



Integration Guide

for the IBM CrossWorlds® System

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

First Edition (July 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.

© Copyright International Business Machines Corporation 2002. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Contents	3	Appendix B. Supported generic business objects.....	85
Before you begin.....	5	Appendix C. Supported application specific business objects	87
Conventions used in this guide	6	Appendix D. Collaboration templates.....	89
Chapter 1. Introduction.....	7	Appendix E. Maps with specific values for WebSphere Commerce and SAP	91
Terminology used in this book.....	7	Appendix F. Binding maps to business objects.....	93
Overview.....	7	Appendix G. Sample BDC program	95
References	9	Notices and trademarks.....	99
Chapter 2. Pre-requisites.....	11		
Software pre-requisites	11		
Hardware pre-requisites	11		
Chapter 3. Integrating WebSphere Commerce business processes with SAP R/3.....	13		
WebSphere Commerce outbound process	13		
WebSphere Commerce inbound process	14		
WebSphere Commerce Request/Reply process	15		
Chapter 4. Installing and configuring.....	17		
Sample topology.....	17		
Installing and configuring WebSphere Commerce	17		
Installing the Commerce Enhancement Pack ..	18		
Installing and configuring WebSphere MQSeries.....	18		
Configuring the WebSphere Commerce messaging system	19		
Installing and configuring CrossWorlds.....	23		
Configuring the CrossWorlds queue manager ..	27		
Configuring the CrossWorlds adapter in WebSphere Commerce	28		
Configuring SAP.....	29		
Organization structure	29		
Customizing the master data.....	36		
Configuring price, freight, and tax	42		
Customizing order numbers	47		
Configuring partner profile communication	48		
Transactions - master data.....	65		
Before publishing the store.....	68		
Publishing a store	69		
Chapter 5. Store customizations	77		
Chapter 6. Verification procedure	79		
Appendix A. Supported messages.....	83		
Asynchronous messages	83		
Synchronous messages.....	84		
Enabling the message flow	84		

Before you begin

The *Integration Guide for the IBM CrossWorlds System* is intended for those who wish to integrate WebSphere Commerce Version 5.4 with backend systems like SAP R/3 using the IBM CrossWorlds system. This guide will assist developers and engagement teams developing an integrated solution. Additionally, demonstrators or marketing personnel who want to demonstrate this functionality can use this book.

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

This guide contains information about the integration of WebSphere Commerce Version 5.4, Business Edition with the SAP R/3 4.6 Enterprise Resource Planning (ERP) system using the CrossWorlds system. In place of WebSphere Commerce Version 5.4 Business Edition you can also use WebSphere Commerce Version 5.4 Professional Edition. It provides information on how asynchronous and synchronous messages can be exchanged between WebSphere Commerce and SAP through CrossWorlds. It gives an overview of a typical end-to-end flow between the two systems, installation and configuration of the various components, and pointers to the CrossWorlds maps and collaborations. This book is divided into the following sections:

The samples provided with this integration solution are for WebSphere Commerce Version 5.4, Business Edition. However, the same principal and pattern can apply to WebSphere Commerce Version 5.4, Professional Edition too.

Chapter 1. Introduction A brief overview of the integration of WebSphere Commerce Version 5.4, Business Edition with SAP R/3 using the CrossWorlds system as well as the definition of the terms used in this book, and references to other related documents.

Chapter 2. Pre-requisites A list of the software and hardware pre-requisites for this integration solution.

Chapter 3. Integrating WebSphere Commerce business processes with SAP R/3 A look at the asynchronous and synchronous message flow between WebSphere Commerce and SAP through CrossWorlds.

Chapter 4. Installing and configuring Installation and configuration instructions for the components of this integration solution.

Chapter 5. Store customizations A list of the customizations made to the ToolTech store model provided with this integration solution.

Chapter 6. Verification procedure A list of instructions to verify the synchronous and asynchronous message flows.

Appendix A. Supported messages A list of the asynchronous and synchronous messages supported by this integration solution.

Appendix B. Supported generic business objects A list of the business objects supported by this integration solution.

Appendix C. Supported application specific business objects A list of the application specific business objects supported by this integration solution.

Appendix D. Collaboration templates A list of the collaboration templates used by this integration solution.

Appendix E. Maps with specific values for WebSphere Commerce and SAP

A list of the hard coded WebSphere Commerce and SAP fields used in the maps provided with this integration solution.

Appendix F. Binding maps to business objects A list of maps that must be explicitly bound to the business objects for the MQSeries and SAP connectors.

Appendix G. Sample BDC program A sample BDC program to import data into the SAP system.

Conventions used in this guide

This guide uses the following conventions:

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

`Monospaced type` indicates examples of text you enter exactly as shown, as well as directory paths.

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

Chapter 1. Introduction

This chapter gives an overview of the integration of WebSphere Commerce Version 5.4, Business Edition with SAP R/3 using the CrossWorlds system. It defines the terms used in this book and provides references to other related documents.

Terminology used in this book

The following 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.

BAPI

BAPIs (Business Application Programming Interfaces) are open business-oriented programming interfaces that can be used by external systems to access business processes and data in the SAP R/3 system.

Reference data

Reference data is a sample set of products and items included in the integration solution 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.
- SAP and SAP R/3.

Overview

As companies use the Internet to open their enterprises to customers, partners and suppliers for greater efficiency and productivity such as linking a purchasing department to outside vendors or adjusting inventory levels to match the latest customer buying patterns, they encounter layers of different enterprise systems that have been purchased over time, many of which are unable to interoperate.

This solution is designed to address one such integration process. It integrates SAP R/3 core business processes with the WebSphere Commerce sell-side solution, using the IBM CrossWorlds capabilities. This application provides the e-commerce functionality of WebSphere Commerce, along with the ERP functionality of the SAP R/3 enterprise system as the supplier and fulfillment center. It enables to build, deploy and integrate high-performance web sites with advanced e-business features using open standards.

WebSphere Commerce also extends the scope of enterprise applications such as SAP R/3 by providing a reliable, scaleable, and open-standards based commerce front-end. For example, an organization using SAP R/3 for enterprise

functions, when integrated with WebSphere Commerce, can easily add the Internet as a new sales channel for its products and services. In the scenario described below, the WebSphere Commerce server sell-side is an e-commerce web front-end to the SAP R/3 enterprise system.

The IBM CrossWorlds system's distributed hub-and-spoke architecture offers process integration solutions that automates and streamlines business processes, sophisticated business object management, inter-applications connectivity, and data integration. CrossWorlds with its dynamic configuration capability automatically propagates to the respective adapters or components. CrossWorlds application connectors provide swift integration to packaged, legacy and mainframe applications as well as e-business enabling technologies.

Business scenario - integration with the seller's backend system

WebSphere Commerce contains information about the products and services of the seller, and the profile and registration information of the customers. WebSphere Commerce processes the buyer's requests, such as placing an order, modifying profile information, checking availability and querying for the status of orders. The new customers created and material data in the SAP system will be uploaded onto WebSphere Commerce on a regular basis. The nature of transactions in this scenario allows WebSphere Commerce to be updated with material and customer data in the SAP system, and alternatively, SAP to be notified of the buyer's requests in WebSphere Commerce.

Extensibility

This solution consists of two parts, connecting WebSphere Commerce with CrossWorlds and connecting CrossWorlds with the SAP backend system.

This solution can be easily extended to integrate WebSphere Commerce with other backend systems. When connecting to other backend systems, there will be no changes in the solution to connect WebSphere Commerce with CrossWorlds. All that is required is to modify the CrossWorlds maps and connector configuration, specific to the system that is used to connect to your backend system.

Business models enabled

In this integration, WebSphere Commerce provides an e-commerce front-end Internet sales channel to the SAP R/3 enterprise system. Any customer registered with the SAP system can use WebSphere Commerce as the front end to browse and shop products that are loaded from SAP onto the WebSphere Commerce site.

Using this capability, buyers in any part of the world can shop online for products using the online stores and catalog display functionality provided by WebSphere Commerce. From the WebSphere Commerce site, buyers can place orders, check price and availability, query for the status of their orders and other relevant information that is present in the SAP system. This synchronization is possible by the initial upload of material data from SAP to WebSphere Commerce Business Edition. 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.

Benefits

- Easily and quickly add a new Internet sales channel to the enterprise backend system.
- Provide customers with access to web site functions such as browsing catalogs, placing orders, and making online payments.
- Synchronize product and customer information.
- Create orders in WebSphere Commerce and send the orders to SAP for order processing and fulfillment.
- Check the status of the order with SAP from the WebSphere Commerce site.
- Leverage the complex business processes supported by the SAP enterprise system.
- Create online catalogs in WebSphere Commerce for SAP materials.

Functionality

This integration solution provides the following functionality:

- **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:** 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 status messages supported are:
 - Order Confirmation
 - Order Delivery
 - Order Invoice
- **Customer Create:** When new customers are created in SAP the details 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 updated in 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.
- **Checking the inventory availability:** This message is used to check the inventory availability of a particular product in SAP from WebSphere Commerce using the `CheckInventoryAvailabilityBE` message.

References

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

- Messaging system information related to the WebSphere Commerce Business Edition 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-4.ibm.com/software/ts/WebSphere MQSeries/>
- IBM CrossWorlds documentation
<http://www-3.ibm.com/software/websphere/crossworlds/library/doc/410/welcome.html>

Chapter 2. Pre-requisites

This section covers the software and hardware pre-requisites for this integration solution. This integration solution assumes a Windows[®] operating environment.

Software pre-requisites

The software pre-requisites include:

Commerce Enhancement Pack

The Commerce Enhancement Pack uses WebSphere Commerce Version 5.4. WebSphere Commerce 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.

WebSphere MQSeries Version 5.2

This component is used as the transport middleware to communicate with various external systems, including CrossWorlds.

The MA88 Product Extension Pack for MQSeries Version 5.2

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.

SAP R/3 4.6

This is an ERP system that contains the master data such as catalog and customer information. 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 or BAPI requests from other applications to be processed by the SAP R/3 system.

IBM CrossWorlds version 4.1.0

The CrossWorlds system is a suite of software integration products that includes pre-built modules for common business integration requirements, and development and management tools. These products supply connectivity for leading e-business technologies and enterprise applications. This integration solution uses the complete CrossWorlds setup, which includes the CrossWorlds ICS (InterChange Server). For more information refer to the documentation that comes with the product.

Hardware pre-requisites

For information on hardware pre-requisites, refer to the documentation that comes with the appropriate software.

Chapter 3. Integrating WebSphere Commerce business processes with SAP R/3

This section describes the flow for the outbound messages from WebSphere Commerce and inbound messages to WebSphere Commerce that allow integration of WebSphere Commerce business processes with SAP R/3. It explains how messages are used to transport information through various components of this integration solution.

WebSphere Commerce outbound process

Transactions in WebSphere Commerce can trigger messages to be sent to other applications through the WebSphere Commerce messaging system. In this integration solution, the order create XML message can be generated from WebSphere Commerce to create an order in a backend system.

When an order is created in WebSphere Commerce, it can be configured such that the order related information is sent to another system for further processing like fulfillment. In this case, orders are sent through XML messages from WebSphere Commerce to CrossWorlds. This XML message can be processed by CrossWorlds and sent to other systems like SAP.

When you place an order in WebSphere Commerce, the `OrderCreate` message in XML format is generated and placed in the MQSeries output queue as shown in Figure 1. The CrossWorlds MQSeries connector agent constantly polls for new messages, which it passes from the MQSeries output queue to the MQSeries connector controller. The controller receives the WebSphere Commerce specific business objects and invokes the maps to generate the generic business objects (GBO). The GBOs are passed to the corresponding collaboration, which processes them. The processed business objects are sent to the SAP connector controller, which uses the maps to create SAP specific business objects and passes them to the SAP connector agent. The SAP connector agent sends the message to create the order to the SAP R/3 system. Similar flows apply to other ERP systems like SAP R/3.

Note: In the following figure GBO stands for generic business object.

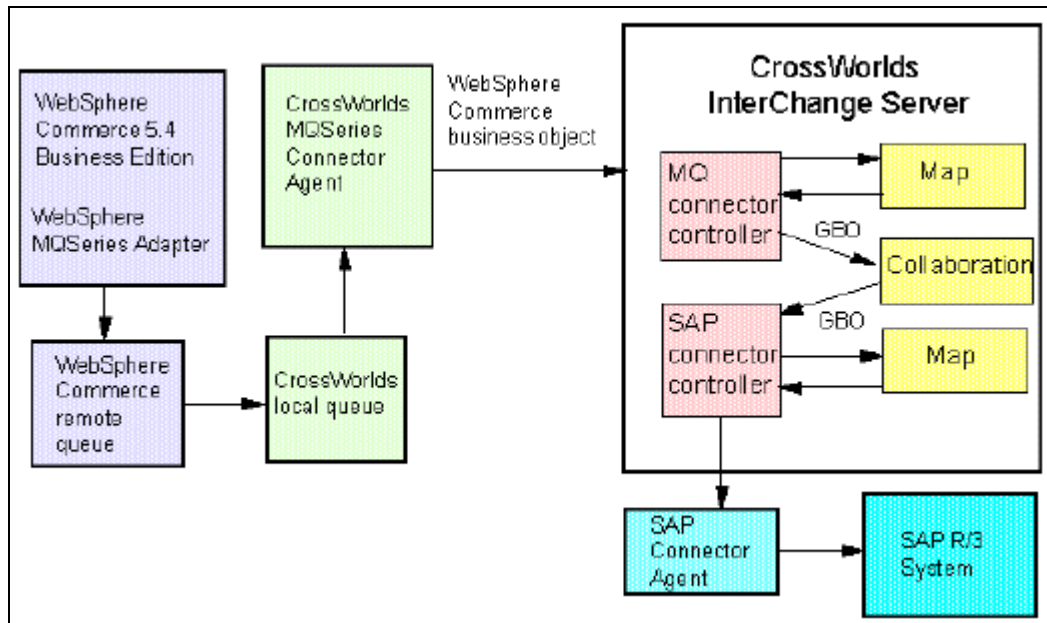


Figure 1: Asynchronous message flow from WebSphere Commerce

WebSphere Commerce inbound process

WebSphere Commerce supports processing message requests to invoke certain business logic. In this scenario, the existing WebSphere Commerce messages are used to invoke business logic triggered by SAP R/3 processing.

During business processing in SAP, you can configure SAP to generate messages that will update WebSphere Commerce. The following business interactions are triggered in SAP to initiate business logic in WebSphere Commerce:

- CustomerCreate
- CustomerUpdate
- OrderStatusUpdate
 - OrderConfirmationStatus
 - OrderDeliveryStatus
 - OrderInvoiceStatus
- ProductPriceUpdate
- ProductInventoryUpdate

The CrossWorlds SAP connector agent polls for the messages, receives and forwards them to the SAP connector controller as shown in Figure 2. The controller receives the messages and invokes the maps to convert the SAP business objects to generic business objects that will be processed by the collaboration. The controller then passes the application specific business objects to the MQSeries connector agent, which in turn sends the corresponding XML message to WebSphere Commerce.

Note: In the following figure GBO stands for generic business object.

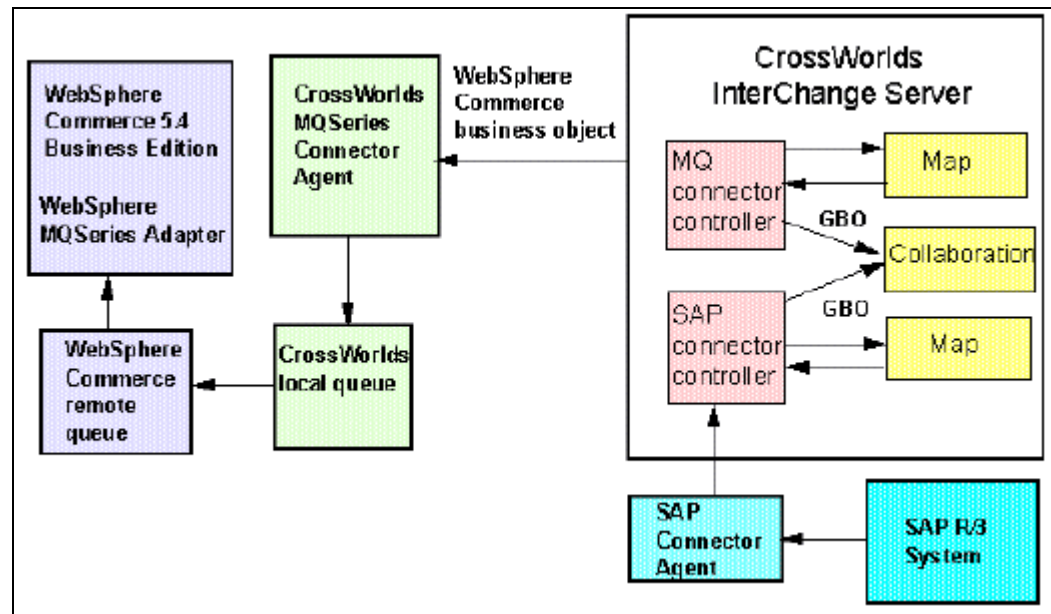


Figure 2: Asynchronous message flow into WebSphere Commerce

WebSphere Commerce Request/Reply process

The WebSphere Commerce messaging system allows business logic to initiate request/reply interactions with other systems when information is required in real time from external systems. In this example, the check product availability interface for external systems is used in combination with the CrossWorlds adapter to receive information about the availability of products from SAP R/3. This model can be used to initiate other interactions with external systems linked to the CrossWorlds system.

This integration solution supports the `CheckInventoryAvailabilityBE` synchronous message flow.

Sending a request

During normal order processing WebSphere Commerce checks for the availability of products. When you use the interface for checking availability in an external system to check product availability in SAP from WebSphere Commerce, an XML message is generated and passed to the CrossWorlds adapter in WebSphere Commerce. The CrossWorlds adapter sends the message to the server access interface in the CrossWorlds system. The server access interface invokes the data handler to generate the WebSphere Commerce business object. This business object is passed to the subscribed map that generates the generic business objects (GBO). The GBOs are passed to the collaboration object specified in the CrossWorlds adapter configuration properties. The collaboration object processes the business object. The processed business objects are then sent to the SAP connector controller, which invokes the related maps to create the SAP specific business objects and passes them to the SAP connector agent. The SAP connector agent makes the Material

Availability BAPI call using the application specific business objects to the SAP R/3 system.

Receiving a response

The SAP system responds with the status of the inventory availability that reaches the SAP connector agent. The SAP connector agent receives the message and generates the application specific business objects. The connector agent sends the business objects to the SAP connector controller. The controller invokes the maps to convert the SAP business objects to generic business objects. The controller then passes the generic business objects to the collaboration, which processes the business objects and sends the processed business objects to the CrossWorlds server access interface. The server access interface then generates the XML message, using data handlers, and responds to the transport adapter in WebSphere Commerce.

This model can be used for other request/reply interfaces initiated from WebSphere Commerce.

Note: In the following figure BO stands for business object.

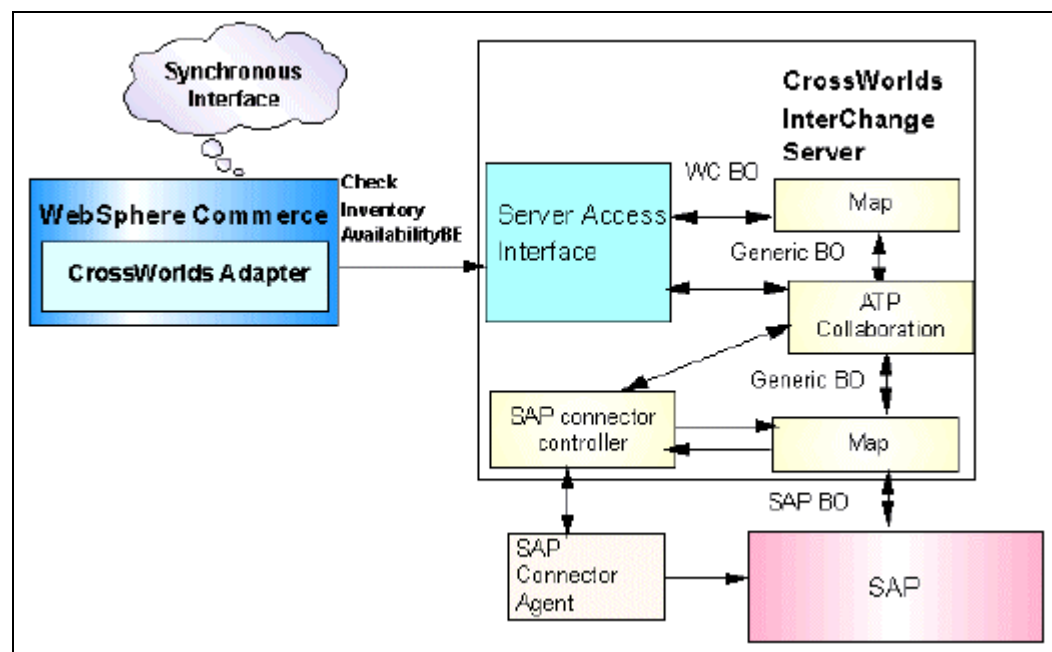


Figure 3: Request/Reply message flow from WebSphere Commerce to SAP

Chapter 4. Installing and configuring

To enable the WebSphere Commerce Business Edition – SAP integration using CrossWorlds, you must perform the following activities:

- Install and configure WebSphere Commerce Version 5.4, Business Edition.
- Install and configure WebSphere MQSeries.
- Configure the WebSphere Commerce messaging system.
- Install and configure IBM CrossWorlds.
- Create or configure the database used to store the CrossWorlds components definition.
- Configure the SAP R/3 system.
- Publish the ToolTech store.

Sample topology

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

Machine 1	Machine 2
Commerce Enhancement Pack using WebSphere Commerce Version 5.4 Business Edition	IBM CrossWorlds
MQSeries	MQSeries
	SAP connector agent
	MQSeries connector agent

Table 1: Sample topology

Installing and configuring WebSphere Commerce

Install WebSphere Commerce Version 5.4, Business Edition. Refer to the documentation that comes with the product for the installation steps and the post-install configuration. The WebSphere Commerce messaging system is equipped to handle the messages to interact with back-end systems.

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

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

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

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

```
'com.ibm.commerce.order.commands.OrderMessagingCmd'
```

2. Update the CMDREG table with the following entry, if IBM Payment Manager is not being used. For this integration solution, the ToolTech store is enabled with IBM Payment Manager.

```
update cmdreg set classname =  
'com.ibm.commerce.payment.commands.DoPaymentCmdImpl'  
where interfacename =  
'com.ibm.commerce.payment.commands.DoPaymentCmd' and  
storeent_id = <store id>
```

Where, <store id> is the store identifier for the ToolTech store

Note: The above commands will take effect only after restarting the server.

Installing the Commerce Enhancement Pack

To install the Commerce Enhancement Pack, download the Commerce Enhancement Pack driver from the following URL and follow the instructions in the `readme.txt` file:

<http://www.ibm.com/software/commerce/epacks>

Installing and configuring WebSphere MQSeries

1. Install WebSphere MQSeries version 5.2. Refer to the WebSphere MQSeries documentation on how to set up either the WebSphere MQSeries binding-mode, or the WebSphere MQSeries client-mode configuration. You can install WebSphere MQSeries in <drive>:\WebSphere directory or in any other location.

Note: You must have WebSphere MQSeries client-mode configuration if you have installed WebSphere Commerce on a different machine.

2. If you do not want to use the default queue manager, `QM_hostname`, then create a new queue manager and specify that as the default queue manager, using WebSphere MQSeries Explorer. If you are unable to use the WebSphere MQSeries Explorer, then use the following commands from the MS-DOS command window to create your queue manager:

```
crtmqm Queue_Manager_Name  
strmqm Queue_Manager_Name
```

Where, `Queue_Manager_Name` - the name you have chosen for the queue manager. `crtmqm` creates the queue manager and `strmqm` starts the queue manager.

3. Set the coded character set identifier of the queue manager to 1208 (UTF8) after you create your WebSphere MQSeries queue manager. 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.

4. 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 the queues are:

```
define qlocal (Your_Inbound_Queue)

define qRemote (Your_Outbound_Queue)

define qlocal (Your_Serial_Inbound_Queue)

define qlocal (Your_Parallel_Inbound_Queue)

define qlocal (Your_Error_Queue)

end
```

Where,

Your_Inbound_Queue is the WebSphere MQSeries queue created for the inbound queue.

Your_Outbound_Queue is the WebSphere MQSeries queue created for the outbound queue.

Your_Serial_Inbound_Queue is the WebSphere MQSeries queue created for the serial inbound queue.

Your_Parallel_Inbound_Queue is the WebSphere MQSeries queue created for the parallel inbound queue.

Your_Error_Queue is the WebSphere MQSeries queue created for the error queue.

Note: The queue names are case sensitive. To create the outbound queue use, `define qRemote Queue_Name`, as this queue must be a remote definition queue to the queue created on the CrossWorlds side, for the MQSeries Connector Agent.

5. Verify the Queue Manager and confirm that the queues created are working, by using the **WebSphere MQSeries First Steps - Post Card** utility.
6. Install the MA88 product extension in the `<drive>:\MQ_install_path\java` directory.

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

Configuring the WebSphere Commerce messaging system

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

1. Open an MS-DOS command window on your WebSphere Commerce machine and do the following:
 - a. 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 - the path in which you installed WebSphere MQSeries.

WAS_install_path - the path in which you installed the WebSphere Application Server.

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

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

Where, *MQ_install_path* is the path in which you installed WebSphere MQSeries.

- c. 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 WebSphere Application Server.

2. Change to the *MQ_INSTALL_PATH/java/bin* directory and run the JMSAdmin program using the following command:

```
JMSAdmin -cfg JMSAdmin.config -t -v
```

3. Ensure that the WebSphere Application Server is running and the correct classpath and environment variable defined previously in this section are added.

- a. 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
```

- b. 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 Application Server. You will see an **InitCtx>** prompt that you can use to run the JMS administration commands.

- c. Register the QueueConnectionFactory and set the coded character set identifier by typing the following commands:

```
i) define qcf (JMS_QueueConnectionFactory)
   qmanager (Your_QueueManager_Name)
```

```
ii) alter qcf (JMS_QueueConnectionFactory)
     ccsid(1208)
```

Where, *JMS_QueueConnectionFactory* is the name of the MQQueueConnectionFactory JMS object.

On executing the above set of commands, an entry for this queue connection factory must be found in the WebSphere Application Server database under the BINDINGBEANTBL table. These objects are registered in the WebSphere Application Server database.

- d. Define the following JMSQueues. The following queue names are used for this integration solution and you can change them to suit your requirements.
 - define
`q(JMS_Outbound_Queue) qmanager(Your_Queue_Manager_Name) queue`
 - define
`q(JMS_Inbound_Queue) qmanager(Queue_Manager_Name) queue(Your_Inbound_QueueName)`
 - define
`q(JMS_Parallel_Inbound_Queue) qmanager(Queue_Manager_Name) queue(Your_Parallel_Inbound_Queue_Name)`
 - define
`q(JMS_Serial_Inbound_Queue) qmanager(Your_Queue_Manager_Name) queue(Your_Serial_Inbound_Queue_Name)`
 - define
`q(JMS_Error_Queue) qmanager(Your_Queue_Manager_Name) queue(Your_Error_Queue_Name)`

Where,

Queue_Manager_Name is the name you have chosen for the queue manager when configuring WebSphere MQSeries. `ctrmqm` creates the queue manager and `strmqm` starts the queue manager.

JMS_Serial_Inbound_Queue is the name of the JMSQueue object.

JMS_Outbound_Queue is the WebSphere MQSeries queue created for the outbound queue.

Your_Serial_Inbound_Queue is the WebSphere MQSeries queue created for the serial inbound queue.

JMS_Parallel_Inbound_Queue is the WebSphere MQSeries queue created for the parallel inbound queue.

JMS_Error_Queue is the WebSphere MQSeries queue created for the error queue.

Your_Queue_Manager_Name is the name of the queue manager created when configuring WebSphere MQSeries.

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

- e. After creating the queues set the following property to the outbound and error queues using the JMSAdmin console. This procedure

indicates that JMS is dealing with a native WebSphere MQSeries application.

```
alter q(JMSOutboundQueue) targclient(MQ)
alter q(JMSErrorQueue) targclient(MQ)
```

- f. Type **end** to exit the JMSAdmin tool. This finishes configuring the Java Messaging Service with WebSphere Commerce.
4. **Start** the WebSphere Commerce Administration Console. Login as a Site Administrator, go to the **Configuration** section and choose the **Transport** option. Select WebSphere MQSeries as your transport and change the status to **activate**.
5. Log out from the Administration Console.
6. When you complete publishing the store as described in “Publishing a store”, log into the Administration Console, this time as a Store Administrator and select the store used by this integration solution. From the **Configuration** section add **MQ Transport** to the store. An entry for this is made in the STORETRANS table.
7. To enable the messaging system transport adapter, launch the WebSphere Commerce Configuration Manager and do the following:
 - a. Select **Host name -> Instance**, and then open the Components folder.
 - b. Select **TransportAdapter**.
 - c. Select the **Enable Component** checkbox and click **Apply**.
 - d. **Exit** the Configuration Manager.
8. 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.

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

Installing and configuring CrossWorlds

To install and configure CrossWorlds, refer to the *CrossWorlds System Installation Guide for Windows* provided with the product. In addition to installing the CrossWorlds InterChange Server, installing MQSeries and SAP connector agents are also mandatory for this solution. This integration solution supports CrossWorlds Version 4.1.0

Loading the business object definitions, maps, and collaboration objects

This integration solution comes with a set of components such as business object definitions, maps, and collaboration objects for each asynchronous and synchronous message. For the list of application specific business objects, see Appendix C. Supported application specific business objects. You can obtain these files from the CrossWorlds exchange site

<http://exchange.crossworlds.com>. Select, **Products > Business Objects & Maps > Industry Specific Business Objects > WebSphere Commerce**.

Note: Some of the WebSphere Commerce and SAP system specific fields in the maps are hard coded. See, Appendix E. Maps with specific values for WebSphere Commerce and SAP for details. You can change these fields according to your WebSphere Commerce and SAP configuration.

The following list of files contains the components that you must load into the CrossWorlds repository:

File name	Remarks
MO_MQSeriesConfig.out	In AppSpecificInfo, for the parameter "OutputQueue", the value must have the appropriate Queue manager and queue name.
Sap_idoccontrol.out	This is the business object definition for SAP
OrderCreate.txt	Contains the definition for the collaboration object, business objects, and maps
ProductPriceUpdate.txt	Contains the definition for the collaboration object, business objects, and maps
ConfirmationStatus.txt	Contains the definition for the collaboration object, business objects, and maps
DeliveryStatus.txt	Contains the definition for the collaboration object business objects, and maps
BillingStatus.txt	Contains the definition for the collaboration object, business objects, and maps
ProductQuantityUpdate.txt	Contains the definition for the collaboration object, business objects, and maps
customerCreate_Update.txt	Contains the definition for the collaboration object business objects, and maps
CheckInventory.txt	Contains the definition for the collaboration object business objects, and maps

Table 2: Components to be loaded into the CrossWorlds repository

1. Before loading any of the flat files listed in Table 2, ensure that you copy the files from `crossworlds\message_dir\NativeMapfiles` to `CrossWorlds_home_dir\DLMS\classes\NativeMaps`.
2. Load the `Sap_idoccontrol.out` into the CrossWorlds repository before you load the other flat files. The `Sap_idoccontrol` object is used as a child object in all other SAP specific business objects.
3. The application specific info text and the message type for the supported verbs for sap specific ASBOSs are updated with the appropriate values.

To load the definition files into the repository use the `repos_copy` utility. For example, to load `ProductQuantityUpdate.txt`, use the following command:

```
(repos_copy <-i [inputfile]> -ai [- sservername] [-
username] [- ppassword])
```

Where, `<-i [inputfile]>` is the business object, for example, `ProductQuantityUpdate.txt` in the `crossworlds\ProductQuantityUpdate\repos_copy\` which is the directory where the definition files are stored.

`[- sservername]` is the CrossWorlds Interchange Server name, which is CrossWorlds in this case.

`[- username]` is the user name for the CrossWorlds Interchange Server, which is admin in this case.

`[- ppassword]` is the password for the CrossWorlds Interchange Server, which is null in this case.

Note: Repeat executing the `repos_copy` utility for each of the definition files.

MO_MQSeriesConfig meta object

The MQSeries static meta-object consists of a list of conversion properties defined for all of the supported WebSphere Commerce specific business objects. For this integration solution, we have provided this meta object in the `crossworlds\MQConfig\MO_MQSeriesConfig.out` file. You must load this meta object into the CrossWorlds repository, and specify the value for the property `ConfigurationMetaObject` as 'MO_MQSeriesConfig' in the definition for `MQSeriesConnector` as mentioned in the following section. Open this meta object in the Business Object designer tool and update the application specific information with the queue manager name that is used in this solution.

Configuring the MQSeries connector agent

The MQSeries connector agent allows CrossWorlds collaborations to asynchronously exchange business objects with applications that issue or receive MQSeries messages when the data changes.

Configuring MQSeries connector involves setting the values for the generic and agent specific configuration properties. You must provide the following details:

1. Specify the list of generic and application specific business objects supported by the MQSeries connector. For more information see, Appendix B. Supported generic business objects, and Appendix C. Supported application specific business objects respectively.

2. Specify the MetaConfiguration Object used by the MQSeries connector agent to receive messages for the supported business objects. The value is 'MO_MQseriesConfig'.

Note: You must load the business objects and other components as specified in the section “Loading the business object definitions, maps, and collaboration objects” before configuring for the connectors.

3. Specify the associated maps with Explicit Binding for the supported business objects. See, Appendix F. Binding maps to business objects to identify the business objects that require maps to be specified.
4. Setting the BO (Business Object) prefix

The application specific business objects for the WebSphere Commerce application are generated with the 'WCS' prefix. Ensure that 'WCS' is set for the BOPrefix property of the MO_DataHandler_Default object. Use the Business Object Designer tool to navigate to the following object hierarchy:

MO_DataHandler_Default - The parent business object

Text.xml - The child object of MO_DataHandler_Default

BOPrefix - Property name (Set WCS as the value for this property)

For further information on how to configure the MQSeries connector agent, refer to Chapter 2 in the *Guide to the CrossWorlds Connector for MQSeries, Connector Version 1.4.x*.

Configuring the SAP connector agent

The CrossWorlds connector for SAP enables the CrossWorlds InterChange Server to exchange business objects with SAP applications. The CrossWorlds SAP business objects are defined in SAP as IDocs. IDocs are part of Application Link Enabling (ALE) module. The IDoc definitions are stored in the Business Object Repository in SAP. CrossWorlds provides an IDoc Handler that supports business objects developed using IDocs.

To install the SAP connector agent, refer to the *CrossWorlds System Installation Guide for Windows*. All SAP connector components can be found in the \connectors\SAP directory.

SAP modules

The three modules used in this integration solution include:

- ALE module
- BAPI module
- RFCServer module

To set values to the configuration properties of the SAP connector refer to the *Guide to the CrossWorlds Connector for SAP R/3 Version 4.x*, Connector Version 4.5.x in the CrossWorlds documentation.

Configuring the SAP connector involves setting the values for the generic and agent specific configuration properties. You must provide the following details:

1. Specify the list of generic and application specific business objects supported by the SAP connector. For more information see, Appendix B. Supported generic business objects, and Appendix C. Supported application specific business objects respectively.

2. For the supported business object specify the associated map with the 'Explicit Binding' option selected. See, Appendix F. Binding maps to business objects to identify the business objects that require maps for conversion.
3. To set up the Transaction ID (TID) management, refer to the *Guide to the CrossWorlds Connector for SAP R/3 Version 4.x, Connector Version 4.5.x* in the CrossWorlds documentation. The corresponding objects are provided as part of the SAP connector installation.
4. To support the BAPI module, copy the following files from the location `crossworlds\CheckAvailability\genfiles` to `\<crossworlds_homedir>\connectors\SAP\Bapi\client`
 - o `Bapi_material_availability.java`
 - o `Bapi_material_availability.class`

Note:

- To support the ALE, BAPI, and RFCServer modules in the SAP connector agent, set the modules property value as 'ALE, BAPI, RFCServer' in the SAP connector definition properties.
- Before connecting to the SAP system, you must configure the RFC Destination with the `program ID` that you have set in the connector properties. To do this, refer to the *Guide to the CrossWorlds Connector for SAP R/3 Version 4.x, Connector Version 4.5.x* in the CrossWorlds documentation

Port connector

The collaboration ports that are not used in a collaboration object to support a message are bound to the port connector. For details, see Appendix D. Collaboration templates.

Creating and configuring the collaboration objects

The collaboration objects associated with the synchronous and asynchronous messages are provided in the definition flat files listed in Table 2.

Before you execute the `CheckInventoryAvailabilityBE` message, you must modify the ATP collaboration template. This modification enables the ATP collaboration to populate the `ATPLine` business object values. To do this,

1. Open the ATP collaboration template in the process designer.
2. Open **Scenarios > Main > Sync Logic(9) > Line Iteration(59) > Send Verb(299)**.
3. In the scenarios diagram for send Verb(299), right click on the action box with the UID **386 Create** and select **Properties**.
4. In the Action properties window, comment the following code segment:


```
ToLineBusObj.setKeys(ATPLine);
```

 Add the following code segment:


```
ToLineBusObj.copy(ATPLine);
```
5. Click **Apply**.
6. Save and compile the ATP template.

For the list of collaboration templates and the corresponding binding information, see Appendix D. Collaboration templates.

Loading and starting the maps

Use the maps provided with this solution to transform the application-specific business objects to generic business objects and vice versa. See, Appendix F. Binding maps to business objects for details.

Note: The control segment properties for the SAP IDocs are set in the SAP specific inbound maps, for example, `Sap_OrderCreate_Map`. You must change these values according to the SAP configuration for partners. Refer to the *Guide to the CrossWorlds Connector for SAP R/3 Version 4.x, Connector Version 4.5.x*. in the CrossWorlds documentation. To edit the values, open the SAP specific maps in the Map Designer tools and change them.

Configuring the database to store the CrossWorlds components definition

The CrossWorlds InterChange Server requires at least one database to hold its repository. The database server that manages this database must allow connections using JDBC (Java Database Connectivity) access. For more information refer to the *CrossWorlds System Installation Guide for Windows* provided with the CrossWorlds documentation.

Configuring the CrossWorlds queue manager

This integration solution requires you to configure an MQSeries Queue Manager and an MQSeries Listener at the CrossWorlds side. Refer to the CrossWorlds documentation for more information.

Enabling asynchronous communication

This section details the steps required for enabling communication between the WebSphere Commerce MQSeries and CrossWorlds MQSeries.

Configuring the WebSphere Commerce MQSeries

Channels are used to communicate between two MQSeries systems. Do the following to configure the WebSphere Commerce MQSeries:

Note: The channel names used here are examples and you can change them.

1. Create two channels in the WebSphere Commerce system using MQSeries Explorer. One sender channel named 'WCBE' and one receiver channel named 'CW'.
2. Create a local queue, for example, using the name 'ToCrossworldsSystem'.
3. Set the ToCrossworldsSystem queue as the transmission queue.
4. Set the following properties for the *Your_Outbound_Queue*. See, "Configuring the WebSphere Commerce messaging system".
 - a. Remote queue name
`ICA/CrossWorlds_server_name/MQSERIESCONNECTOR`. For example, `ICA/CrossWorlds/MQSERIESCONNECTOR`.
 - b. Remote queue manager name
`CrossWorlds_server_name.queue.manager`. For example, `CrossWorlds.queue.manager`.

- c. Set the transmission queue name property to 'ToCrossworldsSystem' as created in step 2,
5. To configure the sender channel do the following:
 - a. Specify the connection name with the IP address and the port, for example, 9.182.12.235(1414). Where, 9.182.12.235 is the IP address of the machine where CrossWorlds is running and 1414 is the default listener port.
 - b. Specify the transmission queue name as 'ToCrossWorldsSystem'.

You have now configured the WebSphere Commerce MQSeries

Configuring the CrossWorlds MQSeries

Do the following to configure the CrossWorlds MQSeries:

1. Create two channels in the CrossWorlds system using MQSeries Explorer. One sender channel named 'CW' and one receiver channel named 'WCBE'.

Note: The name of the sender channel in CrossWorlds must be identical to the name of the receiver channel in WebSphere Commerce. The name of the receiver channel in CrossWorlds must be identical to the name of the sender channel in WebSphere Commerce.

2. Create a new local queue, for example 'ToWCSSystem'. Set the ToWCSSystem queue as the transmission queue.
3. Create a remote definition queue in the CrossWorlds system. This remote definition queue must be used in the CrossWorlds MQSeries connector agent as the output queue. Set the following properties:
 - a. Remote queue name
For example, JMSSerialInboundQueue. This queue must be created as specified in "Installing and configuring WebSphere MQSeries".
 - b. Remote queue manager name `<wcssystem_Q_manager_name>`.
For example, QM_wcsfvt3.
 - c. Set the transmission queue name property to 'ToWCSSystem'.
4. To configure the sender channel do the following:
 - a. Specify the connection name with the IP address and the port, for example, 9.182.12.18(1414). Where, 9.182.12.18 is the IP address of the machine where WebSphere Commerce is running and 1414 is the default listener port.
 - b. Specify the transmission queue name as 'TOWCSSystem'.

Note: After you finish configuring the WebSphere Commerce and CrossWorlds MQSeries start the receiver channel and then the sender channel.

Configuring the CrossWorlds adapter in WebSphere Commerce

This adapter provides a new transport mechanism. The CrossWorlds adapter provides synchronous connectivity to the external systems through the CrossWorlds collaborations utilizing the capabilities of the existing WebSphere Commerce messaging subsystem. To configure the adapter in WebSphere Commerce, refer to the Commerce Enhancement Pack.

Configuring SAP

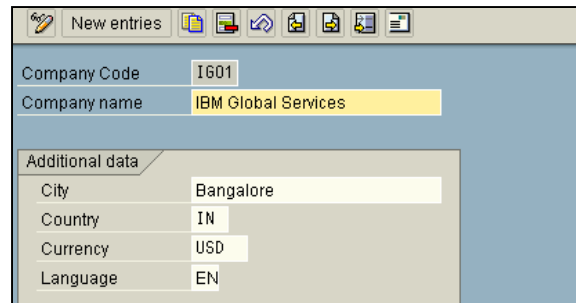
This section details how to configure the SAP R/3 system for this reference application. It includes the transactions to be used, customizations to be made, and a description of creating master data. Assumptions that have been made are listed where appropriate.

- 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



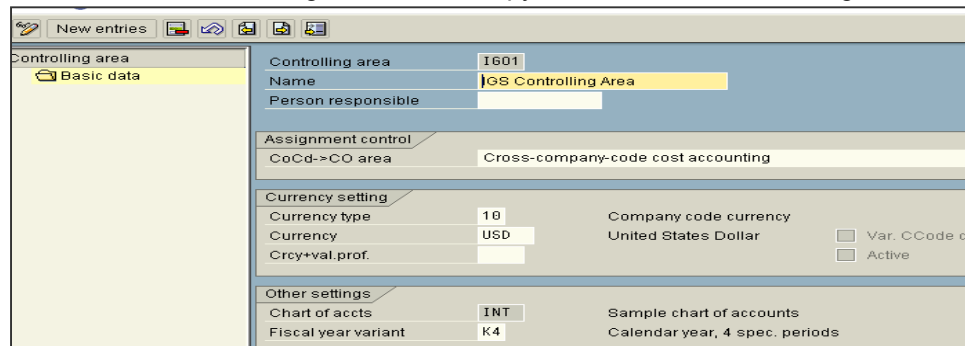
New entries	
Company Code	I601
Company name	IBM Global Services
Additional data	
City	Bangalore
Country	IN
Currency	USD
Language	EN

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.



New entries	
Controlling area	
Basic data	
Controlling area	I601
Name	IGS Controlling Area
Person responsible	
Assignment control	
CoCd->CO area	Cross-company-code cost accounting
Currency setting	
Currency type	18
Currency	USD
Crcy+val.prof.	
Company code currency	
United States Dollar	
<input type="checkbox"/> Var. CCode of	
<input type="checkbox"/> Active	
Other settings	
Chart of accts	INT
Fiscal year variant	K4
Sample chart of accounts	
Calendar year, 4 spec. periods	

Figure 5: Creating the controlling area

Creating a Plant – transaction code EC02

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

The screenshot shows the SAP 'Change View Plants: Details' screen. The title bar includes 'Table view', 'Edit', 'Goto', 'Selection criteria', 'Utilities', 'System', and 'Help'. The SAP logo is in the top right corner. Below the title bar, there is a toolbar with icons for 'New entries', 'Save', 'Cancel', 'Back', 'Forward', 'Print', 'Help', and 'Search'. The main area is divided into two sections. The top section contains the plant details: 'Plant' (WSAL), 'Name 1' (Websales plant), and 'Name 2' (empty). The bottom section is titled 'Detailed information' and contains a table of fields. The fields are: 'Language key' (EN, English), 'House number/street' (19, Fremont Road, WCS lane,), 'P.O. Box' (empty), 'Postal code' (56008), 'City' (empty), 'Country key' (US, USA), 'Region' (NY, New York), 'County code' (empty), 'City code' (empty), 'Jurisdiction code' (empty), and 'Factory calendar' (01, Germany (standard)). A note at the bottom states: 'Note: Address data can only be maintained in the address screen. The changes can only be seen in the overview and detail view after they have been saved.' The status bar at the bottom shows 'GAD (1) (100)', 'sapibm', and 'INS'.

Detailed information	
Language key	EN English
House number/street	19, Fremont Road, WCS lane,
P.O. Box	
Postal code	56008
City	
Country key	US USA
Region	NY New York
County code	
City code	
Jurisdiction code	
Factory calendar	01 Germany (standard)

Note: Address data can only be maintained in the address screen.
The changes can only be seen in the overview and detail view
after they have been saved.

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.

The screenshot shows the SAP transaction EC04 'Change View Sales organizations: Details'. The 'Sales organization' field is set to 'WSO1' and 'Web Sales'. The 'Statistics currency' is 'INR'. The 'Address text name' is 'ADRS_SENDER', 'Letter header text' is 'ADRS_HEADER', 'Footer lines text' is 'ADRS_FOOTER', and 'Greeting text name' is 'ADRS_SIGNATURE'. The 'RefSorg.SalesDocType' is empty, 'Cust.inter-co.bill.' is empty, and 'Sales org.calendar' is '01'. The 'Rebate proc.active' checkbox is unchecked. The 'ALE : Data for purchase order' section shows 'Purch. organization' as empty, 'Plant' as empty, 'Purchasing group' as empty, 'Storage location' as empty, 'Vendor' as empty, 'Movement type' as empty, and 'Order type' as empty.

Figure 7: Creating a sales organization

Creating a Sales division – transaction code EC06

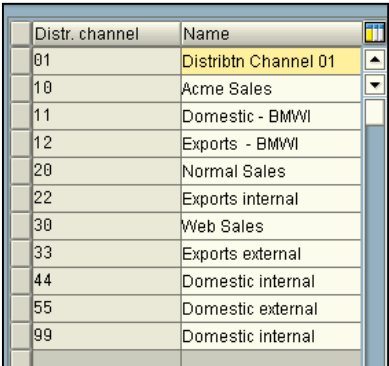
Create a sales division 30 by copying it from the 01 sales division.

Division	Name
01	Product Division 01
10	Acme Sales
11	Lower Segment BMW
12	Higher Segment BMW
20	Normal sales
30	Web Sales
60	Maintenance Services
70	Fresh Implementation
80	T&M Projects
99	Maintenance Services

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.



The screenshot shows a table with two columns: 'Distr. channel' and 'Name'. The table contains the following data:

Distr. channel	Name
01	Distribtn Channel 01
10	Acme Sales
11	Domestic - BMWI
12	Exports - BMWI
20	Normal Sales
22	Exports internal
30	Web Sales
33	Exports external
44	Domestic internal
55	Domestic external
99	Domestic internal

Figure 9: Creating a distribution channel

Creating shipping Point – transaction code EC07

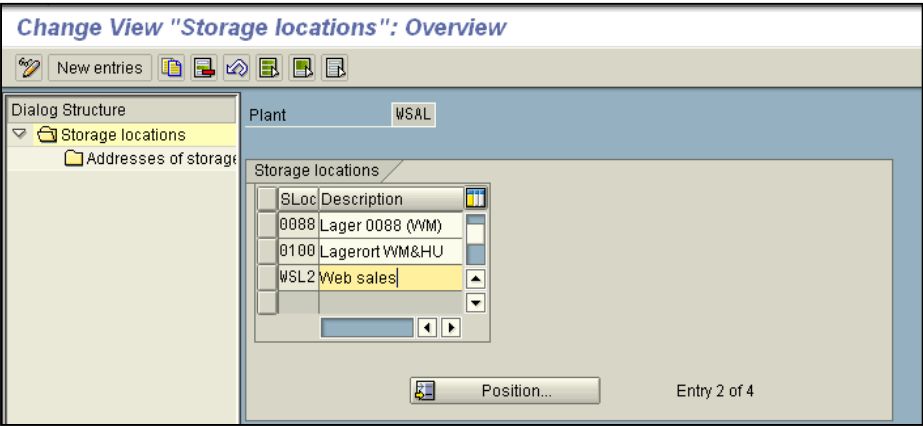


Figure 10: Maintaining storage location

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.

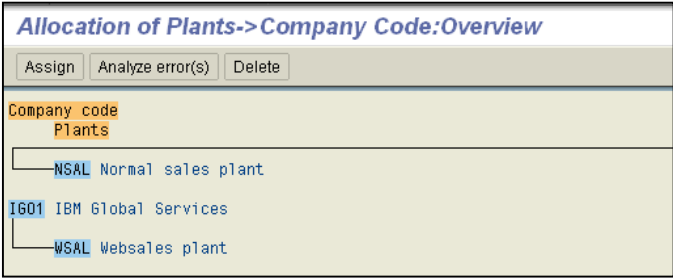


Figure 11: Allocation of plants

2. Allocate the company code to the sales organization.

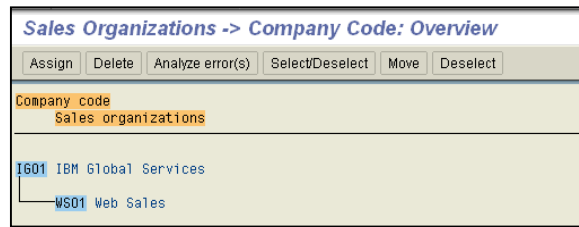


Figure 12: Allocation of the company code

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.

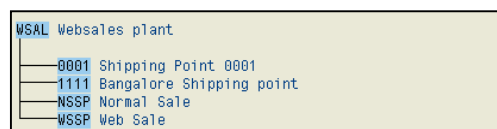


Figure 17: Allocating a shipping point

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

Change View "Shipping Point Determination": Overview

New entries

	SC	LGrp	Plnt	PrShP	MShPt	MShPt	MShPt	MShPt	MShPt	MShPt	MShPt	MShPt	MShPt	MShPt	MShPt
	01	0001	0001	0001											
	01	0001	1000	1000											
	01	0001	1100	NSSP											
	01	0001	1111	1111											
	01	0001	2000	2000											
	01	0001	AKG	0001											
	01	0001	AKG1	0001											
	01	0001	MATL	01											
	01	0001	NSAL	NSSP											
	01	0001	WSAL	WSPP											

Figure 18: Allocating shipping point

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

Maintain the following settings for the sales organization WSO1.

SOrg	DChl	Name	DCh-Cond	Name	DCh-CustMt	Name
2222	33	Exports external	33	Exports external	33	Exports
2222	44	Domestic internal	44	Domestic internal	44	Domes
2222	55	Domestic external	55	Domestic external	55	Domes
2222	99	Domestic internal	99	Domestic internal	99	Domes
AKG	30	Web Sales	30	Web Sales	30	Web S
NS01	10	Acme Sales	20	Normal Sales	20	Norma
NS01	20	Normal Sales	20	Normal Sales	20	Norma
WS01	30	Web Sales	30	Web Sales	30	Web S

Figure 19: Distribution Channel settings

SOrg	Dv	Name	DivCon	Name	DivCus	Name
2222	99	Maintenance Services	99	Maintenance Services	99	Maintenance Service
AKG	30	Web Sales	30	Web Sales	30	Web Sales
NS01	10	Acme Sales	20	Normal sales	20	Normal sales
NS01	20	Normal sales	20	Normal sales	20	Normal sales
WS01	30	Web Sales	30	Web Sales	30	Web Sales

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.

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

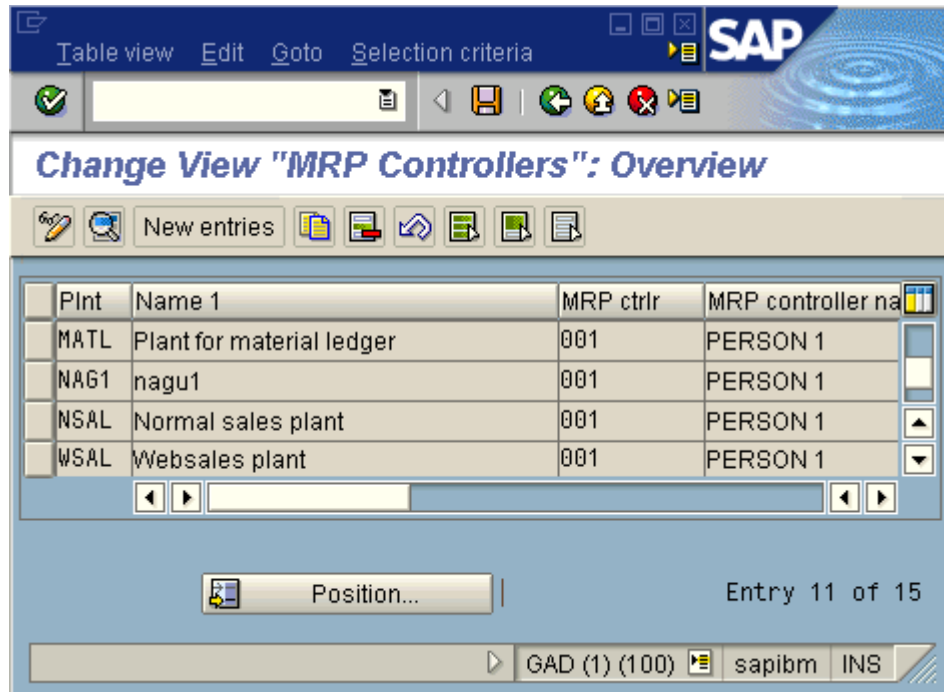


Figure 21: Defining the MRP Controller

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.

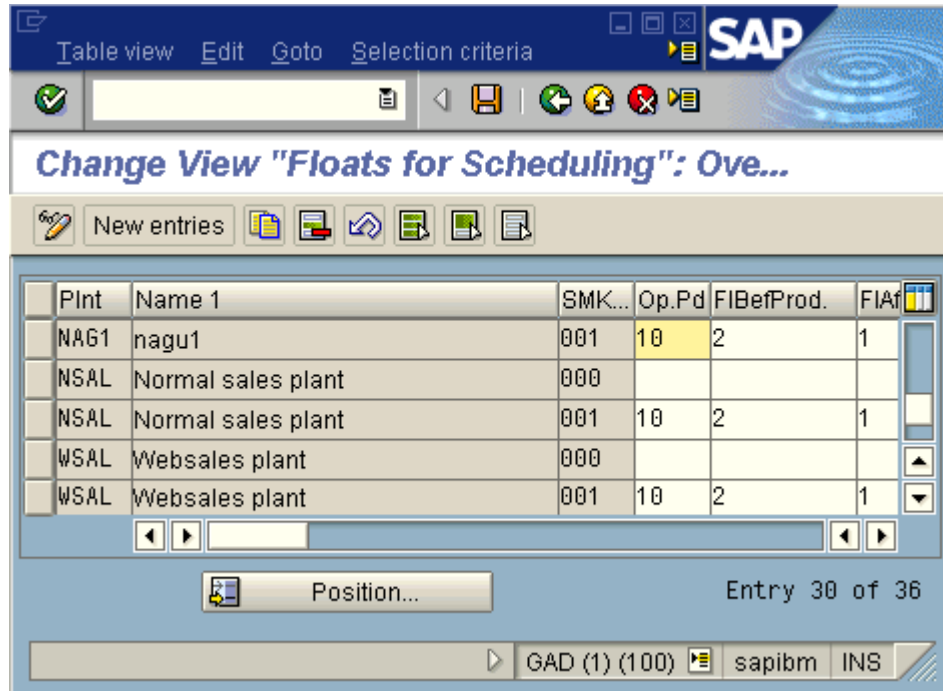


Figure 22: Defining the Floats (Schedule Margin Key)

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 enable 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.

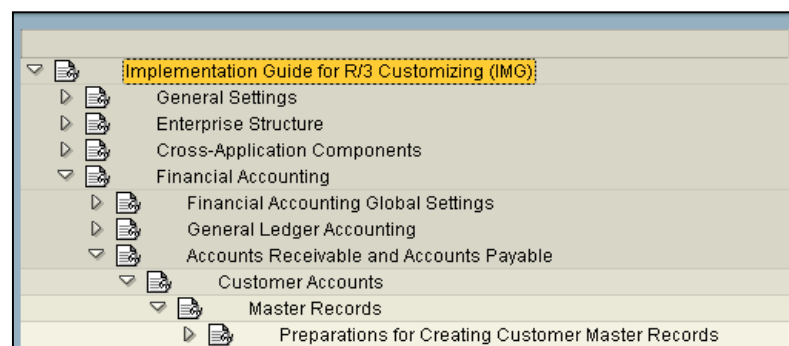


Figure 23: Customizing the master data

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

- 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.

Field	Suppress	Req. entry	Opt. entry	Display
Name 1/last name	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Form of address	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Search term A	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Name 2/first name	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Name 3, name 4	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Postal code, city	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Street	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Location	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Region	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 24: Maintaining the field status address group

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

Field	Suppress	Req. entry	Opt. entry	Display
Account assignment group	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Terms of payment	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Incoterms	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Subsequent invoice processing	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Billing period	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

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.

- To do this, go to transaction code SPRO, and click **SAP Reference IMG**.
- From the tree structure, select **Implementation Guide for R/3 Customizing (IMG)**.
- 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.

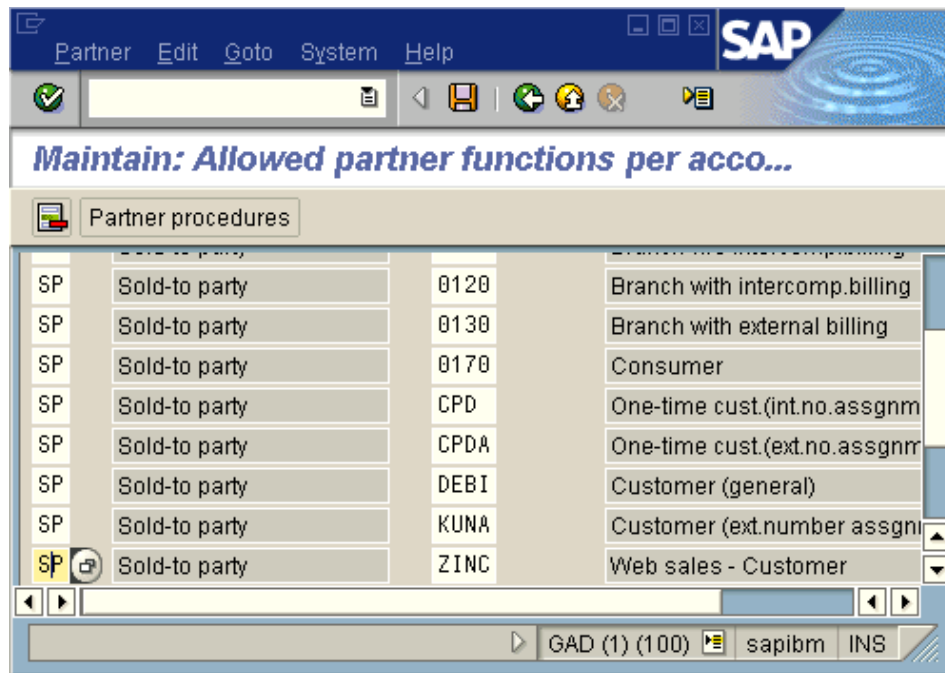


Figure 26: Assigning partner functions

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".

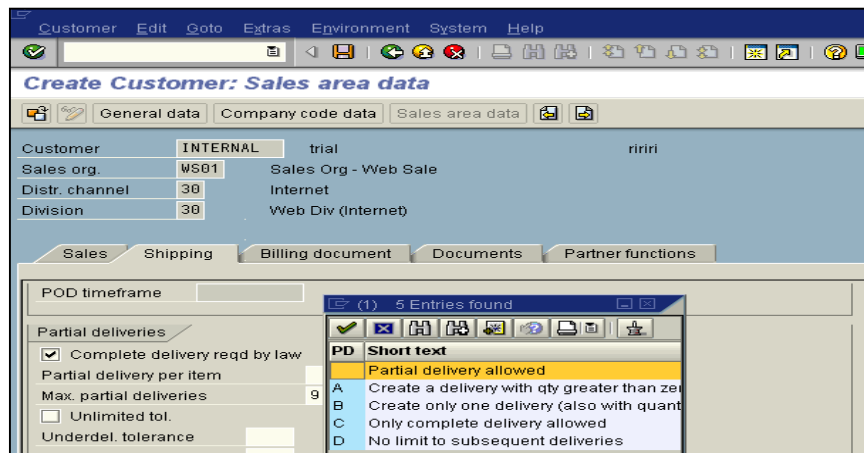


Figure 27: Selecting partial delivery in SAP

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.

Display View "Plants": Details

Plant: WSAL

Name 1: Websales plant

Name 2:

Detailed information

Language key	EN	English
House number/street	19, Fremont Road, WCS lane,	
P.O. Box		
Postal code	56008	
City		
Country key	US	USA
Region	NY	New York

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 show in the following figures.

Change View "Customer Taxes": Overview

New entries

Tax categ.	Name	Tax class.	Description
UTX2	County Sales Tax	0	
UTX2	County Sales Tax	1	
UTX3	City Sales Tax	1	
UTXJ	Tax Jurisdict.Code	0	Exempt
UTXJ	Tax Jurisdict.Code	1	Taxable

Figure 29: Maintaining tax relevancy for customers

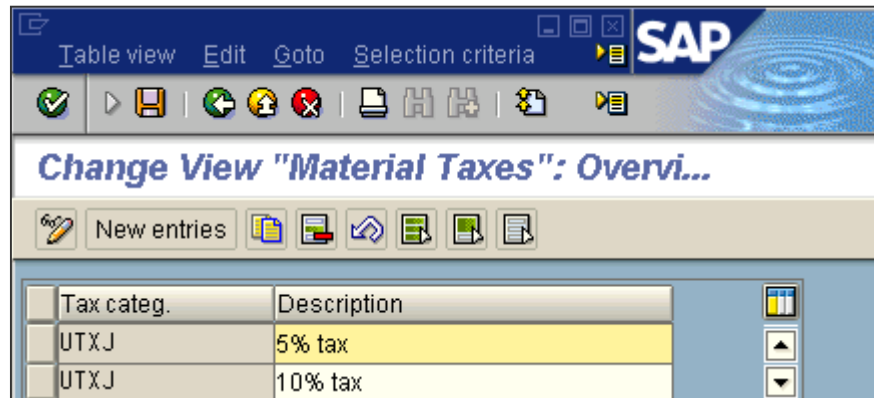


Figure 30: Maintaining tax relevance for materials

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%

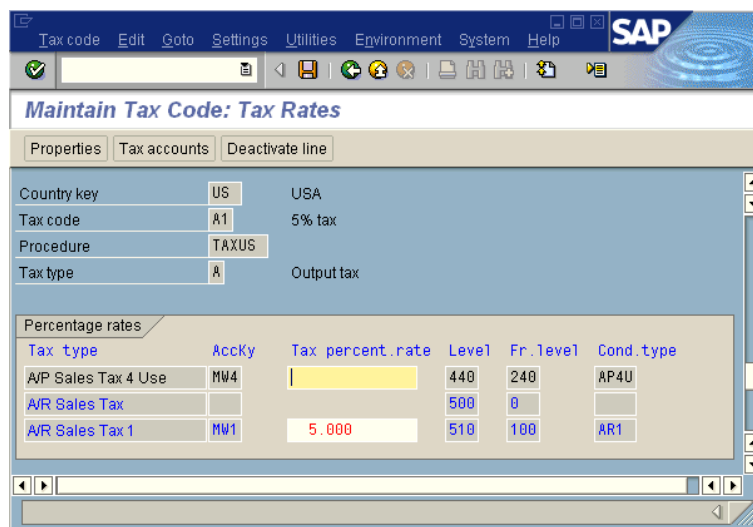


Figure 31: Maintaining tax codes for a country

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.

3010	Matl grp 3010			
3011	Matl grp 3011			

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.

CGrp	Name	
01	Industry	▲
02	Retail	▼
03	Internet Customer	

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.

Display Condition Table (Pricing Sales/Distribution): Field Overview

Technical view | Other description | Field attributes...

Table: 502 Sales org./Distr. ch/Division/Cust.group/Material

☒ With validity period
☐ with release status

Selected fields	FieldCatlg
Long Key Word	Long Key Word
Sales organization	Accounting indicator
Distribution channel	Act. gross inc. tax
Division	Agreement
Customer group	Base unit of measure
Material	Batch
	Bill-to party
	CAP prod. group
	Catalog
	City code
	City of deliv.plant
	Commission group
	Conditn pricing date

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.

Technical view | Other description | Field attributes...

Table: 700 Freight Calculation for WCS/BAPL

☒ With validity period
☐ with release status

Selected fields	FieldCatlg
Long Key Word	Long Key Word
Region of dlv.plant	Accounting indicator
Region	Act. gross inc. tax
Incoterms	Agreement
Incoterms (part 2)	Base unit of measure

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.

Table: 699 Country/PIntRegion/TaxCl.Mat/TaxCl1Cust/Region

☒ With validity period
☐ with release status

Selected fields	FieldCatlg
Long Key Word	Long Key Word
Country	Accounting indicator
Region of dlv.plant	Act. gross inc. tax
Tax class. material	Agreement
TaxClass1-Cust	Base unit of measure
Region	Batch

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.

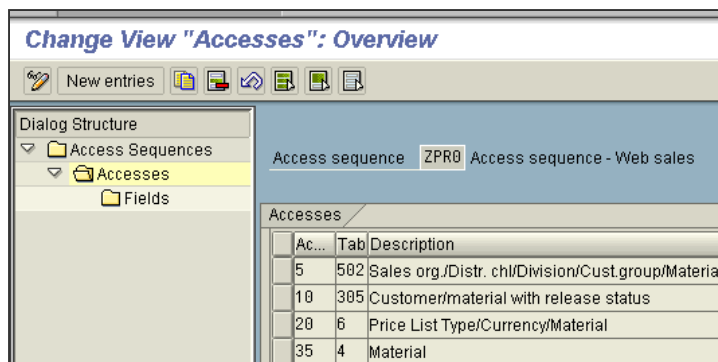


Figure 37: The access sequence that contains tables

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

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

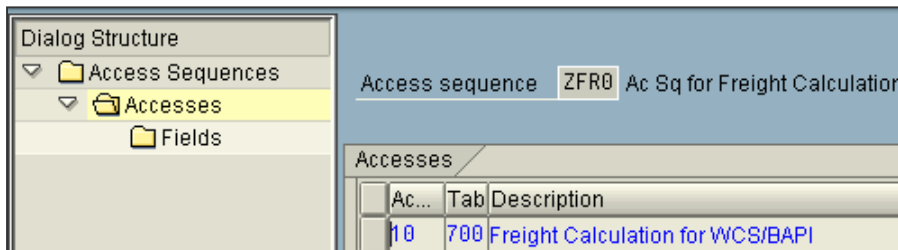


Figure 38: Access sequence ZFR0

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.

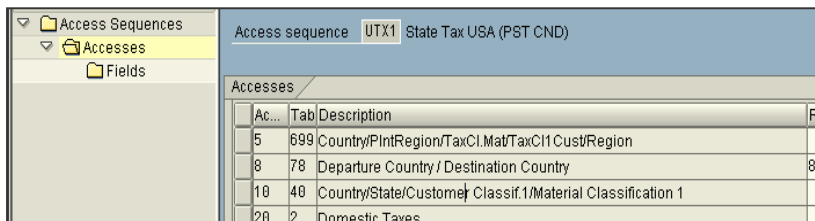


Figure 39: Access sequence UTX1

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.

The screenshot shows the SAP V/06 transaction for creating a condition type. The 'Condit. type' is set to 'ZPR0' and the description is 'Web sales'. The 'Access seq.' is 'ZPR0' with the label 'Access sequence - Web'. Below this, the 'Control data 1' section contains: 'Cond. class' set to 'B Prices', 'Calculat.type' set to 'C Quantity', 'Cond.category' is empty, 'Rounding rule' is 'Commercial', and 'StrucCond.' is empty. The 'Group condition' section has 'Group cond.' and 'RoundDiffComp' both unchecked, and 'GrpCond.routine' is empty. The 'Changes which can be made' section has 'Manual entries' set to 'D Not possible to process manually'. Under this, 'Header condit.' and 'Item condition' are unchecked, while 'Delete' is checked. 'Amount/percent' and 'Value' are unchecked, and 'Qty relation' is checked. A 'Records for access' button is located at the top right of the form.

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.

The screenshot shows the SAP V/06 transaction for creating a condition type. The 'Condit. type' is set to 'ZF00' and the description is 'Shipment cost'. The 'Access seq.' is 'ZFR0' with the label 'Ac Sq for Freight Cal'. Below this, the 'Control data 1' section contains: 'Cond. class' set to 'A Discount or surcharge', 'Calculat.type' set to 'B Fixed amount', 'Cond.category' set to 'F Freight', 'Rounding rule' is 'Commercial', and 'StrucCond.' is empty. The 'Group condition' section has 'Group cond.' and 'RoundDiffComp' both unchecked, and 'GrpCond.routine' is empty. The 'Changes which can be made' section has 'Manual entries' set to 'No limitations'. Under this, 'Header condit.' and 'Item condition' are checked, while 'Delete' is unchecked. 'Amount/percent' and 'Value' are checked, and 'Qty relation' and 'Calculat.type' are unchecked. A 'Records for access' button is located at the top right of the form.

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.

Step	Cntr	CType	Description	Fro	To	Man.	Mdt	Stat	PSub...	Reqt	Alt
810	2	HB00	Count (Value)			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S	2	
810	3	HD00	Freight			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S4	2	
815	0	ZF00	Shipment cost			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S4	2	
820	0	HM00	Order value			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S		
895	0	PDIF	Diff.value (own)			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
900	0		Net Value 2			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3		2

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 KZW15).

Step	Cntr	CType	Description	Fro	To	Man.	Mdt	Stat	PSub...	Reqt	Alt
915	0	UTXJ	Tax Jurisdict.Code			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	S5	10	
916	0	AR1	Output Tax			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
918	0		Shipping Tax			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
919	0	DIFF	Rounding Off			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
920	0		Total			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
930	0	SKT0	Cash Discount			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

Condition subtotal

SubTo	Short text
1	No separate sub-totals
2	Carry over value to KOMP-KZW11
3	Carry over value to KOMP-KZW12
4	Carry over value to KOMP-KZW13
5	Carry over value to KOMP-KZW14
6	Carry over value to KOMP-KZW15

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 KZW16). 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.

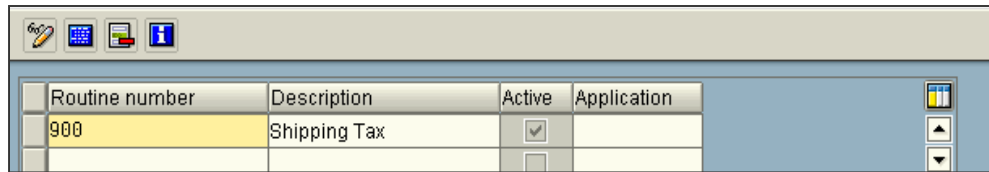


Figure 44: Entering a description in the routine 900 field

- 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.

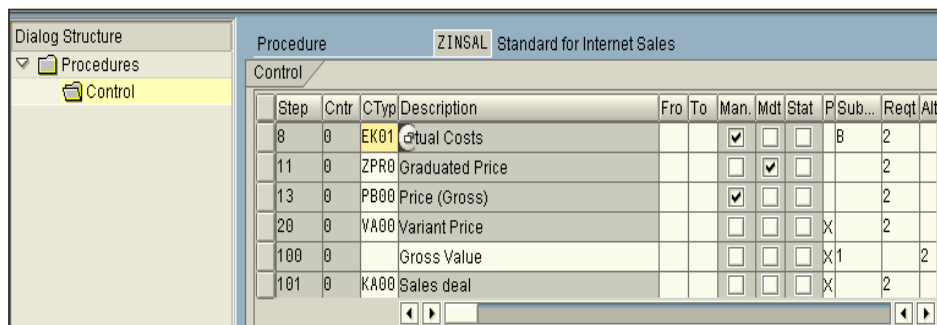


Figure 45: Saving the routine

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.

WSO1	30	30	A	1	ZINSAL	Pricing Procedure - We		
WSO1	30	30	A	2	ZINSAL	Pricing Procedure - We		

Figure 46: Assign pricing procedure

Account determination procedure - transaction code VKOA

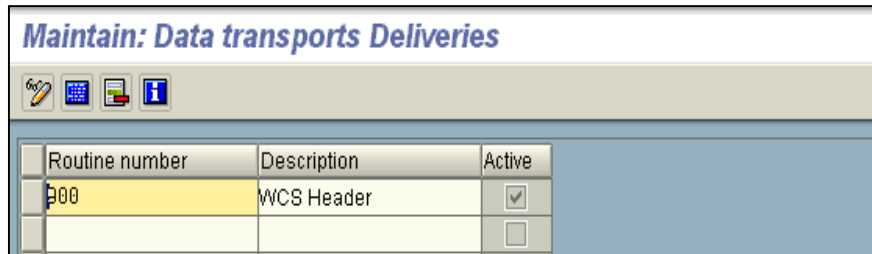
Maintain the General Ledger accounts for Sales Organization WSO1 as per the standard for Chart of Accounts maintained for Sales Organization 0001.

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.



Routine number	Description	Active
900	WCS Header	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

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**.

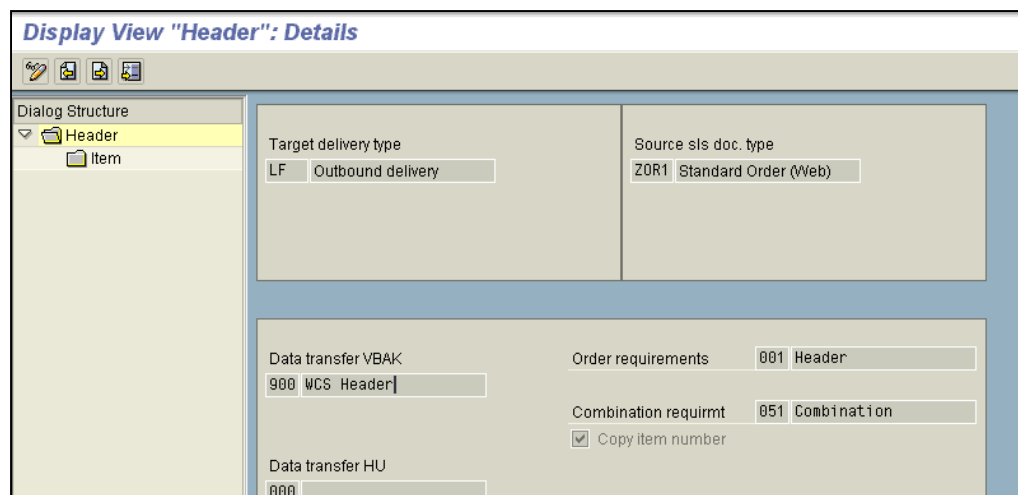


Figure 48: Display View “Header”: Details

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 the following figure. Save the entries.

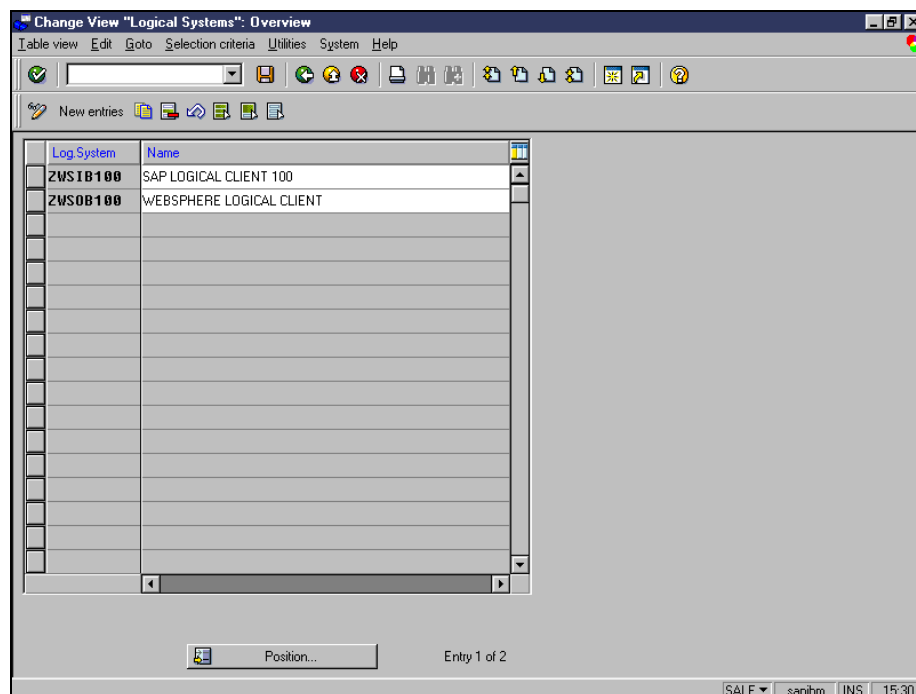


Figure 49: Defining logical systems for SAP and WebSphere Commerce

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.

Change View "Clients": Details

Table view Edit Goto Selection criteria Utilities System Help

New entries

Client **100** Client for AMS

City New Jersey Last changed by **ANANDU**

Logical system **ZWSIB100** Date **06.04.2001**

Std currency **USD**

Client role Test

Changes and transports for client-specific objects

☐ Changes without automatic recording
☒ Automatic recording of changes
☐ No changes allowed
☐ Changes w/o autom. recording, no transport allowed

Client-independent object changes

Changes to Repository and cross-client Customizing allowed

Protection: Client copier and comparison tool

Protection level 0: No restriction

SALE sapib

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.

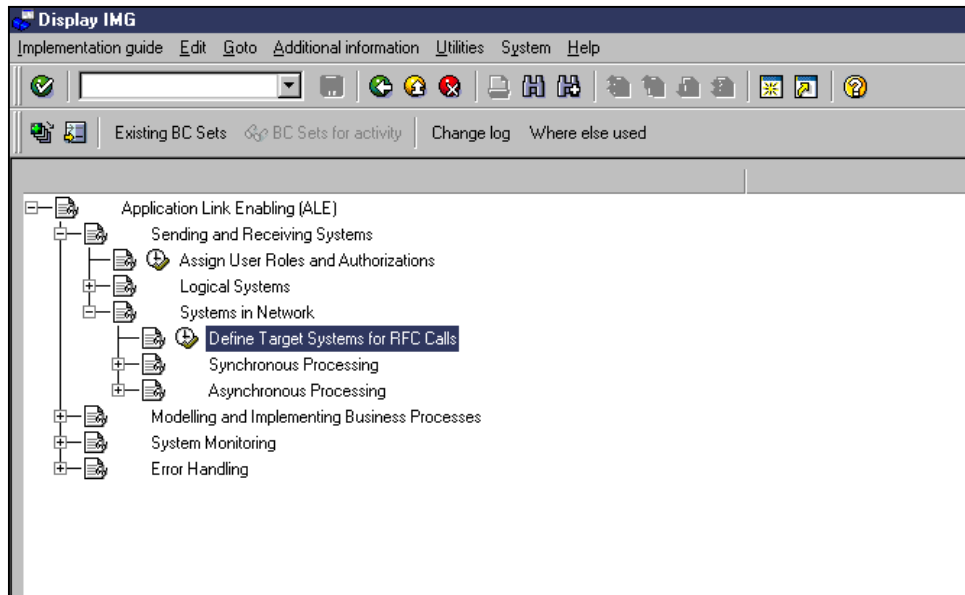


Figure 51: Defining RFC definitions: tree structure

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. 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.

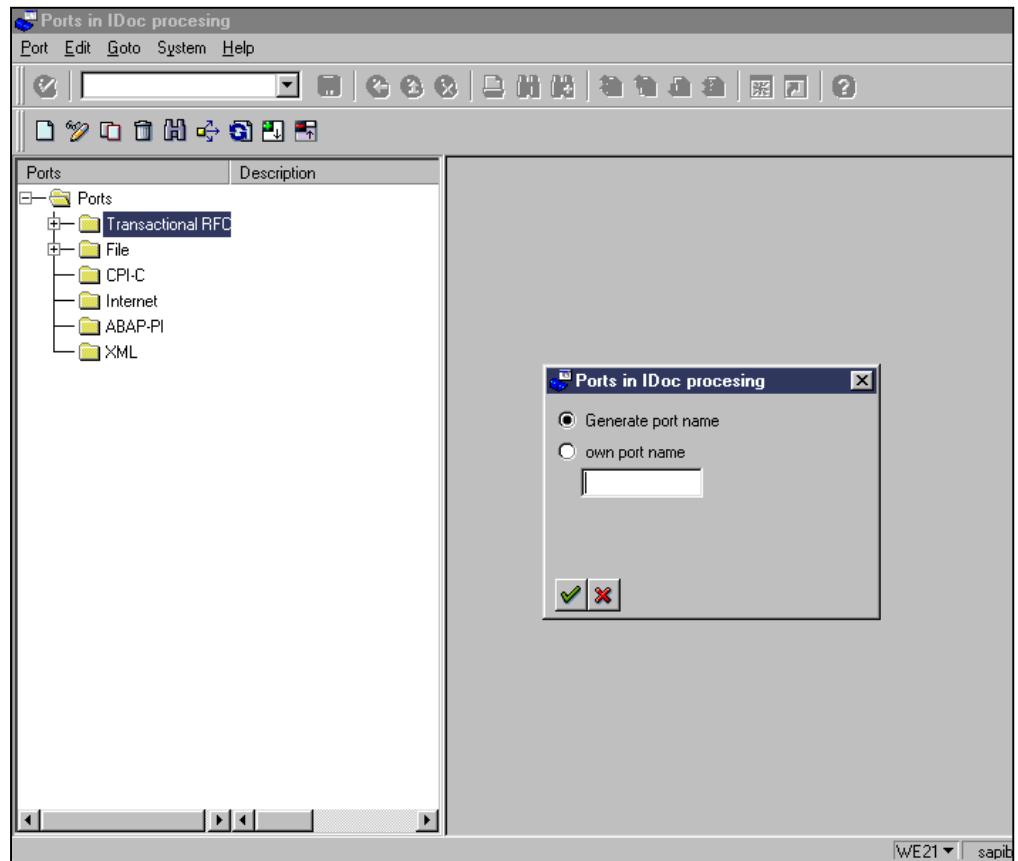


Figure 52: Selecting **Generate port name**

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

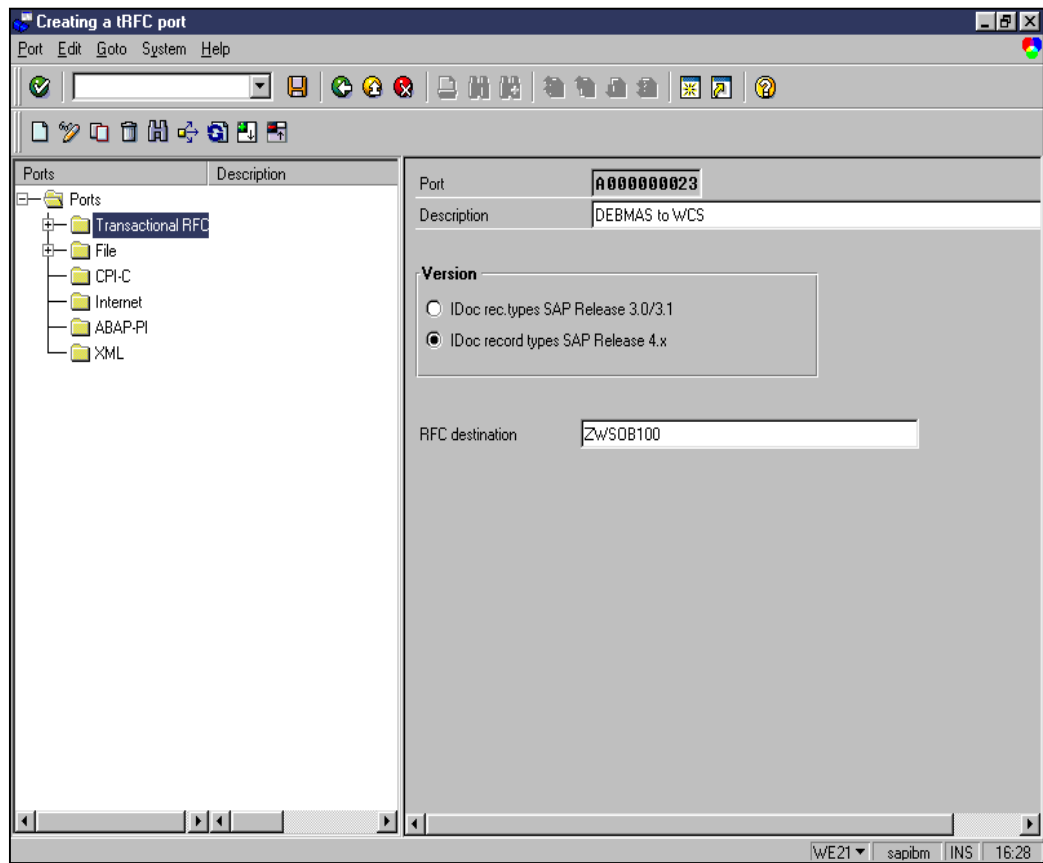


Figure 53: Creating an RFC port dialog

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.

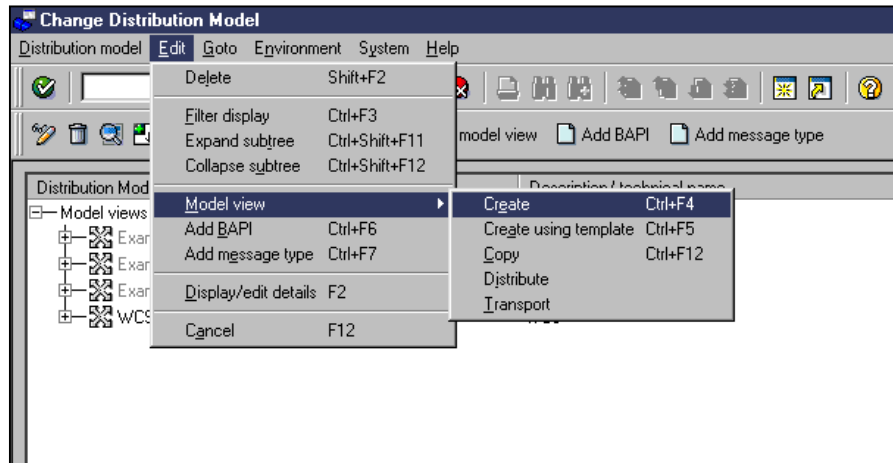


Figure 54: Preparing to create a new model

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

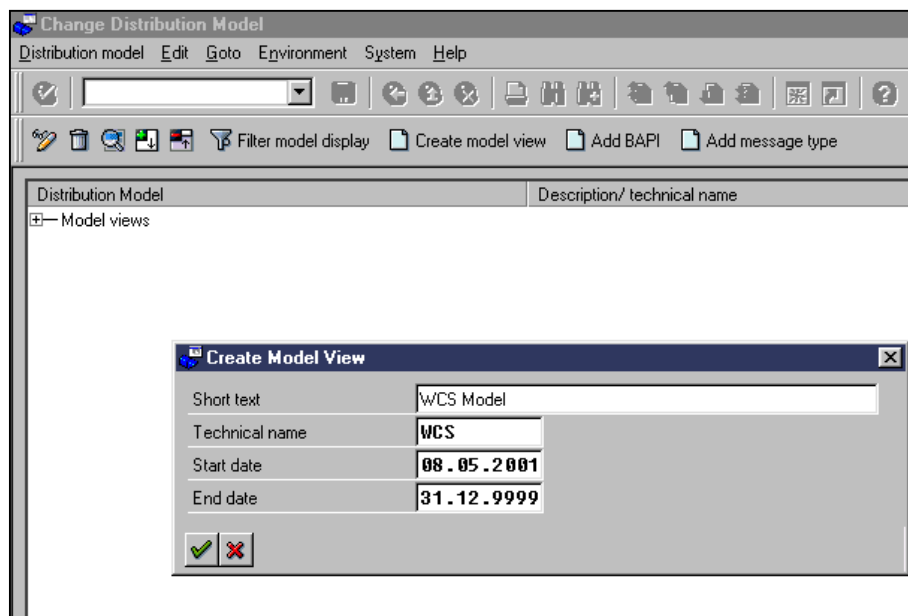


Figure 55: Creating a new model

3. Click **OK**.
4. Select the created model (WebSphere Commerce Business Edition).
5. Click **Add Message Type** as shown in Figure 56.

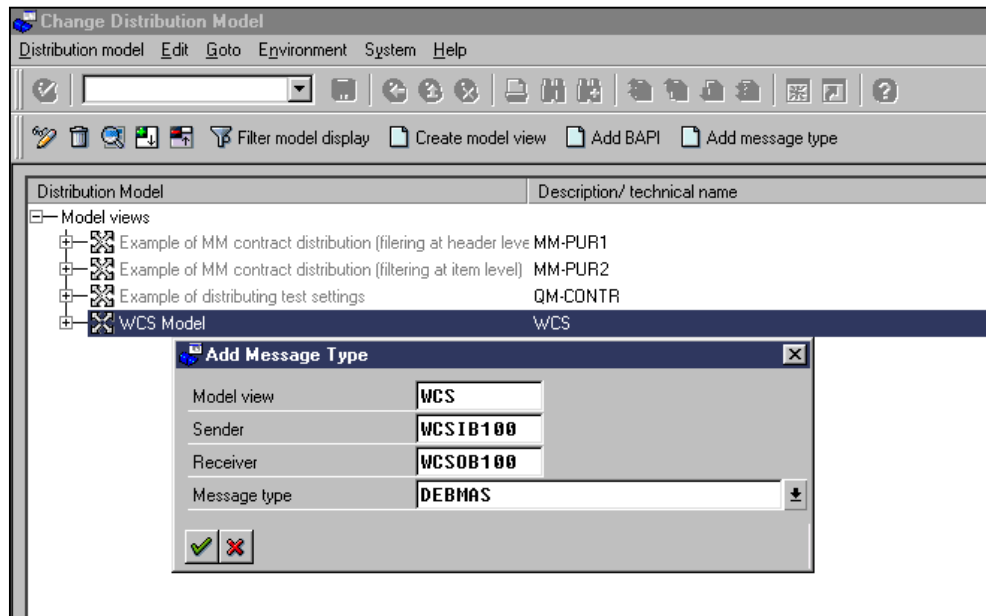


Figure 56: Adding a message type

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**.

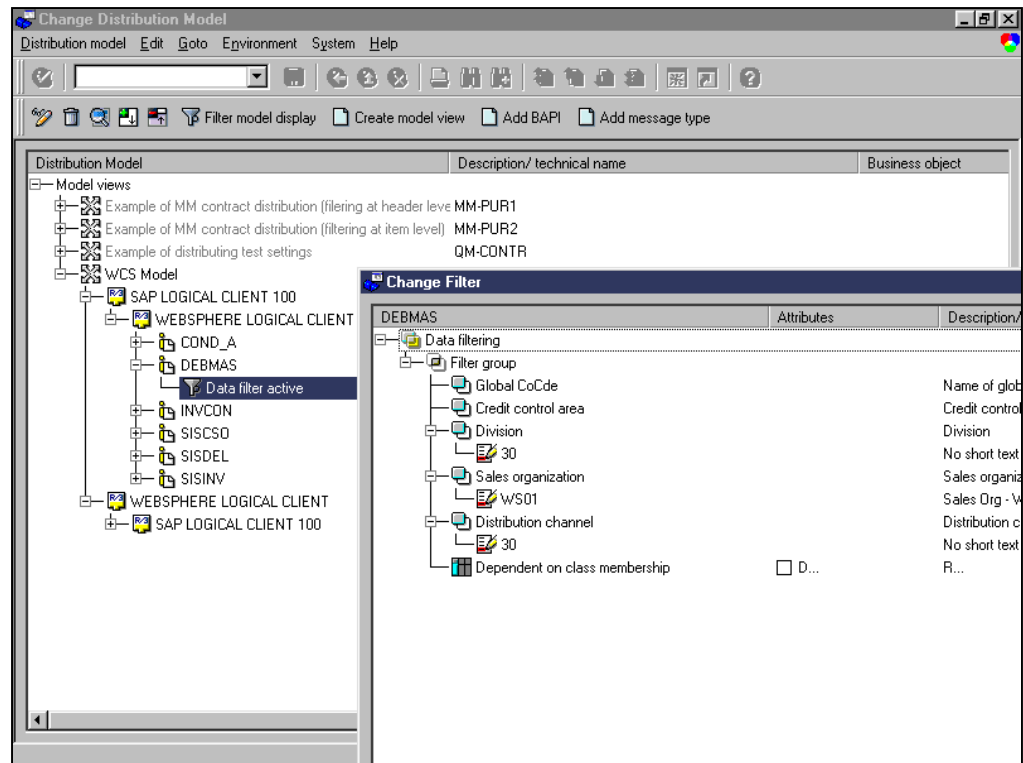


Figure 57: Creating a filter group

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.

Generating partner profile

Program Edit Goto System Help

Model view: WCS to []

Partner system: [] to []

Default parameters for partner profile

Postprocessing: Authorized processors

Type: US User

ID: RAVIA RAVIA

Outbound parmters.

Version: 3 IDoc record types from Version 4.0 onwards

PacketSize: 100 IDocs

Output mode

☒ Transfer IDoc immediately

☐ Collect IDocs and transfer

Inbound parmters.

Processing

☒ Trigger immediately

☐ Trigger by background program

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

- Click on the execution icon to run the program.

Generating partner profile

List Edit Goto System Help

Protocol for generating partner profile

Partner	
System ZWSIB100	System ZWSIB100 as a partner type already exists
System ZWSOB100	System ZWSOB100 as a partner type already exists

Port	
System ZWSOB100	Port A000000021 with RFC destination ZWSOB100 already exists

Outbound parmters.	
System ZWSOB100	Outbound parameters for message type COND_A already exist
	Outbound parameters for message type DEBMA already exist
	Outbound parameters for message type INUCON already exist
	Outbound parameters for message type SISCO already exist
	Outbound parameters for message type SISDEL already exist
	Outbound parameters for message type SISINU already exist
	Outbound parameters for message type SYNCH already exist

Figure 59: Protocol for generating a partner profile

Modifying partner profiles

- Go to transaction WE20.

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

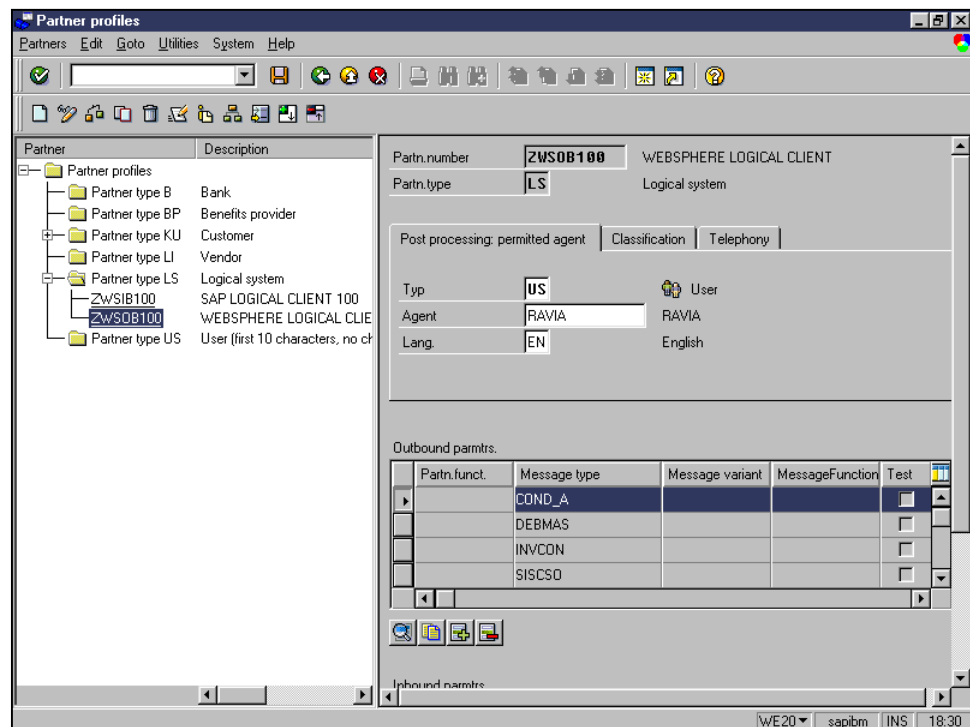


Figure 60: Modifying a partner profile: Selecting the logical system

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.

Partner profiles: Outbound parameters

Outbound parameters Edit Goto System Help

Partn.number: ZWS08100 WEBSPPHERE LOGICAL CLIENT

Partn.type: LS Logical system

Partn.funct.:

Message type: COND_A Conditions: master data for price determination

Message code:

Message function: ☐ Test

Outbound options: Message Control Post processing: permitted agent Telephony EDI standard

Receiver port: A000000021 Transactional RFC COND_A TO WCS

PacketSize: 100

Output mode

☐ Transfer IDoc immed. Output mode 4

☒ Collect IDocs

IDoc type

Basic type: COND_A01 Interchange of conditions: Master data ...

Extension:

View:

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.

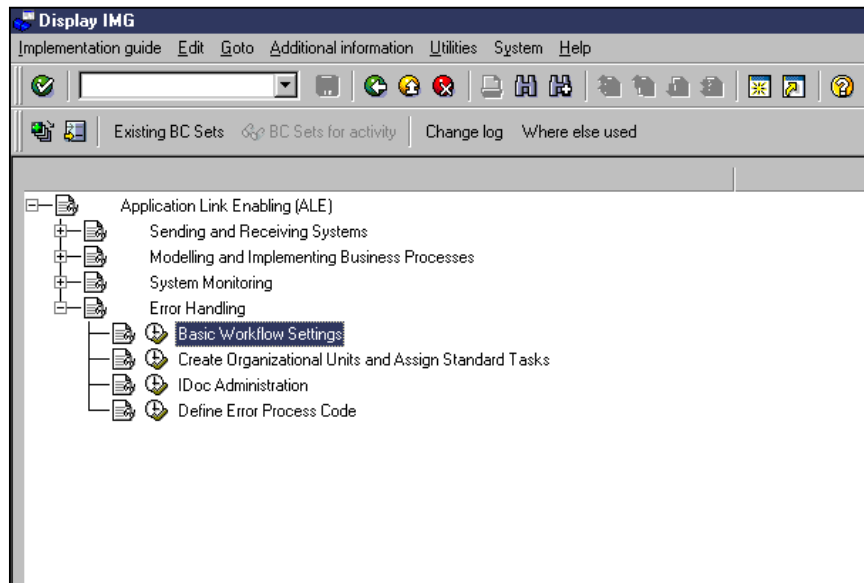


Figure 62: Selecting **Basic Workflow Settings**

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

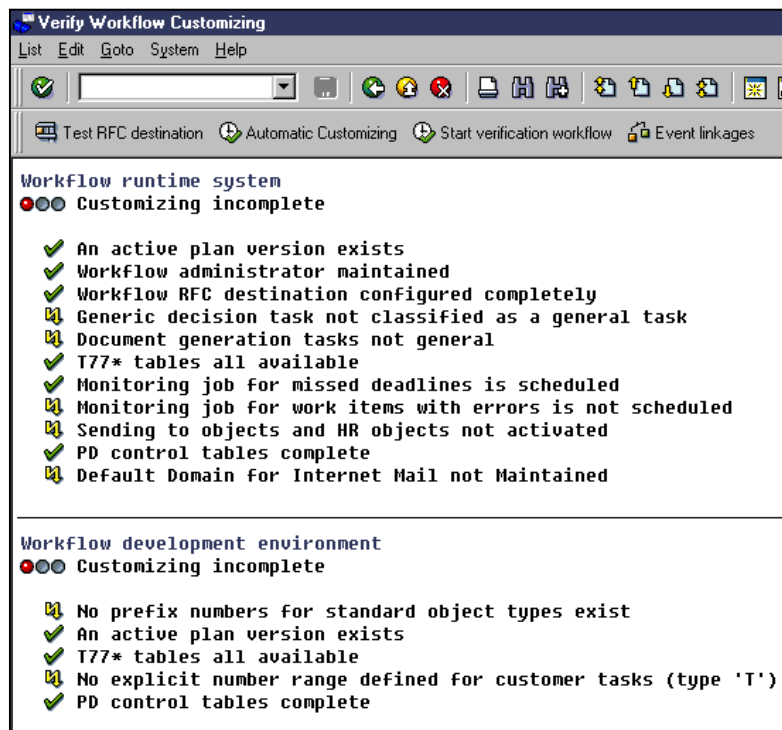


Figure 63: Verifying workflow customization

2. Click **Save**.

Enabling change pointers

1. Go to transaction BD61.
2. Select the check box and save as shown in Figure 64.

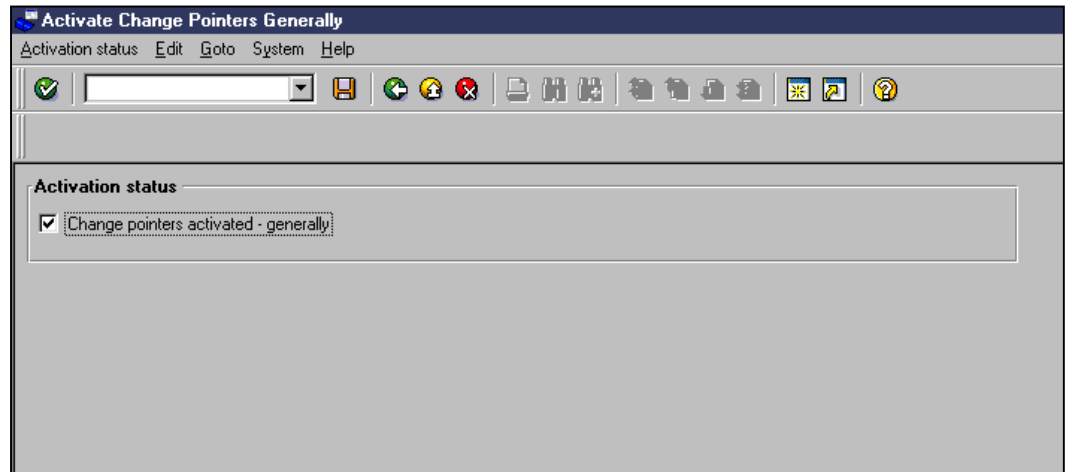


Figure 64: Enabling change pointers

Enabling change pointers for specific message types

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

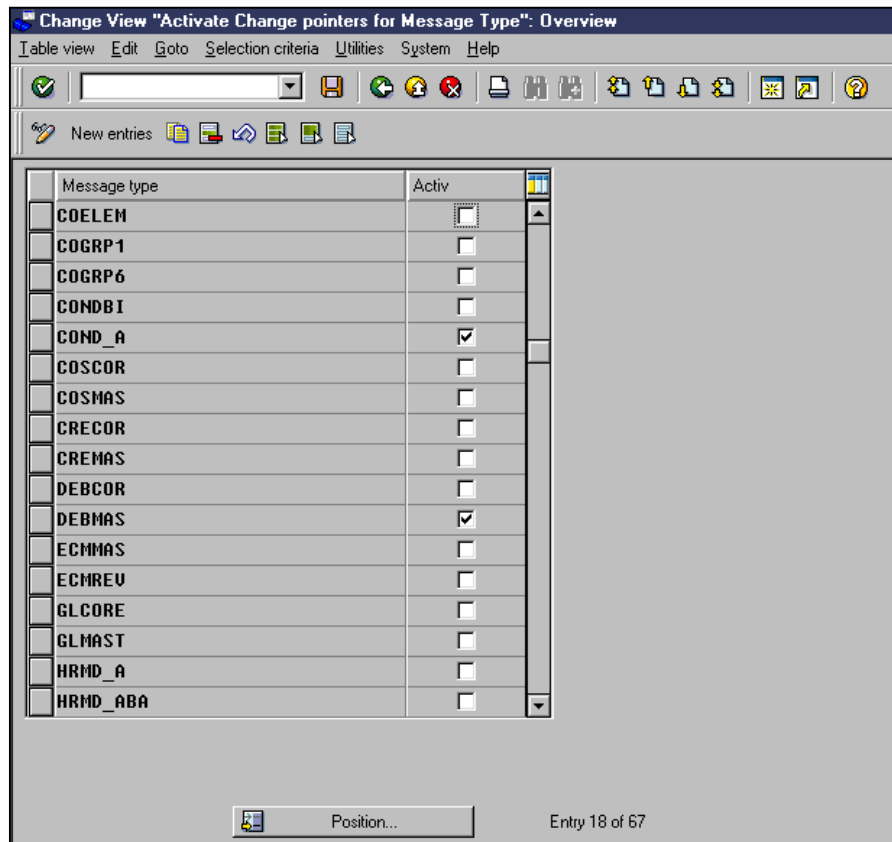


Figure 65: Enabling change pointers for message types

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.

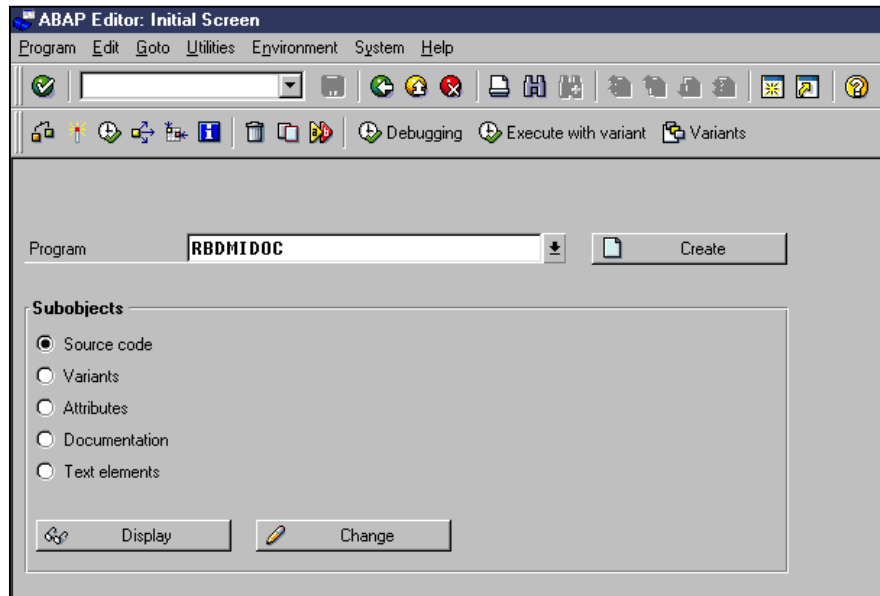


Figure 66: Running the RBDMIDOC program

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

Create Step 0

User:

Program values

ABAP program External command External program

ABAP program

Name:

Variant:

Language:

External command (command pre-defined by system administrator)

Name:

Parameters:

Operating sys.:

Target server:

External program (direct command input by system administrator)

Name:

Parameter:

Target host:

Check Save Print specifications Cancel

Figure 67: Creating variants

3. Go to transaction SM36
4. Create a job ECE_GENERATE_IDOC_FROM_CHG_PNTRS.
5. Type the JobClass and target system. Type all other settings.
6. 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.

The screenshot shows the 'Create Step 0' dialog box. The 'User' field contains 'RAVIA'. Under the 'Program values' section, the 'ABAP program' tab is active. The 'Name' field is filled with 'RBDOUTPU', 'Variant' is empty, and 'Language' is set to 'EN'. Below this, there are sections for 'External command (command pre-defined by system administrator)' and 'External program (direct command input by system administrator)', both of which are currently empty. At the bottom of the dialog, there are four buttons: 'Check' (with a green checkmark icon), 'Save' (with a floppy disk icon), 'Print specifications' (with a printer icon), and 'Cancel' (with a red X icon).

Figure 68: Running the RBDOUTPU program

2. Create a variant schedule for the job.

Transactions - master data

Create the following master data:

- 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.

Change Condition Records

Condition info | Key combination | Select using index

Condition type: ZPR0 Web sales

Key Combination

- ☒ Sales org./Distr. ch./Division/Cust. group/Material
- ☐ Customer/material with release status
- ☐ Price List Type/Currency/Material
- ☐ Material

OK Cancel

Figure 69: Creating the condition records for pricing

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

Change Web sales Condition (ZPR0) : Overview

Sales organization: WS01 Web Sales

Distribution channel: 30 Web Sales

Valid from: 24.05.2002

Material	Description	Rate	Unit	per	U...	C...	S...	Valid on
SAP00106	Material for Trial	86.00	USD	1	EA	C		09.05.2002

GAD (1) (100) sapibm OVR

Figure 70: Creating the Web sales condition

- 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:

Change Shipment cost Condition (ZF00) : Overvi...

Region of div. plant: NY
 Region: IL
 Incoterms: CFR
 Valid from: 24.05.2002

Costs and freight

Freight Calculation for WCS/BAP

Inco. 2	Bezeichnung	Rate	Unit	per	UoM	C..S.	Valid on
TRUCK		20.00	USD		1 KG	D	24.05.2002

GAD (1) (100) sapism OVR

Figure 71: Creating a record for shipment cost

1. Enter the **Region of div. 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 Figure 72 to maintain the records and press enter.

Create Condition Records

Condition info Key combination

Condition type: UTXJ Shipment cost

Key Combination

- ☒ Country/Plnt/Region/TaxCl.Mat/TaxCl1Cust/Region
- ☐ Departure Country / Destination Country
- ☐ Country/State/Customer Classif.1/Material Classification 1
- ☐ Domestic Taxes

✓ ✗

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

Create Tax Jurisdict.Code Condition (UTXJ) : Fast Entry

Country: IN India
 Region of dlv.plant: 10 Karnataka
 Tax class. material: 1 10% tax
 TaxClass1-Cust: 1 Taxable

Country/PlntRegion/TaxCl.Mat/TaxCl1 Cust/Region

R...	Bezeichnung	Rate	Valid on	Valid to	Tax...	Wt...
10	Karnataka	10,000	06.03.2002	31.12.9999	A2	

Figure 73: Creating a tax record

Before publishing the store

Unzip `CWSAPToolTechStore.zip` present in the `WCBE-CrossWorlds-SAP_Integration.zip` into any directory.

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

Populating the units of measure

This integration solution provides a massloadable XML file to upload all the units of measures defined in SAP to WebSphere Commerce. To do this, execute the following steps:

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 integration solution 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: This 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.

Modifying OrderCreateXML.jsp

The default buffer size set for the implicit object out in the jsp is 8 KB. As a result, to enable an order creation with the multiple items in the range of more than 5, we need to increase the buffer size. To do this,

1. Open OrderCreateXML.jsp from
<Drive>\Commerce_Install_Path\AppServer\installedApps\WC_Enterprise_A
pp_demo.ear\wcstores.war.

2. Insert the page directive tag for increasing the buffer size as follows:

```
<%@ page buffer="20 kb" autoFlush="false" %>
```

3. Save the file.

Publishing a store

To enable this integration solution you must create a new store. You can use the sample store provided with this integration solution.

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 CWSAPToolTech.sar file for this integration solution 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 CWSAPToolTech.sar file and Feature_cwsaptooltech_en_US.html present in the store directory, to <Drive>:/Commerce_Install_Path/samplestores/ToolTech folder, where <Drive>:/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="CWSAPToolTech.sar"  
relativePath="ToolTech">  
  
<html locale="en_US"  
featureFile="ToolTech/Feature_cwsaptooltech_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 `CWSAPToolTech.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**. It may take a few minutes to complete publishing the sar file. When complete, the status changes from **Publishing** to **Publishing completed successfully**.

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 integration solution.

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)

Sequence no.	Data at position	Data description	Corresponding fields in SAP
14	MaterialWeight	The nominal weight associated with the material	MARA-BRGEW (Optional)
15	MaterialLength	The nominal length associated with the material.	MARA-LAENG (Optional)
16	MaterialWidth	The nominal width associated with the material.	MARA-BREIT (Optional)
17	MaterialHeight	The nominal height associated with the material.	MARA-HOEHE (Optional)
18	MaterialNominalQuantity	Nominal quantity for a material, used for pricing. For example, if a material is priced as "3 for a dollar", then the nominal quantity of the material is 3, and the price of the material is one dollar	MVKE-AUMNG (Mandatory)
19	MaterialDataIndicator	An indicator that specifies whether the data for that material is for CREATE or UPDATE.	CDHDR-CHANGE_IND (Mandatory)
20	ManufacturerName	The name of the manufacturer of this material	(Optional)
21	ManufacturerPartNumber	The part number used by the manufacturer to identify this material	(Optional)

Table 3: Sequence to extract material data from SAP

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 integration solution.

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
5	Language	Language specification in SAP. An example of language specification in WCS is en_US, for English.	T002T-SPTXT (Mandatory)

Table 4: Format to extract material attribute information from SAP

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,Import_cif_Schema10.xml,Output.xml,Append
itab.txt,Import_cif_Schema11.xml,Output.xml,Append
```

Note: For information on the directory structure, refer to the `readme.txt` in `wscsw.zip`

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 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 (ORENTITY.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.

StoreIdentifier – Identifier of the store published to showcase this integration solution (STORE.STORE_ID).

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

8. Save the changes made 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 reference data onto SAP

To demonstrate this integration you can also use the reference data provided with this integration solution. 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

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

Serial number	Description	Data element as in SAP	Length	Depends on existent SAP configurations and data
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
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\data\load\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

Serial number	Description	Data element as in SAP	Length	Depends on existent SAP configurations
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. Appendix G. Sample BDC program.
4. After loading the reference data, you have to create the inventory records before placing any orders for these materials.

You have now completed the following

- Installing and configuring WebSphere Commerce
- Installing the Commerce Enhancement Pack
- Installing and configuring the WebSphere messaging system
- Installing and configuring CrossWorlds
- Loading the CrossWorlds repository
- Configuring the MQSeries connector agent
- Configuring the SAP connector agent
- Configuring collaboration objects and maps
- Configuring the database to store the CrossWorlds components definition
- Configuring the CrossWorlds queue manager
- Configuring SAP
- Publishing the store

WebSphere Commerce is now enabled to communicate with CrossWorlds. The CrossWorlds system is configured to connect to your SAP R/3 backend system. You have loaded the business objects, maps and collaboration objects, published the ToolTech store, and loaded the master data. To test the synchronous and asynchronous message flows see, Chapter 6. Verification procedure.

Chapter 5. Store customizations

This integration solution 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.

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: Modifications to the `Address.jsp` are made provide a selection box to select the country and state codes.

Shoppingcart page: Modifications to the `shoppingcart.jsp` are made to provide selection of order items and a button to check inventory. One more column is added to show the estimated available dates. This column will be blank before the user clicks the 'Check Availability' button. It will display the date provided by SAP when the user clicks the Check Availability button.

Shipping page: Modifications to `Shipping.jsp` are made to pass, remerge, merge and check parameters.

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 BYTRUCK, BYRAIL, BYAIR 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.

Chapter 6. Verification procedure

This chapter provides the procedural details for ensuring the end-to-end flow for each message. Before you begin to verify confirm that the MQSeries queue manager and the channels are running and are in the active stage. Ensure that the MQSeries and SAP connector agents are running.

Any errors can be detected in the Interchange system log, SAP connector and MQSeries connector agent logs or WebSphere Commerce Business Edition logs. In WebSphere Commerce, the logging for the messaging component must be enabled.

CustomerCreate/Update message (DEBMAS05)

To create or update customer information in SAP and sending the inbound message to WebSphere Commerce, do the following:

1. Generate a DEBMAS05 IDOC message by creating a new customer using the SAP transaction xd01 or by changing the details of an existing customer using the SAP transaction xd02.
2. Enter the details for the customer and save the details.
3. Run SAP transaction se38 to generate the IDoc. Select the program `RBDMIDOC` and run the program.
4. On the next page enter the message type as `DEBMAS` and run the program. This creates the IDoc but does not send it.
5. To send the IDoc, go to SAP transaction se38 and run the program `RBDOUTPU`.
6. Select **dispatch** on the next page and run the program.
7. Type DEBMAS05 as the basic type on the next page and run the program. This sends the DEBMAS05 IDoc.
8. To verify whether the IDoc message is sent out of the SAP system, run transaction we38 and check whether the DEBMAS generated is listed in the outbound list.
9. In WebSphere Commerce, check for the corresponding values in the `USERREG`, `ADDRESS` and `USERS` tables.
10. Login to the CWSAPToolTech store. Click on the **Account** menu and click **Change Personal Information**. Note the change in the address.

CheckInventoryAvailabilityBE request/response

To check the available inventory for the products in SAP do the following:

1. Log in to the WebSphere Commerce store as a customer, using a valid user ID. You can use the `logonId` of the customer created in the SAP side as detailed with the previous procedure.
2. Add products to the shopping cart.
3. Select the checkbox displayed against the product items in the shopping cart.
4. Click on the Check Inventory availability button.
5. The page will be refreshed with the details of available quantity and available dates for the products selected.

OrderCreate message (ORDERS05)

To create an order in WebSphere Commerce and check whether the order is created and processed correctly in the SAP system, 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. You can use the `logonId` of the customer created in SAP.
3. Add products 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. Note down the created Order Number in the Order Confirmation page.
4. To check, whether the order create XML was successfully parsed, formatted, and sent to the SAP system, run the SAP transaction `we02`. ORDERS05 IDoc message must be listed in the inbound section. The R/3 link inbound server shows the receipt of IDocs in the console.
5. Check the status under which the ORDERS05 is displayed. It should be under status 53.

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 automatically by SAP. The IDoc message for this is SISCSO01. The generation of this IDoc message can be verified using SAP transaction `we02`.
2. Open the OrderStatus page. The order number of the order created must be listed in the Confirmed section.
3. 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.
4. Login to the CWSAPToolTech store. Click on the **Order Status** menu and look for the order in the Order(s) Confirmed section.

OrderDeliveryStatus message (SISDEL01)

1. To generate the order delivery status message for the confirmed order and deliver it to the WebSphere commerce do the following:
2. To create the SISDEL01 IDoc, run SAP transaction `va02`.
3. Type the order number generated by SAP. This can be ascertained from the SISCSO01 IDoc that was generated previously.
4. When the page showing the details of the order appears, select **Sales Document - Deliver** from the menu.
5. This process creates the delivery and sends the IDoc message, unless an error occurs. The SAP client in the error log shows the errors.
6. Open the OrderStatus page. The order number of the order created must be listed in the Confirmed section.

7. To verify the status, check the ORDSTAT and ORDISTAT tables in WebSphere Commerce. The status of the corresponding order items must be 'S'.
8. Login to the CWSAPToolTech store. Click on the **Order Status** menu and look for the order in the Order(s) Shipped 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.

ProductInventoryUpdate message (INVCON01)

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

1. To generate INVCON01 IDoc, run transaction mb1c. The next page appears.
2. Enter the movement type, plant and Storage Location Details and execute (F8). The next page appears.
3. Enter the product sku number and the quantity. This message supports the inventory update of a single product only.
4. To verify in WebSphere Commerce, check the inventory of the ordered material in the INVENTORY table.

OrderInvoiceStatus message (SISINV01)

To verify the generation of the order invoice status message and process it correctly, do the following:

1. Run the transaction vf01. A page displays asking for the billing type.
2. Select Invoice (F1) from the menu. This automatically retrieves the document number for the delivery created previously.
3. If the document number does not display, then select the document number from the menu. This creates the SISINV01 IDoc.
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 CWSAPToolTech 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 the ProductPriceUpdate 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 execute (Press F8).

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 (F8).
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 at this stage.
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.
11. Login to the CWSAPToolTech store. Check the price of the order item after adding it to the shopping cart.

Appendix A. Supported messages

This section describes the asynchronous and synchronous messages supported by this integration solution. These messages are used to exchange information between the WebSphere Commerce and SAP systems. For example, order status, inventory update, and so on.

Asynchronous messages

Asynchronous messages do not elicit an immediate response to a request. For example, when WebSphere Commerce executes the `OrderCreate` message it does not expect an immediate reply from the SAP system. As a result, the buyer is not updated immediately regarding the status of the order. The following are the asynchronous messages supported by this integration solution:

- **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 message is generated by the SAP system. There are three order status messages:
 - 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.

Synchronous messages

A synchronous message elicits an immediate response to a request. For example, when WebSphere Commerce requests the SAP system to check the inventory availability, the SAP system replies immediately with the details of the inventory check. As a result the buyer does not have to wait for a response and can view the availability status before placing the order. The following is the synchronous message supported by this integration:

Inventory availability check message (outbound from WebSphere Commerce)

This message is used to check the inventory availability of a product in SAP from WebSphere Commerce.

To enable and use the `CheckInventoryAvailabilityBE` message using the CrossWorlds adapter, refer to the Commerce Enhancement Pack.

Enabling the message flow

To enable a synchronous or an asynchronous message flow, do the following:

1. Ensure that the corresponding application specific business objects (for WebSphere Commerce and SAP) and maps are loaded into the CrossWorlds repository. For details, see Table 7: Collaboration templates.
2. Open the Properties window for the MQSeries connector. Specify the WebSphere Commerce specific business objects in the Supported Business Objects tab. Select the **Agent supported** checkbox for the objects.

Specify the related generic business objects. Uncheck the **Agent supported** checkbox. For more details about the supported objects, see Appendix B. Supported generic business objects. Repeat this step for the SAP connector with the SAP specific business objects.

3. Ensure that the collaboration objects are configured and bound with the appropriate connectors. See, Appendix D. Collaboration templates for details.
4. Ensure that the business objects and solution specific maps are explicitly bound for the connectors. For details see, Appendix F. Binding maps to business objects.
5. Ensure that the collaboration, connectors, and maps are running.
6. Start the MQSeries and SAP connector agents. Initiate the message flow.

Appendix B. Supported generic business objects

Ensure that the SAPConnector, MQSeriesConnector, and the PortConnector are bound to the following generic business objects.

Note: Disable the agent support for all generic business objects

MQSeriesConnector	SAPConnector	PortConnector
Customer	ATP	ATP
CustomerPartner	ATPLine	ATPLine
InventoryLevel	Customer	Contact
InventoryLocation	CustomerPartner	Customer
Item	InventoryLevel	Item
Order	InventoryLocation	Order
OrderBillingStatus	Item	
OrderDeliveryStatus	Order	
OrderStatus	OrderBillingStatus	
PriceRecord	OrderDeliveryStatus	

Table 5: Supported generic business objects

Note: The port connector is used to bind those ports of a collaboration that do not require a process flow through the application specific connectors.

Appendix C. Supported application specific business objects

Ensure that the MQSeriesConnector and SAPConnector are bound the following application specific business objects.

Note: Enable agent support for all application specific business objects.

MQSeriesConnector	SAPConnector
MO_DataHandler_Default	MO_Server_DataHandler
MO_MQSeriesConfig	sap_bapi_material_availability
WCS_Create_WCS_Customer	sap_cond_a02
WCS_Update_WCS_Customer	sap_debmas05
WCS_Report_NC_PurchaseOrder	sap_invcon01
WCS_Update_WCS_OrderStatus	sap_orders05
WCS_Update_WCS_ProductInventory	sap_siscso01
WCS_Update_WCS_ProductPrice	sap_sisdel01
WCS_Request_WCS_BE_ProductInventory (Synchronous)	sap_sisinv01
	SAP_TransId

Table 6: Supported application specific business objects

Appendix D. Collaboration templates

The following table lists the messages, their collaboration templates and ports:

Message	Collaboration object	Collaboration template name	From port	To port	DestinationAppRetrieve	Other ports
InventoryCheck from SAP	ProductQuantityUpdate_From_Sap	InventoryLevelManager	SAPConnector	MQSeriesConnector	MQSeriesConnector	ToltemWrapper – SAPConnector
ProductPriceUpdate from sap	ProductPriceUpdate_From_Sap	PriceManager	SAPConnector	MQSeriesConnector	MQSeriesConnector	ToltemWrapper – PortConnector
OrderBillingStatus from SAP	OrderBillingStatus_From_Sap	OrderBillingStatus	SAPConnector	MQSeriesConnector	MQSeriesConnector	
OrderDeliveryStatus from sap	OrderDeliveryStatus_From_Sap	OrderDeliveryStatus	SAPConnector	MQSeriesConnector	MQSeriesConnector	
OrderConfirmation Status	OrderConfirmationStatus_From_Sap	OrderStatus	SAPConnector	MQSeriesConnector	MQSeriesConnector	
CustomerCreate/Update from sap	CustomerCreateUpdate_From_Sap	CustomerSync	SAPConnector	MQSeriesConnector	MQSeriesConnector	ToCustomerPartnerWrapper - PortConnector
CheckInventoryAvailabilityBE	SAP_CheckInventoryAvailability	ATP	External Connector Incoming map: WC_REQUEST_PROD_INV_TO_GBO Outgoing map: GBO_TO_EC_REQUEST_PROD_INV	SAPConnector	SAPConnector	ToLine – SAPConnector DestinationAppRetrieveLine – SAPConnector
OrderCreate from WebSphere Commerce	OrderCreate_From_WCBE	SalesOrderProcessing	MQSeriesConnector	SAPConnector	SAPConnector	ToCustomerWrapper – PortConnector ToContactWrapper – PortConnector ToltemWrapper - PortConnector
ProductQuantityUpdate	ProductQuantityUpdate_From_Sap.out	InventoryLevelManager	SAPConnector	MQSeriesConnector	MQSeriesConnector	ToltemWrapper – SAPConnector

Table 7: Collaboration templates

Note: When using external connectors you must explicitly specify the incoming and outgoing maps.

Appendix E. Maps with specific values for WebSphere Commerce and SAP

The following are WebSphere Commerce and SAP specific values in the maps provided with this integration solution. You can change them according to your configuration settings.

Map name	Hard coded attributes
GBO_ATPLine_To_SAP	Unit_of_measure_for_display, Plant, Checking_rule
GBO_CustomerCreate_To_WCS	Value, AddressType, Type
GBO_CustomerUpdate_To_WCS	PreferredLanguage, CustomerStatus, value, AddressType, type
Sub_GBO_DeliveryStatusLine_To_WCS	Type
GBO_OrderCreate_To_SAP	Name_of_table_structure Partner_type_of_recipient Partner_number_of_recipient Partner_type_of_sender Sender_port Partner_number_of_sender SAP_Release_for_Idoc Name_of_basic_type Idoc_type Logical_message_type EDI_message_type Partner_function_e_g_sold_to_party Qualifier_for_IDOC_date_segment IDOC_qualifier_reference_document
Sub_GBO_OrderCreate_To_SAP	Plant IDOC_object_identification_such_as_material_no_customer IDOC_qualifier_reference_document
GBO_ProductPriceUpdate_To_WCS	Precedence Published
GBO_ProductQuantityUpdate_To_WCS	MerchantID FulfillmentCenterID

Map name	Hard coded attributes
GBO_ProductQuantityUpdate_from_SAP	itemId Plant AdjustedQty InvLocationId

Table 8: Maps with WebSphere Commerce and SAP specific values

Note: Ensure that the values entered in the fields described previously are in accordance with your configuration settings.

Appendix F. Binding maps to business objects

The following tables lists the maps that you must explicitly bind to the business objects for MQSeries and SAP connectors:

MQSeries connector

Business object	Map
OrderBillingStatus	GBO_BillingStatus_To_WCS
OrderStatus	GBO_ConfirmationStatus_To_WCS
WCS_Report_NC_PurchaseOrder	GBO_OrderCreate_From_WCS
PriceRecord	GBO_ProductPriceUpdate_To_WCS
Customer	GBO_CustomerCreateUpdate_polymap
OrderDeliveryStatus	GBO_DeliveryStatus_To_WCS
InventoryLevel	GBO_ProductQuantityUpdate_To_WCS

Table 9: MQSeries connector business objects and maps

SAP connector

Business object	Map
ATPLine	GBO_ATPLine_To_SAP
sap_sisco01	GBO_ConfirmationStatus_From_SAP
sap_sisdel01	GBO_DeliveryStatus_From_SAP
sap_sisinv01	GBO_BillingStatus_From_SAP
Order	GBO_OrderCreate_To_SAP
sap_debmas05	GBO_CustomerCreateUpdate_From_SAP
sap_invcon01	GBO_ProductQuantityUpdate_From_SAP
SAP_TransId	SAP_TransId_Mgmt
sap_bapi_material_availability	GBO_ATPLine_From_SAP
sap_cond_a02	GBO_ProdcutPriceUpdate_From_SAP

Table 10: SAP connector business objects and maps

Appendix G. 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' 'WS01'.
    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' 'WS01'.
    perform chngfld using 'KOMG-VTWE' '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 that 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

IBM, the IBM logo, CrossWorlds and the CrossWorlds logo are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or foreign countries.

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

IBM	WebSphere
AIX	DB2 Universal Database
DB2	MQSeries

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

Solaris is a trademark of Sun Microsystems, Inc.

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.

Oracle and Oracle Names are registered trademark of Oracle Corporation. Oracle Objects, Oracle Workflow, Oracle8, Oracle8i, and Net8, are trademarks of Oracle Corporation.

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.