RFC destinations

1. Click **Define the Target Systems for RFC Calls**, as shown in Figure 51.

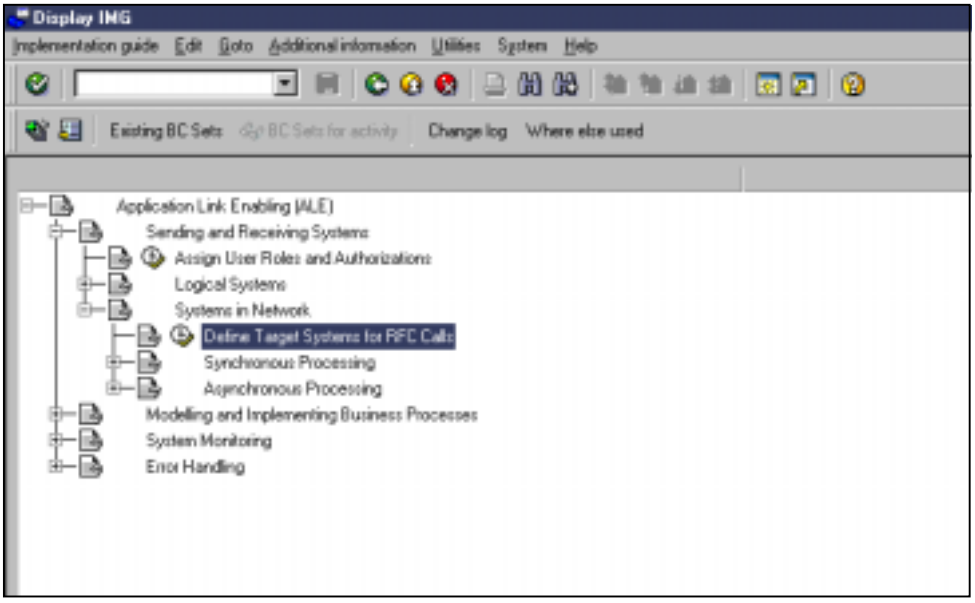

Figure 51: Defining RFC definitions: tree structure

2. Select the R/3 connections and click **Create**.

3. Select the **TCP/IP Connection** and click **Create**.

4. Type the target host RFC destination, user ID, and password for the SAP client.

5. Type the RFC destination, connection type, gateway host, and gateway service for WebSphere Commerce.

6. Ensure that the value of `programid` that is entered is same as the value of the `programid` parameter in the configuration file (`out.ini`) for MQSeries link for R/3 outbound server.

7. Save the data and test the connection.

Defining ports

1. Go to transaction WE21
2. Select Transactional RFC and click **Create**.
3. Select **Generate port name** as shown in Figure 52 and proceed.

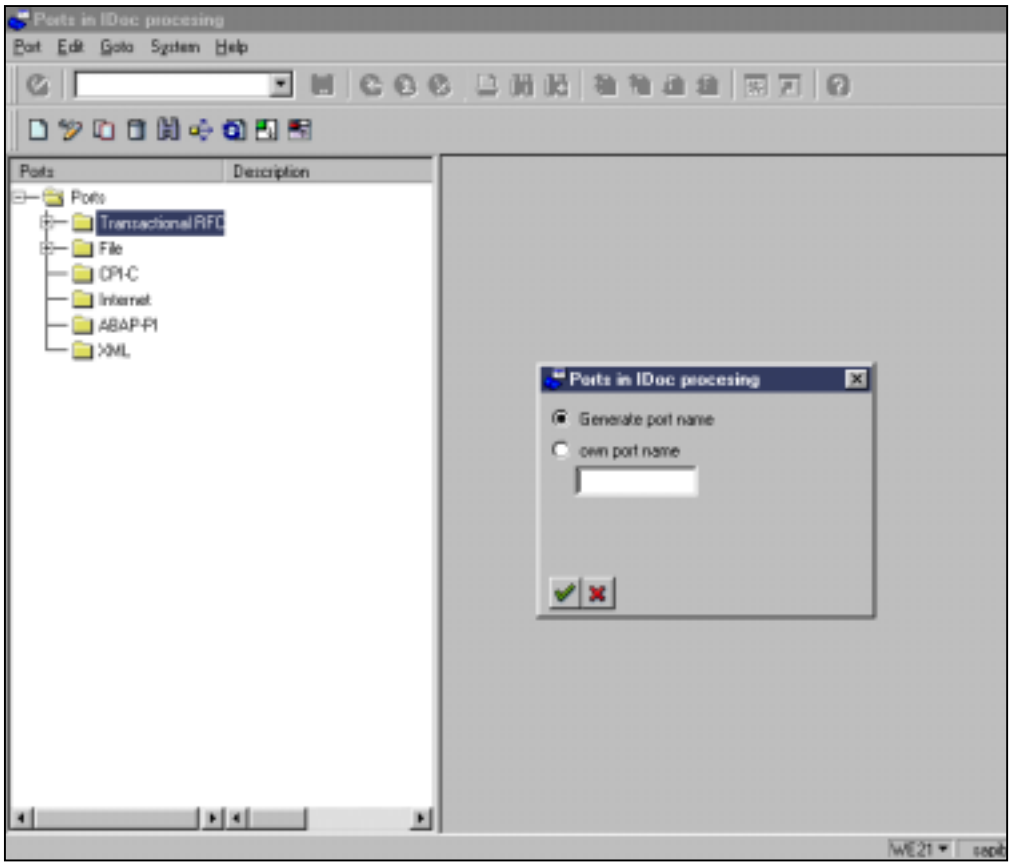

Figure 52: Selecting **Generate port name**

4. Enter a description and RFC destination. Select IDoc record types **Release 4.x**, as shown in Figure 53.

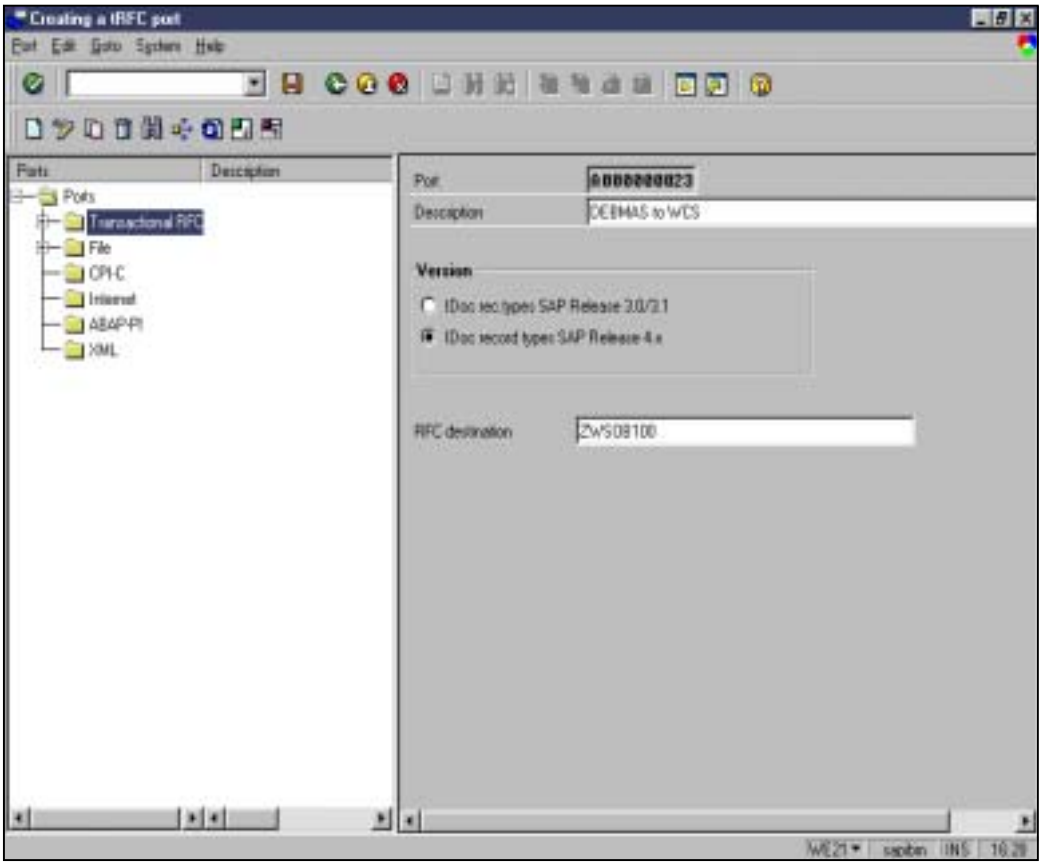

Figure 53: Creating an RFC port dialog

5. Save the RFC destination.

Creating a Customer Distribution Model

1. Go to transaction BD64 and prepare to create a new model, as shown in Figure 54.

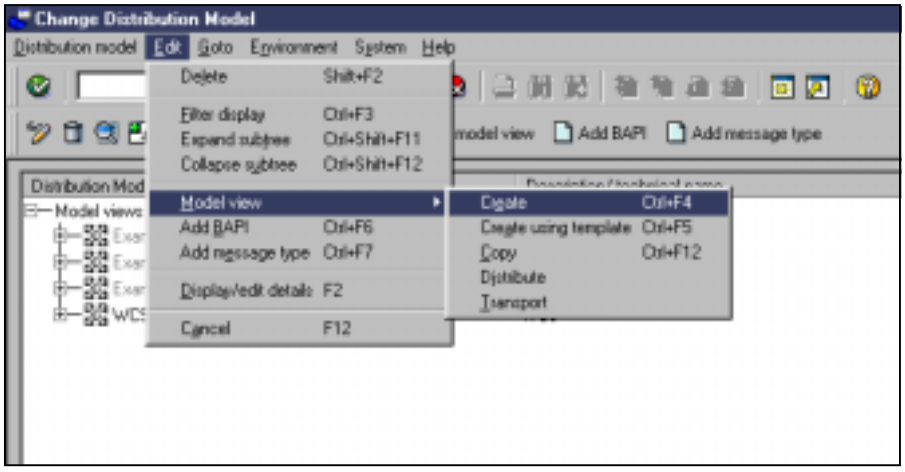

Figure 54: Preparing to create a new model

2. Create the new model, as shown in Figure 55.

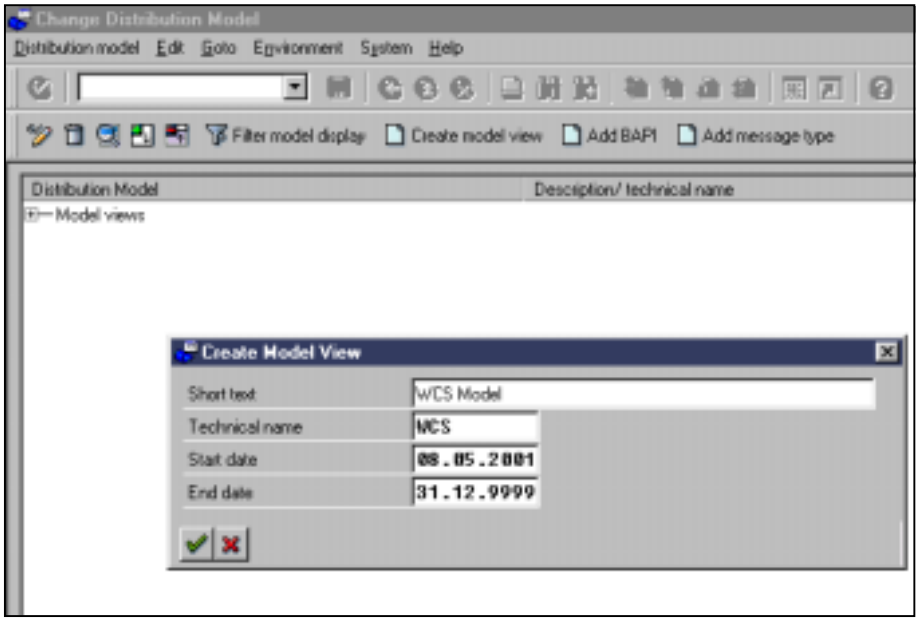

Figure 55: Creating a new model

3. Click **OK**.

4. Select the created model (WebSphere Commerce Business Edition).

5. Click **Add Message Type** as shown in Figure 56.

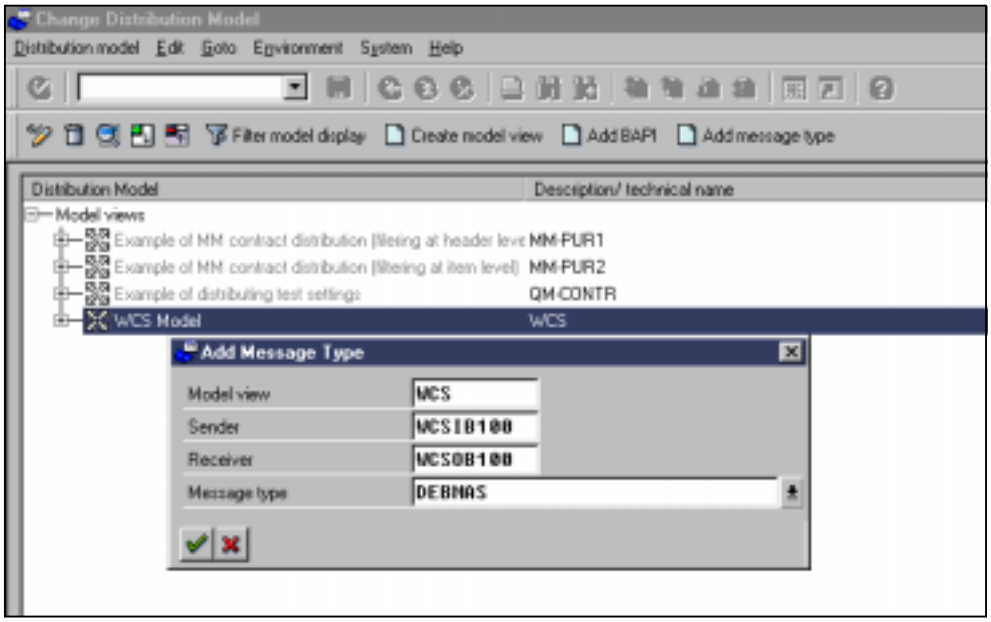

Figure 56: Adding a message type

6. Type the data and click **OK** to proceed.

7. Type all the outbound messages to be enabled and save the model.

8. Select the message type and click **Filter Model Display**.

9. Select Data Filtering and click **Create Filter Group**.

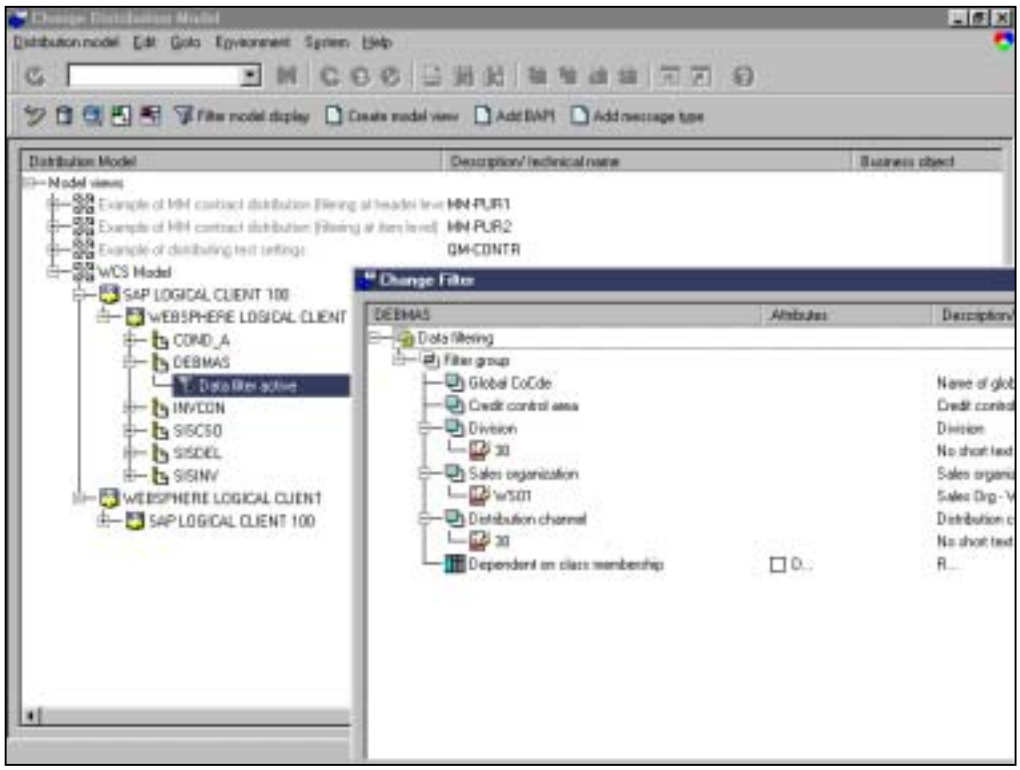

Figure 57: Creating a filter group

10. Type the required data and click **OK** to proceed.

11. Save the data.

Generating partner profiles

1. Go to transaction BD82.

2. Type the model view as shown in Figure 58.

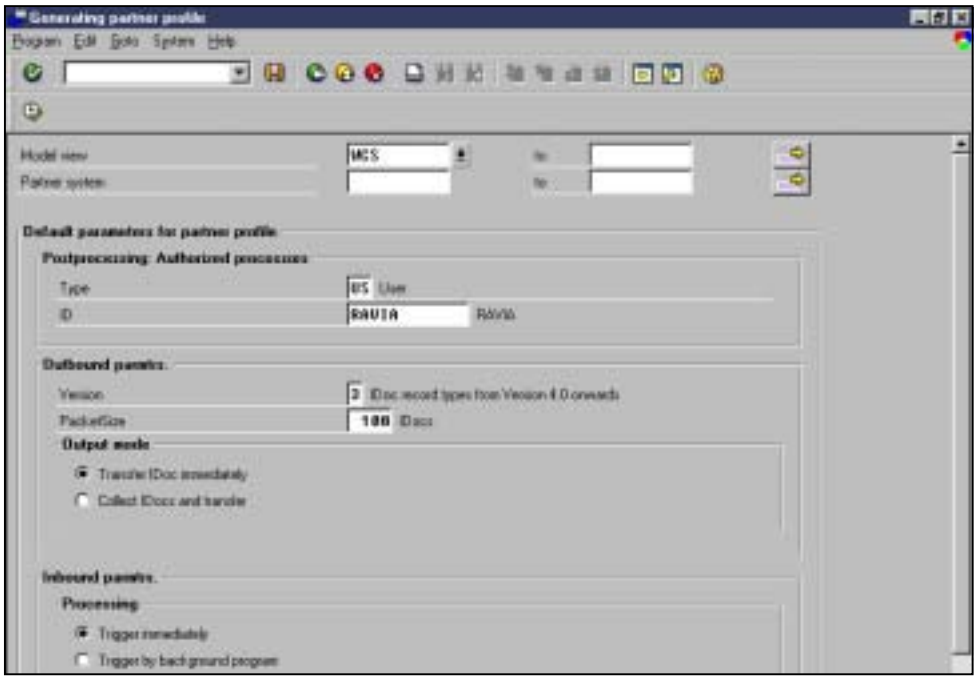

Figure 58: Generating a partner profile: choosing the model view

3. Click on the execution icon to run the program.

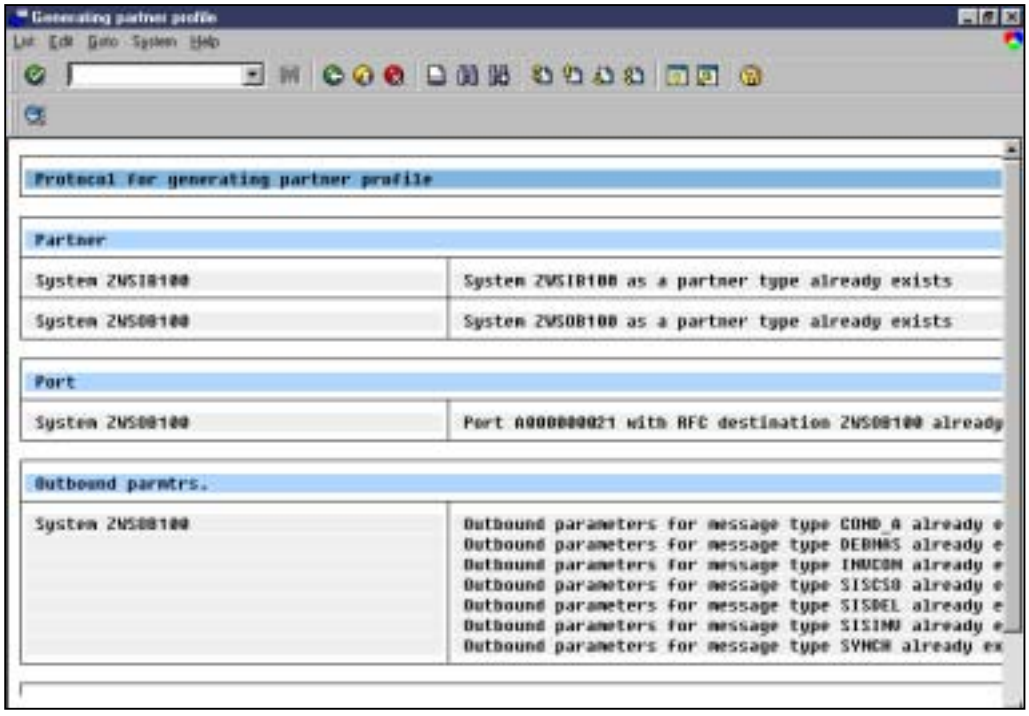

Figure 59: Protocol for generating a partner profile

Modifying partner profiles

1. Go to transaction WE20.

2. Select the logical system, as shown in Figure 60.

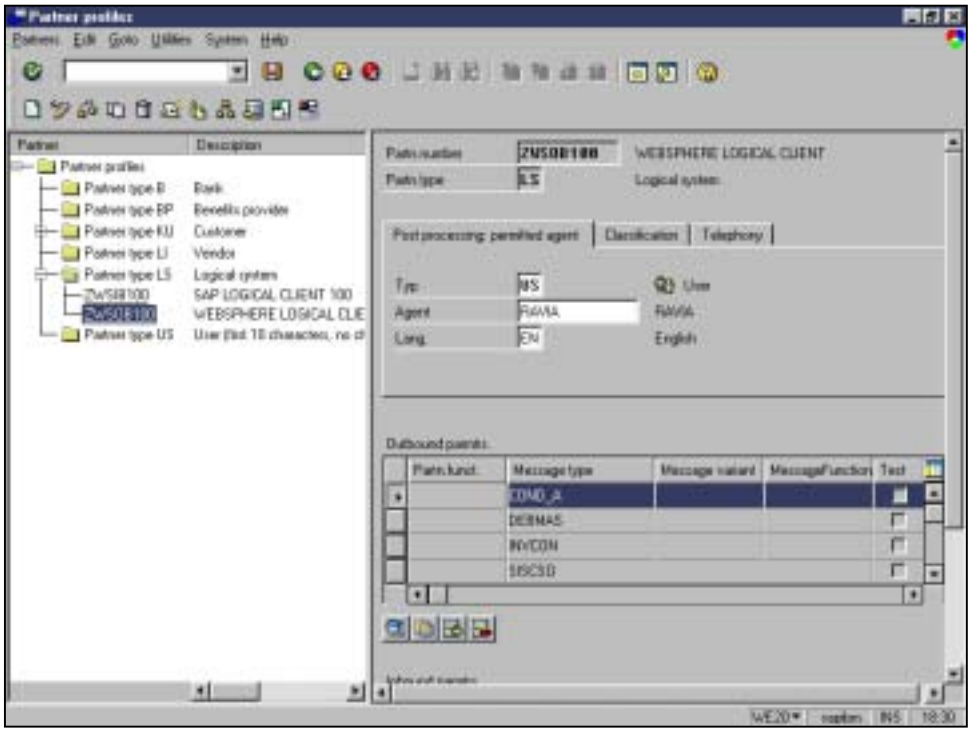

Figure 60: Modifying a partner profile: Selecting the logical system

3. Select the required outbound or inbound parameters.

4. Click **Change**.

5. Check the receiver port.

6. Make changes if required, as shown in Figure 61.

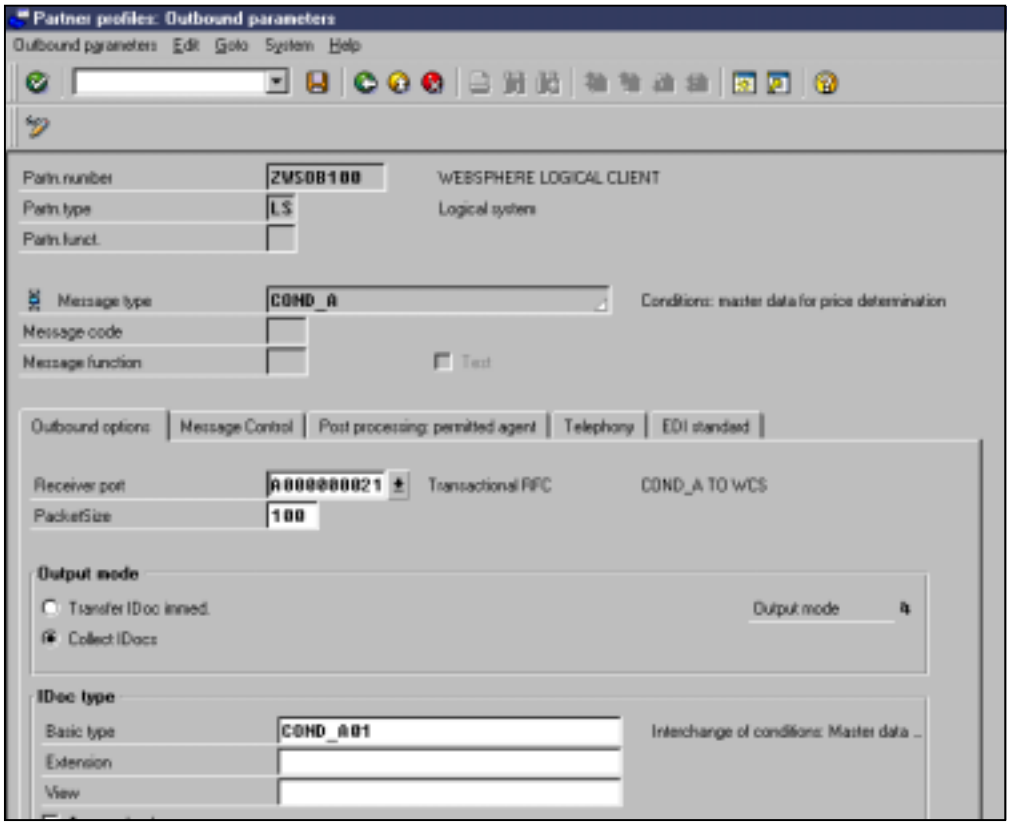

Figure 61: Modifying a partner profile: Changing the receiver port

7. Enter the **PacketSize** and select the **Output mode**.

8. Click **Save**.

Customizing workflow

1. Go to transaction ALE and select **Error Handling** > **Basic Workflow Settings**, as shown in Figure 62.

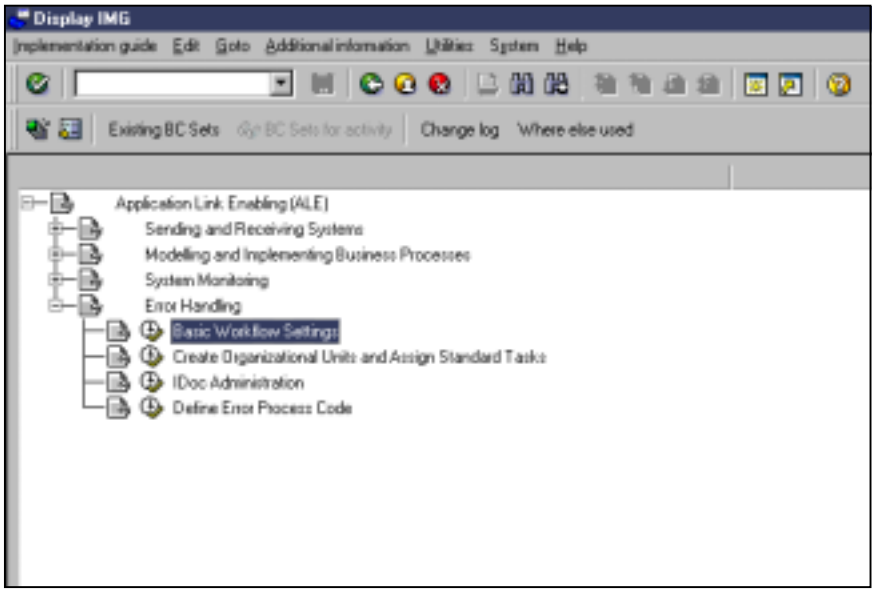

Figure 62: Selecting **Basic Workflow Settings**

The workflow customization page displays, as shown in Figure 63.

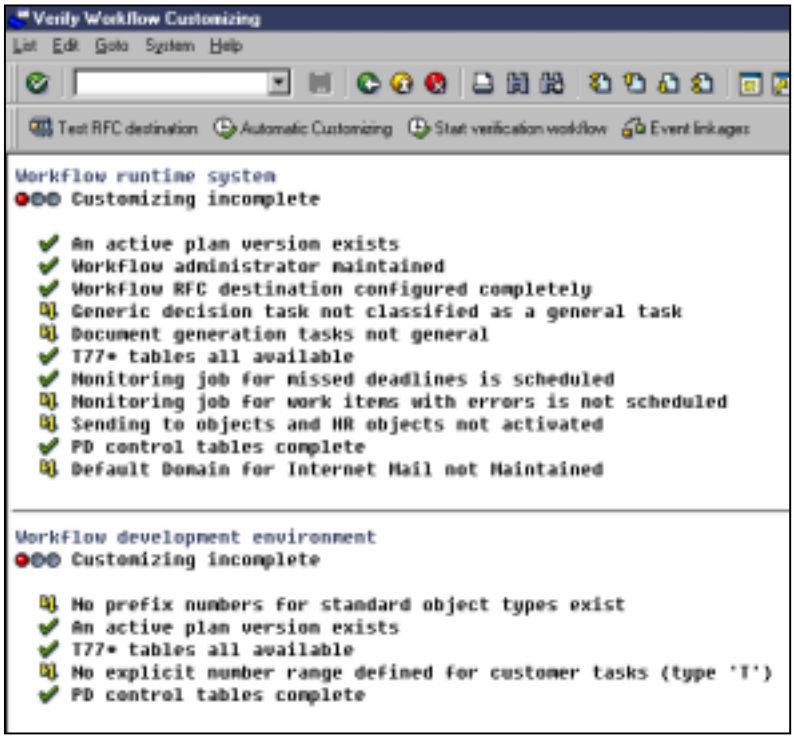

Figure 63: Verifying workflow customization

2. Click **Save**.

Enabling change pointers

1. Go to transaction BD61.

2. Select the check box and save as shown in Figure 64.

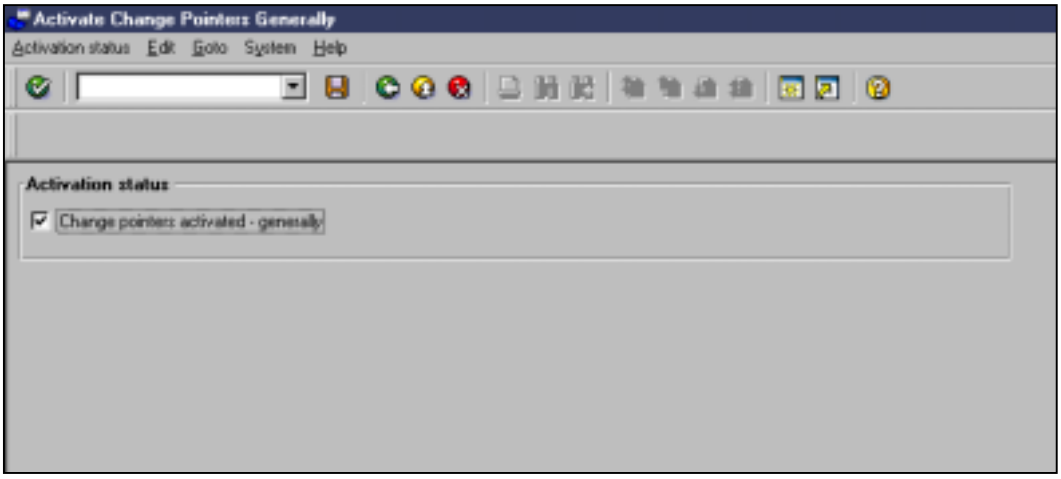

Figure 64: Enabling change pointers

Enabling change pointers for specific message types

1. Go to transaction BD50 and enable the change pointers (for example, DEBMAS and COND_A), as shown in Figure 65.

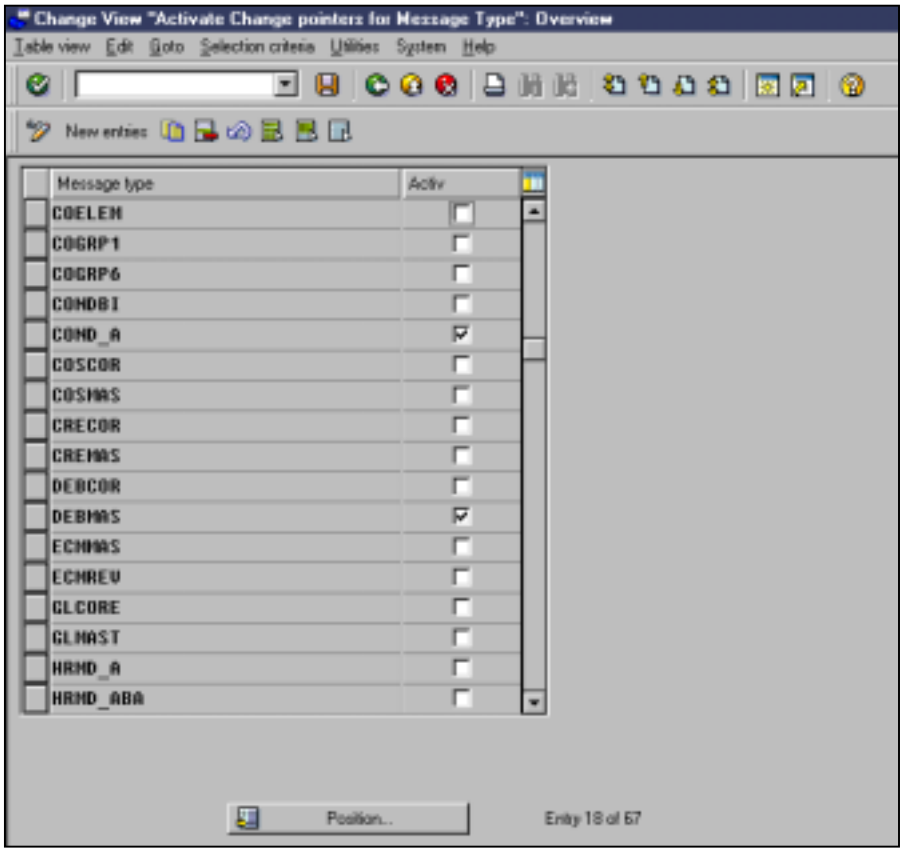

Figure 65: Enabling change pointers for message types

2. Click **Save**.

Generating outbound IDocs from change pointers

1. Go to transaction SE38. Type `RBDMIDOC` in the program field, as shown in Figure 66.

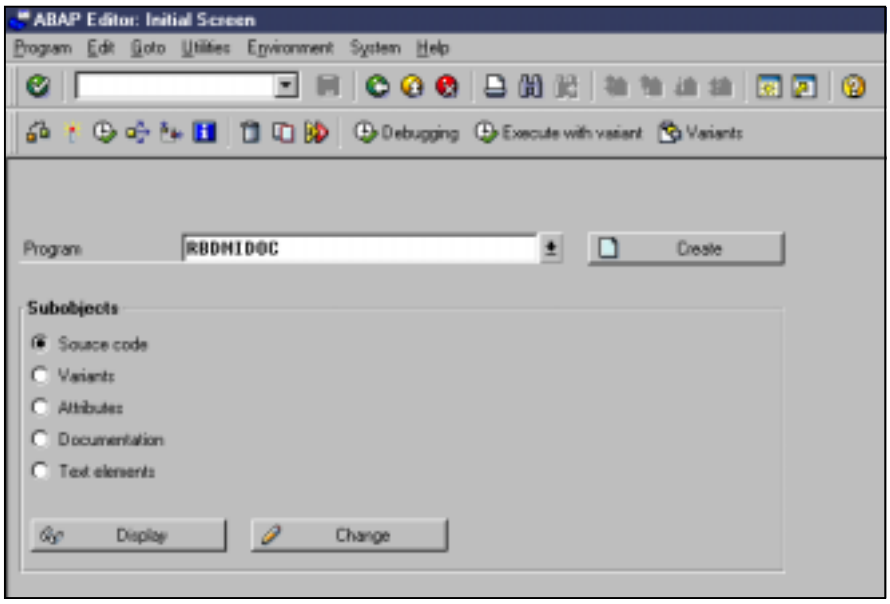

Figure 66: Running the RBDMIDOC program

Create DEBMAS_DOCS and COND_A_DOCS variants (select the background option), as shown in
Figure 67.

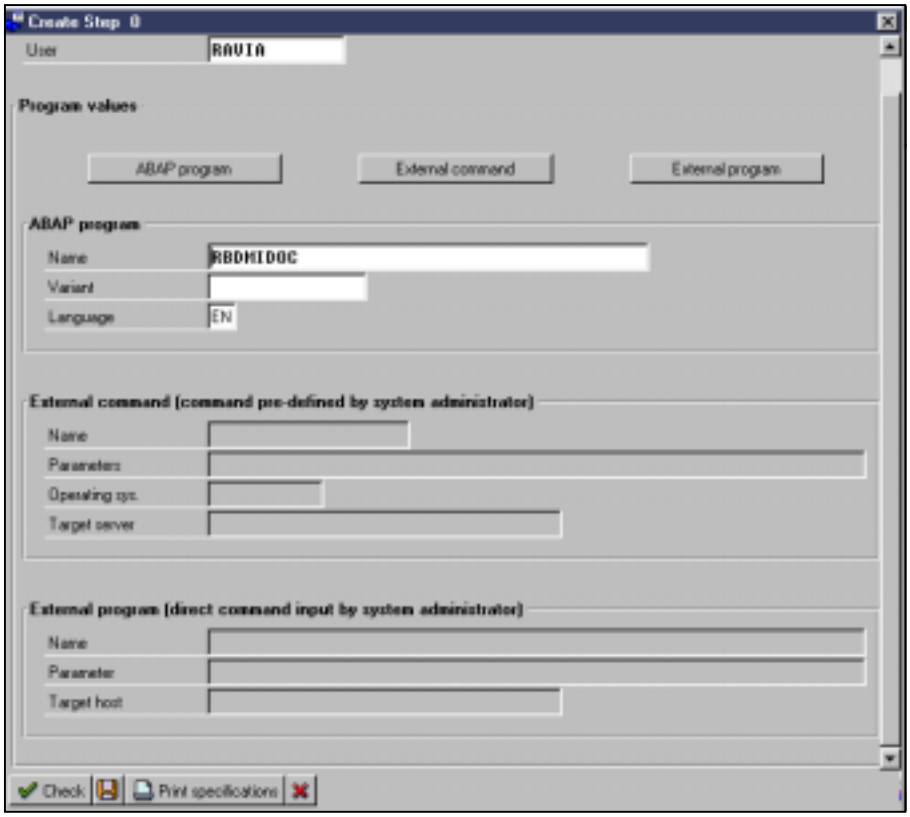

Figure 67: Creating variants

2. Go to transaction SM36
3. Create a job ECE_GENERATE_IDOC_FROM_CHG_PNTRS.
4. Type the JobClass and target system. Type all other settings.
5. In **Job Step** type the program name and variant, and schedule the job.

Dispatching IDocs

1. Go to transaction SE38. Type RBDOUTPU as the program name, as shown in Figure 68.

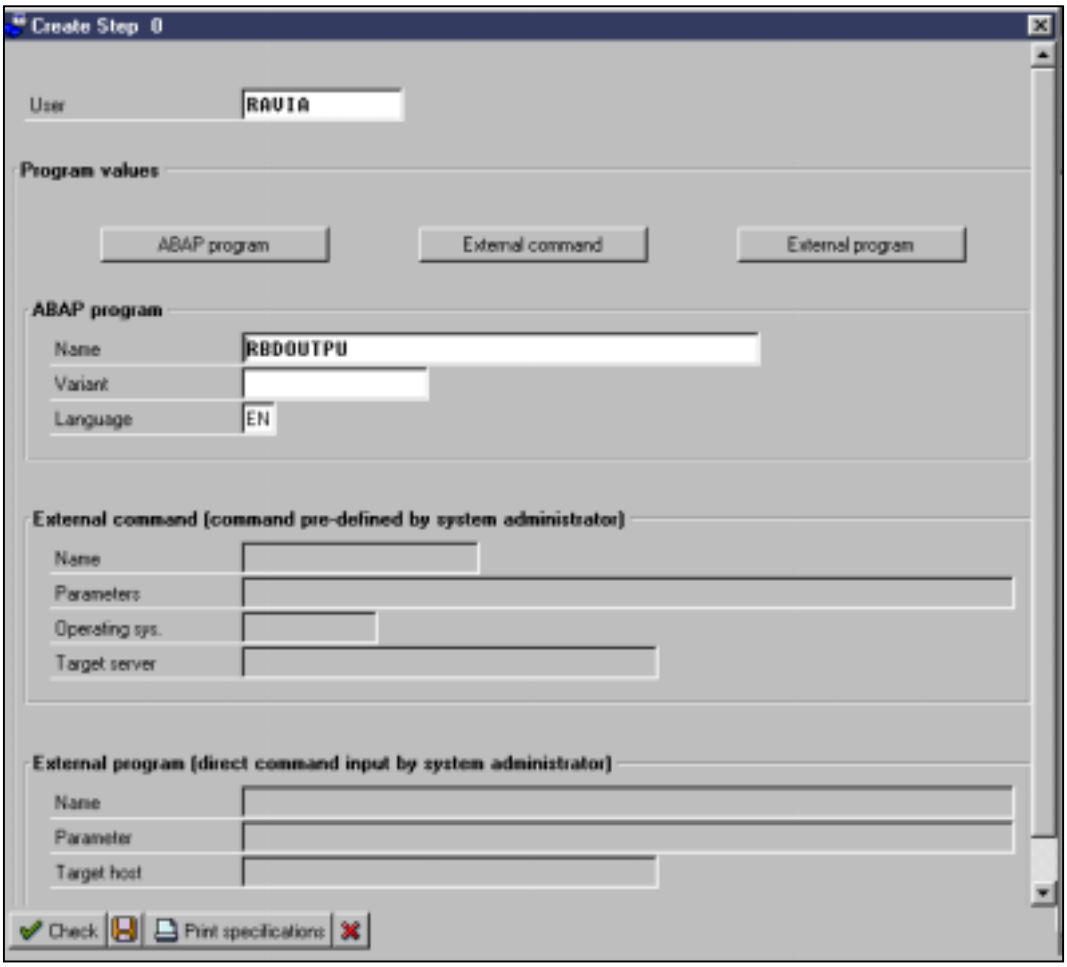

Figure 68: Running the RBDOUTPU program

2. Create a variant schedule for the job.

Transactions - master data

Create the following master data, condition records:

- Pricing records – ZPR0
- Shipment cost – ZF00
- Tax - UTXJ

Pricing records

To create the pricing records use transaction VK11 and do the following:

1. From the Create Condition Records initial screen, enter the **Condition type** as ZPR0 and press enter. A pop-up window appears listing the tables in

which the condition records can be stored. The first table is for the customer group records, the second table is for customer or material specific prices, the third table is for pricelist and the fourth table is the for material price.

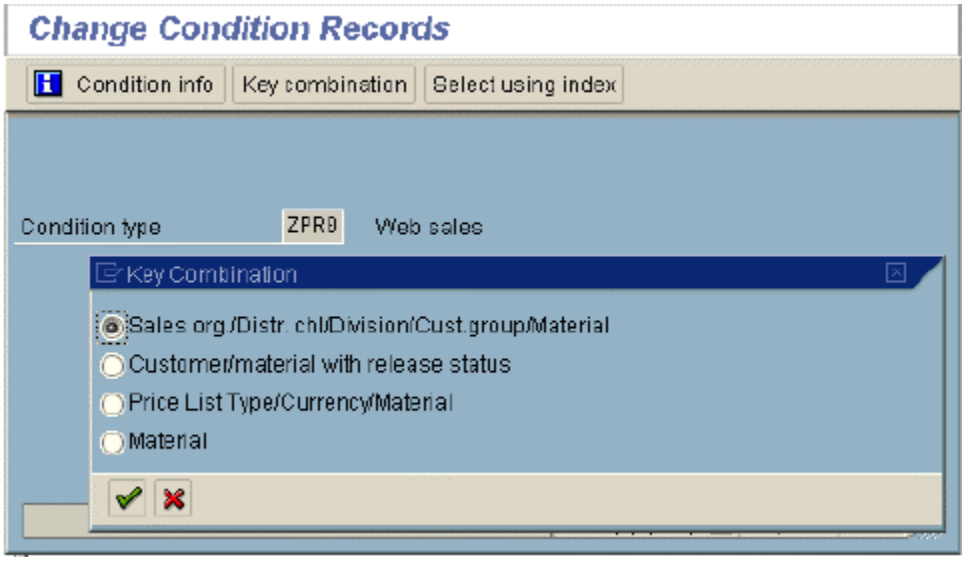

Figure 69: Creating the condition records for pricing

a. Select the last table and press enter. Make the required entries as shown in Figure 70 and save the records.

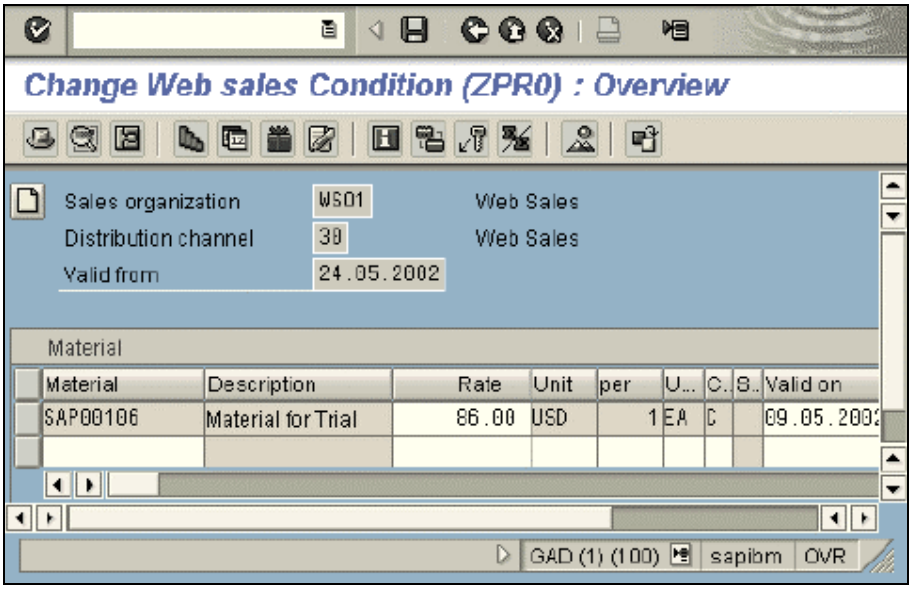

Figure 70: Creating the Web sales condition

b. To maintain records for the other tables, select the respective table from the first screen and create the records.

Shipment cost

To maintain records for the shipping cost, enter the condition type ZF00 in the Create Shipment cost Condition initial screen, press enter, and do the following:

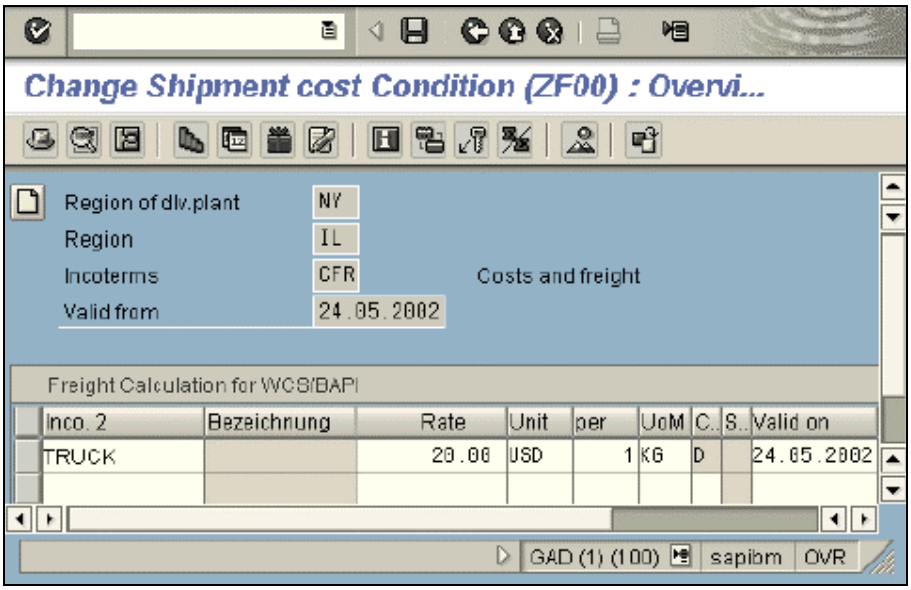

Figure 71: Creating a record for shipment cost

1. Enter the **Region of dlv. plant**, which is the delivering plant.
2. Enter the customer's **Region**.
3. Enter the **Incoterms.**
4. Enter the mode of transport in uppercase for Incoterms2.
5. Enter the rates for currency, unit and the validity dates. Save the record.

Tax

To maintain records for tax do the following:

1. Enter the condition type UTXJ in the Create Condition Records initial screen and press enter.
2. Select the first table as shown in the following figure to maintain the records and press enter.

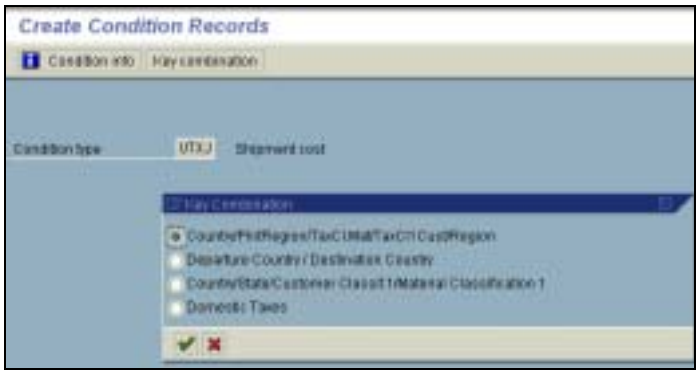

Figure 72: Creating a tax record

3. Enter the **Country**, **Region of div.plant**, **Tax class.material**, **TaxClass1-Cust**, which is the tax classification of the customer, region of the customer, and the tax code. Press enter and save the records

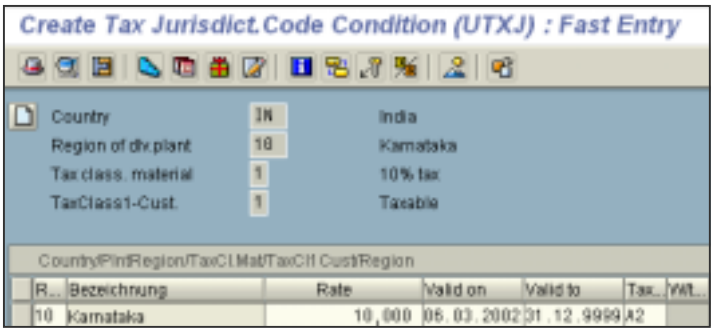

Figure 73: Creating a tax record

Chapter 8. Loading master data

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

- Loading SAP materials and customer data onto WebSphere Commerce.
- Loading reference data onto SAP.

Loading SAP materials and customer data onto WebSphere Commerce

To synchronize material data between WebSphere Commerce and SAP, use catalog management utilities like TextTransformer, XMLTransformer, IdResolver and Massloader. For more information refer to the WebSphere Commerce loader utility documentation.

To load files onto WebSphere Commerce you must do the following:

- Extract material data from SAP into a pre determined flat file.
- Transform the data to an XML as expected by the catalog management utilities.
- Resolve the XML for all primary key constraints and upload to WebSphere Commerce.

Extracting data from SAP

To extract data from SAP R/3 containing material and customer master data**,** you need to write report programs in the ABAP editor (transaction code se38) and 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”. For more information see Appendix D: Sample extract program .

Loading material data

To load the material data onto WebSphere Commerce using the massload scripts provided, ensure that you extract the material data from the SAP R/3 system in the following sequence with a delimiter. The delimiter expected by the massload script is "~". If you are using a different delimiter, you need to change this 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)
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)

Sequence no.	Data at position	Data description	Corresponding fields in SAP
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 which 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)
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 format with the delimiter "~". If you are using a different delimiter, you need to change this 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)

Sequence No.	Data at position	Data description	Corresponding fields in SAP
		characteristic name	(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 `ManifestFile.txt` present in the `store\dataload\material` directory, and replace `itabmara.txt` with the output file name derived from the material information extract program. Replace `itab.txt` 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` available in the integration package that you have downloaded.

3. Save and close this file.

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

   `DB_NAME` - Database type ‘db2’.

   `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:

   `<Drive:>\`*`Commerce_Install_Path`*`\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 store owner (ORGENTITY.ORGENTITY_ID).

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

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

   `ImportLocation=<Drive:>\`*`Commerce_Install_Path`*`\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 present 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:

`Application_server_install_path``\installedApps\WC_Enterprise_App_`*`Instance_Name.`*`ear\wcstores.war\`*`Store_Name`*`\images.`

Where, *`Instance_Name.`*`ear` is the name of the commerce instance in your installation and *`Store_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. Ensure that the customer data is in the following sequence and delimited by “~”. If you are

using a different delimiter, you need to change this 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)
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)

Sequence. No.	Data field	Data description	Corresponding fields in SAP
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` present in the `store\dataload\customer` directory, and replace `itabkna2.txt` with the output file name derived from the customer information extract program.

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

   `DB_NAME` - Database type 'db2'.

   `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:

   `<Drive:>\`*`Commerce_Install_Path`*`\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=<Drive:>*Commerce_Install_Path*\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:

   <Drive:>*Commerce_Install_Path*\bin\setenv.bat

8. Save all the changes made and run the batch file CustomerUpload.bat 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 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, sqllibDir. Ensure that the path settings specified in MaterialUpload.bat or CustomerUpload.bat 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.

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

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

4. The username and/or password supplied is 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` is not found or the file name specified for `ID_PROPERTIES_File` in `MaterialUpload.bat` or `CustomerUpload.bat` 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` present 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 pre-requisite to load material data is to define the material groups in the SAP R/3 system. 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 customizing
10	General Item Category Group	MTPOS_MARA	4	Yes
11	Gross Weight	BRGEW	17	No

Serial number	Description	Data element as in SAP	Length	Depends on existent SAP configurations and data
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
32	Period Indicator	PERKZ	1	No

Serial number	Description	Data element as in SAP	Length	Depends on existent SAP configurations and data
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` present 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.

The pre-requisite to load this data is to define the price condition in SAP system.

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 MM01 transaction and then record VK12 transaction.

2. Based on the values in your SAP installation, make the necessary changes to the data in the reference data files before importing.

3. Use transaction SE38 to execute the recorded programs. See Appendix E: Sample BDC program for a sample import BDC program.

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

Chapter 9. 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

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/2000.

WMQI services

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

WebSphere Commerce instance

You can start the WebSphere Commerce instance from the Administrative 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 *`MQ_link_installation_path`*`\bin`

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

```
smqsi –iin.ini
```

Outbound server

1. Copy the `out.ini` file from the `samples` directory to *`MQ_link_installation_path`*`\bin`

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

```
smqso –iout.ini
```

Testing individual messages

OrderCreate 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 the SAP system.

2. Log in to the WebSphere Commerce store as a customer, using a valid user ID.

3. Add items to the shopping cart and submit an order. This must generate the XML order create message. It may take some time to deliver the message, as determined by the scheduler configuration in WebSphere Commerce. By default, the scheduler process runs at an interval of 5 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 passed through WMQI but encounters an error in the R/3 link inbound server processing, then details of this will be placed in the error file associated with the inbound server.

OrderConfirmationStatus 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, the order confirmation status message is sent by SAP. 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 entries are made in the ORDSTAT and ORDISTAT tables.

3. Login to the SAPToolTech store. Click on the **Order Status** menu and look for the order in the Order(s) Confirmed section.

Any 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.

OrderDeliveryStatus 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 generated by SAP. This can be ascertained from the SISCSO01 IDoc that was generated earlier. If the order creation was not successful, 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 unless an error occurs. The SAP client in the error log shows the errors.

5. Login to the SAPToolTech store. Click on the **Order Status** menu and look for the order in the Order(s) 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 order status messages to be versioned. Depending on the option selected, either the existing status record will be updated or a new record will be added to these tables. By default, the order status header and the order status item are not versioned.

ProductInventoryUpdate 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 earlier. Another screen showing the details of the order displays.

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.

OrderInvoiceStatus 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 create SISINV01 IDoc. When vf01 runs a page is displayed asking 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 is not displayed, select the document number from the menu.

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

5. Login to the SAPToolTech store. Click on the **Order Status** menu and look for the order in the Order(s) Invoiced section.

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

ProductPriceUpdate 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, enter `ZPR0`. When the transaction runs, you will be asked for the key combination.

2. Based on your preference, select the option. Enter the required fields on the next screen, and run the transaction.

3. On the next page, change the rate for a 'WCBE-known' material and save the document.

4. Run SAP transaction se38 to generate the IDocs.

5. Enter the program name as `RBDMIDOC`. Run the program.

6. On the next page, enter 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 program RBDOUTPU.

8. Select **dispatch** on the next page and run the program.

9. On the next screen type `COND_A02` as the basic type and run the program. This should dispatch the COND_A02 IDOC.

10. Verify the change in price by checking the OFFERPRICE and OFFER table in WebSphere Commerce Business Edition.

11. Login to the SAPToolTech store. Check the price of the order item after adding it to the shopping cart.

CustomerNew/Update message (DEBMAS05)

To check if the customer new/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. You need to run SAP transaction se38 to generate the IDoc. Run the program `RBDMIDOC`.

3. On the next page enter the message type as `DEBMAS` and run the program. This creates the IDoc but does not send it.

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

5. Select **dispatch** on the next page and run the program.

6. Type DEBMAS05 as the basic type on the next page 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. Login to the SAPToolTech store. Click on the **Account** menu 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 8. Loading master data.

2. Modify the ManifestFile.txt and MaterialUpload.bat as mentioned in Chapter 8. Loading master data.

3. Run the `MaterialUpload.bat` 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 8. 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 and CustomerUpload.bat as mentioned in Chapter 8. Loading master data and run the `CustomerUpload.bat` from a DB2 command window.

4. To verify, login to the SAP ToolTech 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.

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

Chapter 10. 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. It provides the necessary interfaces to extend the current support for adding new inbound and outbound messages.

Inbound message into WebSphere Commerce

Use the `user_template.xml` 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. Execute the following steps to add 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 e.g. MATMAS in SAP R/3 is for Materials Master.

2. Configure SAP R/3 system to generate and send the IDoc message. This should happen whenever a new material or product is added or a current material or product attribute has changed.

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

4. WebSphere Commerce Business Edition 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. From the **Start** menu go to **Programs**, **IBM WebSphere Commerce** and select **Configuration**. Expand InstanceList node and select the relevant *`instance_name`*. Select the **Messaging** node under *`instance_name`* \Instance Properties node. After selection, on the right side frame add the details of the new DTD under the Inbound Message DTD files parameter.

5. Provide information in the `user_template.xml` 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 `sys_template.xml` for similar entries of the messages that are currently supported by WebSphere Commerce Business Edition.

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

7. Import the XML DTD and 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 Business Edition.

Outbound message from WebSphere Commerce

To add a new outbound message, write a new controller 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 Business Edition online documentation. Given below is the sample for the customer or user registration message. The same steps can be used to create a new message.

1. Add a new controller 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 is supported by the SAP R/3 system.

4. Import the DTD and IDoc C-header into WebSphere MQ Integrator 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 as the ESQLs add the SAP header information.

Chapter 11. Store customizations

This reference application is built on top of the ToolTech store model. The reference store contains catalog data and web assets in the English language only. For more information, refer to the Store Developer's Guide. The following changes are made to the ToolTech store 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`, `offer.xml`, and so on.

Disable ATP: `Store.xml` is modified to change the ALLOCATIONGOODFOR to 0.

Inventory: `Storefulfill.xml` is modified to populate the inventory details for the reference catalog items.

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

Address page: Modification to `Address.jsp` has been made to provide a selection box to select the country and state codes.

Shipping page: Modifications to Shipping.jsp are made to use `OrderDisplay` as the redirection URL instead `AllocationCheck`.

Order Display Pending and Order Confirmation pages: The `OrderDisplayPending.jsp` and `Confirmation.jsp` are modified not to include subtotals, tax, and shipping details.

Track Order Status: Modifications to the `TrackOrderStatus.jsp` allows 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: Modification to `OrderDetail.jsp` displays 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 TRUCK, RAIL, AIR respectively, and shipping carrier 'XYZ Carrier' with CFR. en_US/Shipping.xml is modified to set the description for the above shipping modes to 'CFR-TRUCK', 'CFR-RAIL', and 'CFR-AIR' respectively. These shipping modes can be used while placing the order using 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 terms and conditions. For more details, refer to the WebSphere Commerce documentation.

Appendix A: 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 is 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.

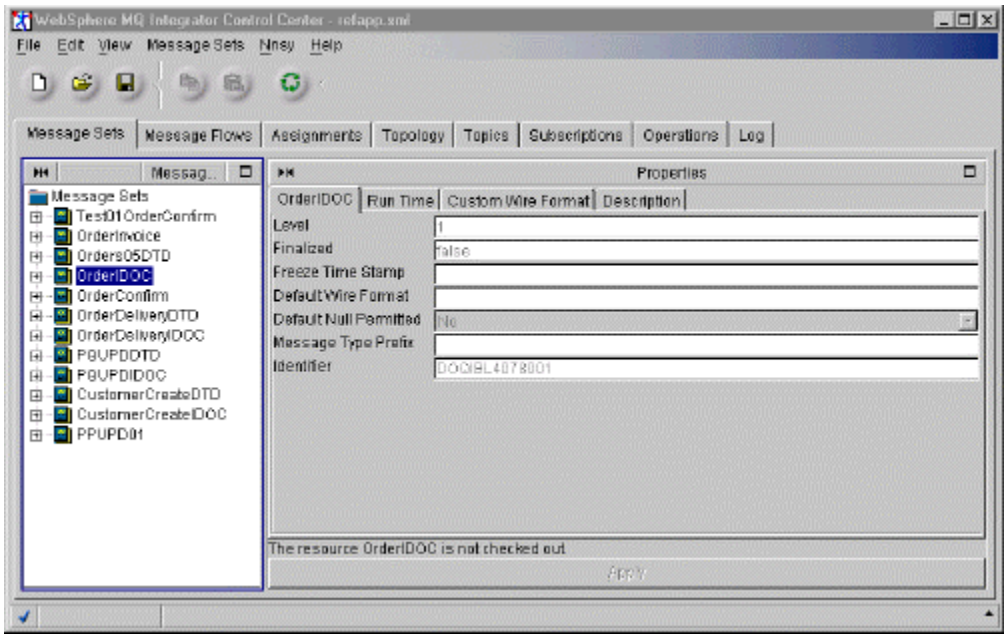

Figure 74: WMQI message sets

Appendix B: 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.

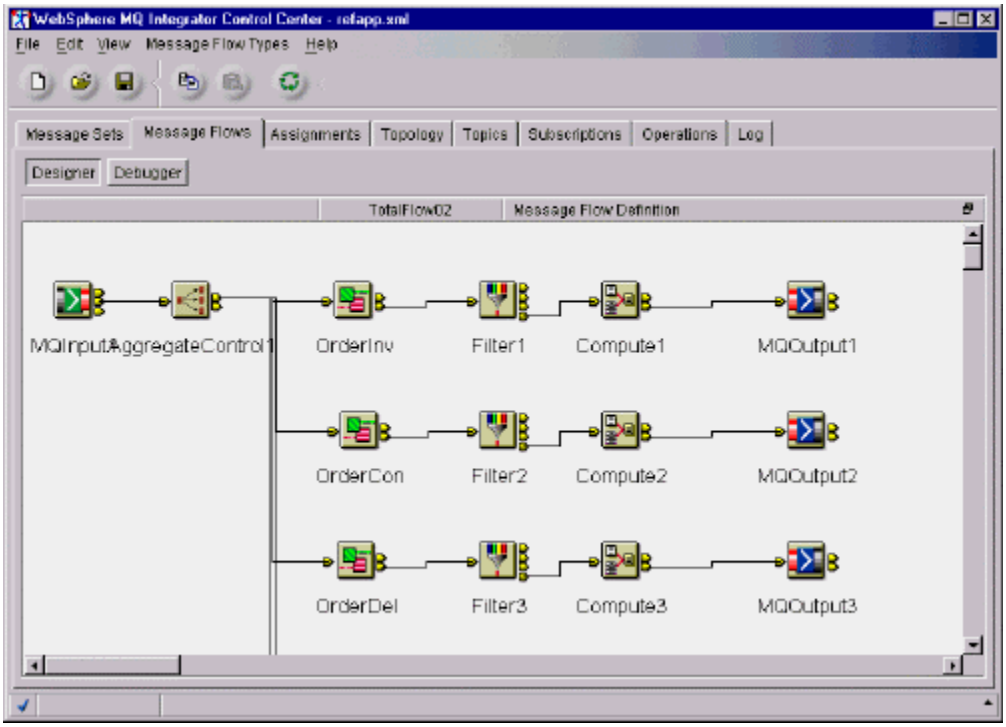

Figure 75: WMQI message flows

Appendix C: 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		Defaulted to SB		

XML element	WebSphere Commerce table name	Database column	Length	Description	IDoc segment	IDoc field	Length	Field description
				SB - default (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

XML element	WebSphere Commerce table name	Database column	Length	Description	IDoc segment	IDoc field	Length	Field description
Profile	USERS	PROFILETYPE	2	C (base profile data), B (business profile data)		Defaulted to "C"		
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

XML element	WebSphere Commerce table name	Database column	Length	Field description	IDoc segment	IDoc field	Length	Field description
Placed Date	ORDSTAT	OSPLTIME	TIMESTAMP	Order placed timestamp.	E1CVBAK	AUDAT	8	Document date
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)

XML element	WebSphere Commerce table name	Database column	Length	Field description	IDoc segment	IDoc field	Length	Field description
Status	ORDSTAT	OSSTATUS	32	Default value of 'C' is set by the command				
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

XML element	WebSphere Commerce table name	Database column	Length	Field description	IDoc segment	IDoc field	Length	Field description
TotalPriceInfo (currency attribute)	ORDISTAT	OICPCUR	10	Currency in which the price is expressed	E1CVBAP	WAERK	5	Document currency
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				

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
PlacedDate	ORDSTAT	OSPLTIME	25	OrderPlaced Date				
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

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
ConfirmedQuantity	ORDISTAT	OIQTCONFIRM	10	Quantity of items confirmed	E2CLIP2	APKBMENG	18	Cumulative confirmed quantity in sales 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				

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
PlacedDate	ORDSTAT	OIPLTIME	25					
Shipping Info Attribute Shipping condition	ORDISTAT	OISCOND	2	Code indicating if partial shipment of the order is accepted	E2CLIPS	KZTLF	1	Partial delivery at item level
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				

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
Status	ORDSTAT	OSSTATUS		Defaulted to 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								

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
ItemNumberByBackend	ORDISTAT	OIMITEM	30	Order item number generated by the backend system	E2CVBRP	POSNR	6	The number that uniquely identifies the item in the 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	E2CVBRP	KZWI4	15	Subtotal for the pricing condition

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
				order item				pricing condition
TotalTaxOnShippingPrice	ORDISTAT	OISHTXTOT	21	Total tax on shipping charges for the order item	E2CVBRP	KZWI6	15	Subtotal for the 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
			company zip 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						

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
ContactInfo	NA						
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							

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
AddressLine				E1EDKA1	STRAS	35	Street and house 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

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
RequisitionerInfo				E1EDPA1	PARVW='AG' (Sold-To Party)	3	Partner function
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

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
ContactPersonName	ADDRESS						
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						

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
ScheduledShipDate	Na						
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

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
Method	SHIPMODE	CODE	A merchant assigned code	E2EDK17	LKTEXT when INCOTERMS QUALF = '002'	70	Delivery condition text
BuyOrgAccountingDetail							
Percentage							
MonetaryAmount							
Currency(Attribute)							
BudgetCode							
Description							
CalculationCode							
OrderCustomerField	ORDERS	Field1					
OrderCustomerField	ORDERS	Field2					
OrderCustomerField	ORDERS	Field3					
UserData							
UserDataField name Attribute							
UserDataField							
ReportPOItem							

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
ItemLineNumber	Na						
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

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
Currency(attribute)							
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							

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
Title							
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

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
Telephone	ADDRESS	PHONE2		E1EDPA1	TELF2	25	Telephone number 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		

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
					ORDER DATE)		
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							

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

XML element	WebSphere Commerce table name	Database column	Description	IDoc segment	IDoc field	Length	Field description
RequestedShipDate							
ScheduledShipDate							
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		

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
TradingPositionContainerID	TRADEPOSCN	TRADEPOSCN_ID	8	TradingPosition ContainerID		Get the value of TradingPositionContainerID 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)		

Appendix D: Sample extract program

The following is a sample program to extract the material number and material group from an SAP system:

```
REPORT ZRAYMARA LINE-SIZE 300 NO STANDARD PAGE HEADING.

TABLES:
  MARA,             "General Material Data
  MARC.
SELECT-OPTIONS:
  MATNUM FOR MARA-MATNR.          "Material number

DATA:
  BEGIN OF ITAB OCCURS 0,
    MATNR LIKE MARA-MATNR,         "Material number
    MATKL LIKE MARA-MATKL,         "Material group
  END OF ITAB,
  BEGIN OF ITAB1 OCCURS 0,
    MATNR LIKE MARA-MATNR,         "Material number
    DEL1 VALUE '~',
    MATKL LIKE MARA-MATKL,         "Material group
  END OF ITAB1.

SELECT MATNR MATKL INTO CORRESPONDING FIELDS OF TABLE ITAB
FROM MARA

WHERE MATNR IN MATNUM.
LOOP AT ITAB.
  MOVE-CORRESPONDING ITAB TO ITAB1.
  APPEND ITAB1.
  CLEAR ITAB.
ENDLOOP.
CALL FUNCTION 'DOWNLOAD'
     EXPORTING
          CODEPAGE                = 'IBM'
          FILENAME                = 'c:\itabmara.txt'
          FILETYPE                = 'ASC'
     TABLES
          DATA_TAB                = ITAB1
     EXCEPTIONS
          INVALID_FILESIZE        = 1
          INVALID_TABLE_WIDTH     = 2
          INVALID_TYPE            = 3
          NO_BATCH                = 4
          UNKNOWN_ERROR           = 5
          GUI_REFUSE_FILETRANSFER = 6
          CUSTOMER_ERROR          = 7
         OTHERS                  = 8.
```

Appendix E: Sample BDC program

The following is a sample BDC program to import data into SAP system:

```
report zraybdcvk12 .

data :

  bdcdata like bdcdata occurs 0 with header line,

  messtab like bdcmsgcoll occurs 0 with header
line,

  pid like rmmg1-matnr,

  begin of line_itab,

     kschl(4)," like rv13a-kschl,

     low(18)," like a118-matnr,

     kbetr(14)," like konp-kbetr,

     konwa(5)," like konp-konwa,

  end of line_itab,

  itab like standard table of line_itab with header
line,

  begin of upload_line,

    data(285) type c,

  end of upload_line,

  upload_itab like standard table of upload_line
with header line.

parameters:

  file like rlgrap-filename default
'C:\TESTBDC5.TXT',

  delim(1) type c default '~'.



call function 'UPLOAD'

     exporting

          filename                = file
```

```
          filetype                = 'ASC'

     tables

          data_tab                = upload_itab.

if sy-subrc <> 0.

  message id sy-msgid type sy-msgty number sy-msgno

          with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.

endif.

loop at upload_itab.

  if not upload_itab is initial.

    split upload_itab-data at delim into itab-kschl
itab-low itab-kbetr

  itab-konwa.

  append itab.

  endif.

  clear itab.

  clear upload_itab.

endloop.

loop at itab.

check not itab is initial.

  perform chngscr using 'SAPMV13A' '0100'.

  perform chngfld using 'BDC_OKCODE' '/00'.

  perform chngfld using 'RV13A-KSCHL' itab-kschl.

  perform chngscr using 'SAPLV14A' '0100'.

  perform chngfld using 'BDC_CURSOR' 'RV130-
SELKZ(04)'.

  perform chngfld using 'BDC_OKCODE' '=WEIT'.

  perform chngfld using 'RV130-SELKZ(01)' ' '.

  perform chngfld using 'RV130-SELKZ(04)' 'X'.

  perform chngscr using 'RV13A004' '1000'.
```

```
  perform chngfld using 'BDC_OKCODE' '=ONLI'.

  perform chngfld using 'F003-LOW' itab-low.

  perform chngfld using 'F001' 'WSO1'.

  perform chngfld using 'F002' '30'.

  perform chngfld using 'SEL_DATE' sy-datum.

  perform chngscr using 'SAPMV13A' '1004'.

  perform chngfld using 'BDC_OKCODE' '/00'.

  perform chngfld using 'KOMG-VKORG' 'WSO1'.

  perform chngfld using 'KOMG-VTWEG' '30'.

  perform chngfld using 'KOMG-MATNR(01)' itab-low.

  perform chngfld using 'KONP-KBETR(01)' itab-
kbetr.

  perform chngfld using 'KONP-KONWA(01)' itab-
konwa.

  perform chngscr using 'SAPMV13A' '1004'.

  perform chngfld using 'BDC_OKCODE' '=SICH'.

  perform calltran using 'VK12'.

  clear itab.

endloop.

form chngscr using    p_prog

                      p_dynpro.

  clear bdcdata.

  bdcdata-program = p_prog.

  bdcdata-dynpro = p_dynpro.

  bdcdata-dynbegin = 'X'.

  append bdcdata.

endform.

form chngfld using fnam fval.

  clear bdcdata.
```

```
  bdcdata-fnam = fnam.

  bdcdata-fval = fval.

  append bdcdata.

endform.                    " CHNGFLD

form calltran using    tran.

  refresh messtab.

  call transaction tran using bdcdata

                   mode 'E'

                   update 'L'

                   messages into messtab.

  if sy-subrc eq 0.

    get parameter id 'MAT' field pid.

    write : / 'The material Number AFFECTED is
',pid.

  elseif sy-subrc <> 0.

    message i001(zray) with sy-subrc 'Program not
successful'.

    exit.

  endif.

endform.                    " calltran
```

Notices and trademarks

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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

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

IBM	DB2 Extenders
AIX	DB2 Universal Database
DB2	MQSeries
VisualAge	WebSphere

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

Solaris, 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, R/3, R/2, and mySAP.com are trademarks or registered trademarks of SAP AG.

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

Credit card images, trademarks and trade names provided in this product should be used only by merchants authorized by the credit card mark’s owner to accept payment via that credit card.