IBM WebSphere Business Integration Collaborations
for Product Information Management Version 1.1.0
IBM WebSphere Business Integration Collaborations
for Retail Message Manager Version 4.4.0

IBM

Installation Guide

> **Note!**
>
> Before using this information and the product it supports, be sure to read the general information under "Notices and Trademarks" on page 49.

Contents

Installation guide

This Installation guide describes how to install and configure components of the Product Information Management for Retailers solution. In order to implement the solution, you will need to refer to and be familiar with the following:

- System Installation Guide for Windows®
- System Installation Guide for UNIX®
- System Administration Guide
- Technical Introduction to IBM® WebSphere® InterChange Server
- Implementation Guide for WebSphere InterChange Server

Obtaining solution components

Before following the installation procedure, obtain the IBM WebSphere Business Integration Collaborations for Retail Message Manager and Product Information Management by downloading it from Passport Advantage® (www.lotus.com/passportadvantage). Refer to Passport Advantage for downloading instructions. Refer to the section "Installing the components" on page 8 for instructions on extracting and installing the components.

The artifacts of the Product Information Management for Retailers solution include the following collaborations:

- **IBM WebSphere Business Integration Collaboration for Product Information Management.** The templates included with this product do the following:
  - Validate the item data, help manage the item approval process, and route the approved item information to other enterprise systems.
  - Manage the processes used to temporarily store, update, retrieve, and delete data such as business objects during the review and approval processes. As a result, required item information is available to the trading partner as long as an item is being processed, even if the approval process spans days, weeks, or even months.
  - Extract e-mail message text, subject text, and recipient(s) from configurable attributes in a triggering business object, and use those attribute values as input to the sendEmail API. The attributes of the business object can contain the actual message text, subject text, or address(es), or point to filenames that contain those values.
  - Extract Price or PriceBracket notifications for user processing.
- **IBM WebSphere Business Integration Collaboration for Retail Message Manager.** The templates included with this product handle all dialog necessary for the following activities:
  - Communicating with UCCnet
  - Requesting worklists from UCCnet
  - Providing a streamlined process for receiving XML data from the UCCnet standard product registry
  - Initiating processing of received data
  - Sending responses back to UCCnet

Assorted business objects, maps, and other components necessary for solution operation are also included with these collaborations.

Solution components

The following tables list the components used to develop the Product Information Management for Retailers solution. Note that there are many other existing IBM WebSphere Business Integration collaborations, business objects, and maps that can be utilized in your customized solution.

Repository files

The following repository files are included with Retail Message Manager:

BIA_BO_UCCnetMM.jar
BIA_BO_UCCnetMMSamples.jar
BIA_CT_UCCnetMM.jar
BIA_NM_UCCnetMM.jar
BIA_NM_UCCnetMMSamples.jar

The following repository files are included with Product Information Management:

BIA_BO_PIM.jar
BIA_BO_PIMSamples.jar
BIA_CT_PIM.jar
BIA_NM_PIMSamples.jar
BIA_REL_PIMSamples.jar

Collaboration templates

The following collaboration templates are included:

Table 1. Collaboration templates

Name	Location
ItemValidation	BIA_CT_PIM.jar repository file
ItemCollector	BIA_CT_PIM.jar repository file
ItemDispatcher	BIA_CT_PIM.jar repository file
Process_Reviewed_Item	BIA_CT_PIM.jar repository file
Role_Email	BIA_CT_PIM.jar repository file
DataStore	BIA_CT_PIM.jar repository file
ItemStore	BIA_CT_PIM.jar repository file
IdentifierStore	BIA_CT_PIM.jar repository file
MessageStore	BIA_CT_UCCnetMM.jar repository file
UCCnet_requestWorklist	BIA_CT_UCCnetMM.jar repository file
UCCnetMessageReceive	BIA_CT_UCCnetMM.jar repository file
UCCnetMessageSend	BIA_CT_UCCnetMM.jar repository file

Business objects

The following business objects are included:

Table 2. Business objects

Name	Location
DataStoreRetail_Item	BIA_BO_PIM.jar repository file
DataStoreSampleObject	BIA_BO_PIM.jar repository file
DataStoreUCCnetGBO_identifier	BIA_BO_PIM.jar repository file

Table 2. Business objects (continued)

Name	Location
DataStoreUCCnetGBO_storable	BIA_BO_PIM.jar repository file
Retail_Item	BIA_BO_PIM.jar repository file
SampleObject	BIA_BO_PIM.jar repository file
SerialIdentifier	BIA_BO_PIM.jar repository file
SerialItem	BIA_BO_PIM.jar repository file
SerialMessage	BIA_BO_PIM.jar repository file
SerialObject	BIA_BO_PIM.jar repository file
Retail_Price	BIA_BO_UCCnetMM.jar repository file
UCCnetDTD_envelope	BIA_BO_UCCnetMM.jar repository file
UCCnetGBO_envelope	BIA_BO_UCCnetMM.jar repository file
UCCnetGBO_identifier	BIA_BO_UCCnetMM.jar repository file
UCCnetGBO_RI_S	BIA_BO_UCCnetMM.jar repository file
UCCnetGBO_storable	BIA_BO_UCCnetMM.jar repository file
UCCnetJMSDTD_envelope	BIA_BO_UCCnetMM.jar repository file
UCCnetJMSXSD_envelope	BIA_BO_UCCnetMM.jar repository file
UCCnetTPIDTD_envelope	BIA_BO_UCCnetMM.jar repository file
UCCnetTPIXSD_envelope	BIA_BO_UCCnetMM.jar repository file
UCCnetXSD_envelope	BIA_BO_UCCnetMM.jar repository file
MQWF_Retail_Item	BIA_BO_PIMSamples.jar repository file
Retail_Item_ASBO	BIA_BO_PIMSamples.jar repository file
Retail_Price_ASBO	BIA_BO_UCCnetMMSamples.jar repository file

Maps

The following maps are included in the solution:

Table 3. Maps

Name	Location
Retail_Price_to_UCCnetJMSXSD_envelope	BIA_NM_UCCnetMM.jar repository file
Retail_Price_to_UCCnetTPIXSD_envelope	BIA_NM_UCCnetMM.jar repository file
Retail_Price_to_UCCnetXSD_envelope	BIA_NM_UCCnetMM.jar repository file
Sub_UCCnetXSD_envelope_to_Retail_Item	BIA_NM_UCCnetMM.jar repository file
UCCnetDTD_envelope_to_UCCnetGBO_envelope	BIA_NM_UCCnetMM.jar repository file
UCCnetGBO_envelope_to_Retail_Item	BIA_NM_UCCnetMM.jar repository file
UCCnetGBO_envelope_to_Retail_Price	BIA_NM_UCCnetMM.jar repository file
UCCnetGBO_envelope_to_UCCnetDTD_envelope	BIA_NM_UCCnetMM.jar repository file
UCCnetGBO_envelope_to_UCCnetGBO_identifier	BIA_NM_UCCnetMM.jar repository file
UCCnetGBO_envelope_to_UCCnetGBO_storable	BIA_NM_UCCnetMM.jar repository file
UCCnetJMSDTD_envelope_to_UCCnetGBO_envelope	BIA_NM_UCCnetMM.jar repository file
UCCnetJMSXSD_envelope_to_UCCnetGBO_envelope	BIA_NM_UCCnetMM.jar repository file
UCCnetGBO_envelope_to_UCCnetTPIDTD_envelope	BIA_NM_UCCnetMM.jar repository file
UCCnetGBO_envelope_to_UCCnetTPIXSD_envelope	BIA_NM_UCCnetMM.jar repository file

Table 3. Maps (continued)

Name	Location
UCCnetGBO_envelope_to_UCCnetXSD_envelope	BIA_NM_UCCnetMM.jar repository file
UCCnetGBO_RI_S_to_UCCnetGBO_envelope	BIA_NM_UCCnetMM.jar repository file
UCCnetTPIDTD_envelope_to_UCCnetGBO_envelope	BIA_NM_UCCnetMM.jar repository file
UCCnetTPIXSD_envelope_to_UCCnetGBO_envelope	BIA_NM_UCCnetMM.jar repository file
UCCnetXSD_envelope_to_UCCnetGBO_envelope	BIA_NM_UCCnetMM.jar repository file
UCCnetGBO_envelope_to_UCCnetJMSDTD_envelope	BIA_NM_UCCnetMM.jar repository file
UCCnetGBO_envelope_to_UCCnetJMSXSD_envelope	BIA_NM_UCCnetMM.jar repository file
MQWF_Retail_Item_to_Retail_Item	BIA_NM_PIMSamples.jar repository file
Retail_Item_to_MQWF_Retail_Item	BIA_NM_PIMSamples.jar repository file
Retail_Item_to_Retail_Item_ASBO	BIA_NM_PIMSamples.jar repository file
Retail_Price_to_Retail_Price_ASBO	BIA_NM_UCCnetMMSamples.jar repository file

Other components

The following components are also included in the solution:

Table 4. Messages

Name
CwMapMessages

Table 5. Relationship

Name
CMDTOWPN

Table 6. Library file for collaboration templates

Name
BIRetail.jar

Table 7. Workflow process definitions

Name
Retail.fdl

Supported operating environments

The following operating environments are supported in this release:

- Windows 2000 (Professional, Server, or Advanced Server) with Service Pack 4
- IBM AIX® 5.2
- Solaris 8 (2.8)

Hardware prerequisites

The Product Information Management for Retailers solution has the same processor, memory, disk space, and high-availability requirements as an IBM WebSphere InterChange Server installation, as detailed in the System Installation Guide for Windows and System Installation Guide for UNIX. For installation of the

solution, two self-extracting, executable Windows files are used. The following disk space is required for downloading and extracting the two Windows files:

Table 8. Installation disk space requirements for the Product Information Management for Retailers solution

Collaboration name	Disk space required for downloaded self-extracting executable file:	Disk space required for set of extracted installer files:	Disk space required for installed files:
Retail Message Manager	24.4 MB (the self-extracting executable file can be deleted after it is run)	24.7 MB (the installer files can be deleted after they are run)	8.0 MB
Product Information Management	23.1 MB (the self-extracting executable file can be deleted after it is run)	23.5 MB (the installer files can be deleted after they are run)	6.7 MB

Software prerequisites

Ensure that you have installed the following prerequisites appropriately for your platform:

- IBM WebSphere InterChange Server (ICS), version 4.2.2.2
- IBM WebSphere Business Integration Toolset, version 4.2.2.2
- IBM DB2 Universal Database™ Enterprise Server Edition, version 8.1.2 (FixPack 2) (Oracle and Microsoft® SQL Server databases are also supported) **Note:** When using DB2®, a C compiler must be installed on the same server on which the ICS resides so that the stored procedures can be compiled during the relationship deployment. This is not required for Microsoft SQL Server or Oracle databases.
- IBM WebSphere MQ, version 5.3.0.2 with CSD 7
- IBM WebSphere MQ Workflow, version 3.4 ServicePack 2
- IBM WebSphere Business Integration Data Handler for XML, version 2.6.0.
- IBM WebSphere Business Integration Adapters, version 2.5.0, which include the following:
  - IBM WebSphere Business Integration Adapter for WebSphere MQ Workflow, version 2.6
  - IBM WebSphere Business Integration Adapter for e-Mail, version 5.3
  - IBM WebSphere Business Integration Adapter for JText, version 5.5
  - IBM WebSphere Business Integration Adapter for JDBC, version 2.5
  - IBM WebSphere Business Integration Adapter for JavaMessage Service (JMS), version 2.6
- IBM WebSphere Business Integration Adapters, version 2.4.0, which include the following:
  - IBM WebSphere Business Integration Adapter for Trading Partner Interchange, version 3.4.0
  - IBM WebSphere Business Integration Adapter for iSoft, version 1.4

Note: Systems supporting the UCCnet Document Type Definition (DTD) only (such as those using IBM WebSphere Business Integration Collaborations V4.2.x), are not upward-compatible with systems supporting both the UCCnet Document Type Definition (DTD) and UCCnet XML Schema Definition (XSD) (such as those based on IBM WebSphere Business Integration Collaborations V4.2.1.1 and later).

Installation instructions for IBM WebSphere InterChange Server, IBM WebSphere Business Integration Toolset, and IBM WebSphere MQ, are provided in the System Installation Guide for Windows or System Installation Guide for UNIX. Installation instructions for IBM WebSphere MQ Workflow, IBM WebSphere Business Integration Data Handler for XML, the database, and individual IBM WebSphere Business Integration Adapters, are detailed in their respective installation documents.

Installing and configuring the solution

This installation guide provides step-by-step instructions for installing and configuring the Product Information Management for Retailers solution. It is recommended that you proceed through the sections in order:

1. "Planning the configuration" on page 7 details the connectivity and messaging decisions.

2. "Configuring the system environment" on page 7 details how to set up the system environment to run the Product Information Management for Retailers solution on all supported platforms.

3. "Installing the components" on page 8 instructs how to import artifacts and install them in the ICS repository, ensuring that the solution code is accessible to your system.

4. "Creating and configuring the metaobjects" on page 10 instructs how to create and configure the metaobjects required to properly process UCCnet messages.

5. "Creating and configuring the connectors" on page 14 describes how to create and configure the connectors needed to operate the Product Information Management for Retailers solution.

   Notes:

   a. The term *connector*, used throughout this document, refers to the runtime portion of an IBM WebSphere Business Integration Adapter. References to specific connectors are related to specific adapters, for example, *EmailConnector* refers specifically to the runtime component of an IBM WebSphere Business Integration Adapter for e-Mail.

   b. If you are exchanging messages with UCCnet through an AS2/EDIINT interface protocol, use a TPIConnector, iSoftConnector, or JMSConnector, depending on the connectivity type used.

   c. If you are exchanging messages through the UCCnet Command Line Utility (CLU) or are testing your installation, use a JTextConnector.

   d. The term "AS2 channel connector" refers to a TPIConnector, iSoftConnector, JText*xxx*Connector or JMSConnector depending on the connectivity type used and the protocol used to exchange messages.

6. "Creating and configuring the collaboration objects" on page 31 details the collaboration objects that must be created from the collaboration templates, and how to set each collaboration object's port connections and configuration properties.

7. "Deploying the solution" on page 44 details how to deploy the solution.

8. "Configuring the relationship" on page 45 describes the database configuration entries needed for the relationship.

9. "Creating the Retail database and tables" on page 9 details how to set up the Retail database for the solution.

10. "Configuring WebSphere MQ Workflow and WebSphere MQ" on page 46 instructs how to configure WebSphere MQ Workflow and WebSphere MQ for use with the solution.

Planning the configuration

Before you install and configure the Product Information Management for Retailers solution, you must determine how you will connect to UCCnet and what message format and protocols you will use.

Connectors

The way you connect to UCCnet determines the connector that you use to communicate with it.

- If you exchange messages with UCCnet using an AS2/EDIINT interface protocol, you can use a TPIConnector, an iSoftConnector, or you can use WebSphere Business Integration Connect in conjunction with a JMSConnector.
- Use the TPI connector if you communicate with UCCnet through Trading Partner Interchange servers.
- Use the iSoftConnector if you communicate with UCCnet through an iSoft Peer-to-Peer Agent.
- Use the JMS connector if you communicate with UCCnet through WebSphere Business Integration Connect Advanced/Enterprise Edition.
- If you exchange messages through the UCCnet Command Line Utility (CLU), using WebSphere Business Integration Connect Express, or are testing your installation, you can use a JTextConnector.

Because the actual connector you use is dependent on your set up, this documentation uses AS2 channel connector as a general term for the TPIConnector, iSoftConnector, JTextConnector, or JMSConnector

Configuring the system environment

Complete the following steps to ensure that the solution code is accessible to your system. Be sure to follow the instructions appropriate for your platform where indicated.

1. Edit the CWCLASSES path, as follows:
   - **Windows**: Edit the *<WebSphereICS_installation_dir>*\bin\start_server.bat file by appending the end of the CWCLASSES path to include the BIRetail.jar file, as shown in the following example:

     ```
     CWCLASSES=...;%CROSSWORLDS%\lib\BIRetail.jar
     ```
   - **UNIX**: Edit the *<WebSphereICS_installation_dir>*/bin/CWSharedEnv.sh file and alter the CWCLASSES path to include the BIRetail.jar file, as shown in the following example:

     ```
     CWCLASSES=${CWCLASSES}:${CROSSWORLDS}/lib/BIRetail.jar
     ```

2. If you are running the System Manager from the WebSphere Studio Workbench SDK, edit the *<WebSphereICS_installation_dir>*\bin\cwtools.cfg file by adding the BIRetail.jar file and its appropriate path information to the line `classpath=` in this file, as in the following example:

   ```
   classpath=<directory_location_of_BIRetail.jar_file>\BIRetail.jar
   ```

3. Configure the ICS with email information by doing the following:
   a. Open the System Manager.
   b. Connect to the ICS, assuming it is available.
   c. Right-click on the ICS name and select **Edit Configuration**.
   d. On the **E-mail** tab, select **Connector mail** from the **E-mail send type** menu.
   e. Close the Edit Configuration window and save it when prompted.

Note: The EmailConnector requires that the E-mail collaboration template be active on the server. The E-mail collaboration template is installed by default during installation of the InterChange Server, and must always be active, although it might not appear in any Component Library, and does not appear in the System Manager System View window. If you drop the InterChange Server repository and redeploy the Item Synchronization for Suppliers solution from the System Manager, you must restore the E-mail collaboration template to the repository by entering the following command (on one line):

- **Windows operating systems:**

```
repos_copy -sICS_server_name
           -uICS_admin_ID
           -pICS_admin_password
           -ai -i<ICS_installation_path>\repository\Email.jar
           -xcompilepackage
```

- **UNIX operating systems:**

```
repos_copy -sICS_server_name
           -uICS_admin_ID
           -pICS_admin_password
           -ai -i<ICS_installation_path>/repository/Email.jar
            -xcompilepackage
```

This command activates the E-mail collaboration template and does not require a restart of the InterChange Server.

4. Stop the ICS.

Installing the components

Complete the following steps to ensure that the solution code is accessible to your system. Be sure to follow the instructions appropriate for your platform where indicated.

1. If the WebSphere Interchange Server is not already started, start it in design mode (-design parameter).

2. Install the artifacts downloaded from www.lotus.com/passportadvantage into the same directory in which the WebSphere Interchange Server is installed by performing the following steps (see the section "Obtaining solution components" on page 1 for information on how to obtain the artifacts from Passport Advantage):

   a. Move each downloaded artifact executable file to a temporary directory.

   b. Run each executable file. The following files are extracted from each:

      - media.inf
      - setup.jar
      - setupwin32.exe (installer executable file for Windows operating systems)
      - An English language readme file and translated versions of it

   c. Run the setupwin32.exe installer executable file for each artifact to install the components.

3. On UNIX systems, transfer the file `<WebSphereICS_installation_dir>\lib\BIRetail.jar` from the Windows system to the UNIX system by using File Transfer Protocol (FTP) in binary mode. Place it in the `<WebSphereICS_installation_dir>/lib/` directory.

4. Append files, as follows:

   - **Windows:** Append the file `<WebSphereICS_installation_dir>\samples\CwMapMessages.txt` to the end of

the file
`<WebSphereICS_installation_dir>\DLMs\messages\CwMapMessages.txt`.
- **UNIX:**
  a. Transfer the file
     `<WebSphereICS_installation_dir>\samples\CwMapMessages.txt` to the
     UNIX system by using FTP in ASCII mode.
  b. Append this file to the end of the file
     `<WebSphereICS_installation_dir>/DLMs/messages/CwMapMessages.txt`.
5. Import the repository files into the System Manager as follows:
   a. Create a new Integration Component Library (ICL).
   b. Right-click the new ICL name, and select **Import Repository File**.
   c. Use the Browse button to navigate to where the repository .jar files are
      located and select one of the files.
   d. Click **Open**.
   e. Click **Finish**.
   f. Repeat this process for each of the remaining repository files.

   See the *Implementation Guide for WebSphere InterChange Server* for more
   information on this process.
6. If you are using DB2, ensure that a C++ compiler has been installed on the
   system where the InterChange Server is located in order for the Relationship
   file to be successfully compiled when it is deployed.

Creating the Retail database and tables

Create a database with the three tables described below. The database name must
match the database name used in the definition of the Database URL of the
JDBCConnector

Note: These tables assume that you are using DB2 as your database. Some values
may differ for other database software. :

Table 9. Table name: SerialItems

Columns	Type	Size	Nullable	Primary key
Objectkey	VARCHAR	255 Bytes	No	Yes
Objectdata	CLOB	1 MB	No	No

Table 10. Table name: SerialMessages

Columns	Type	Size	Nullable	Primary key
Objectkey	VARCHAR	255 Bytes	No	Yes
Objectdata	CLOB	1 MB	No	No

Table 11. Table name: SerialIdentifier

Columns	Type	Size	Nullable	Primary key
Objectkey	VARCHAR	255 Bytes	No	Yes
Objectdata	CLOB	512 Bytes	No	No

Creating and configuring the metaobjects

You must configure, and in some cases create, the following metaobjects to properly process UCCnet XML messages. To perform these tasks, complete the following steps:

1. Edit the MO_DataHandler_DefaultXMLConfig metaobject by setting or adding the following attributes, then save it as MO_DataHandler_UCCnetXMLConfig.

   Note: In the table below, some values are reformatted to fit in the table cells. The actual values do not include spaces.

Table 12. Selected attribute values for MO_DataHandler_UCCnetXMLConfig metaobject

Attribute name	Column	Setting
BOPrefix	Default	• UCCnetDTD (for iSoft connectivity using the DTD XML definition) • UCCnetXSD (for iSoft connectivity using XSD XML definition) • UCCnetTPIDTD (for TPI connectivity using DTD XML definition) • UCCnetTPIXSD (for TPI connectivity using XSD XML definition) • UCCnetJMSDTD (For WebSphere Business Integration Connect-Java Message Service connectivity using DTD XML definition.) • UCCnetJMSXSD (For WebSphere Business Integration Connect-Java Message Service connectivity using XSD XML definition.)
DefaultEscapeBehavior	Default	true

DTDPath	Default	Path to Envelope.dtd file (for DTD support) or Envelope.xsd file (for XSD support), for example:
		`<WebSphereICS_installation_dir>\UCCnet\DTDs\2.2\Envelope.dtd`
		
		or
		
		`<WebSphereICS_installation_dir>\UCCnet\XSDs\uccnet\2.2\Envelope.xsd`
		
		(**Note**: The value of this attribute assumes use of the UCCnet 2.3 XSD and DTD. The attribute values shown are examples only. The actual value must be the fully qualified path to the Envelope.xsd or Envelope.dtd file on your system. The files are available from the UCCnet eRoom. You must have a valid UCCnet eRoom user ID and password to obtain the files.)

IgnoreUndefinedElements	Default	true
Validation	Default	false. (You must set this value to false because the UCCnetGBO_envelope has been extended to support extra fields. Failure to do this results in Datahandler validation errors)

2. Edit the MO_DataHandler_DefaultXMLConfig business object by setting the following attribute:

Table 13. Selected attribute value for MO_DataHandler_DefaultXMLConfig business object

Attribute name	Column	Setting
BOPrefix	Default	`Retail_Item` (The XML datahandler uses the BOPrefix value to create the filenames for the XML files written to the OutputDir, then the datahandler appends sequential numbers to the names.)

3. Edit the MO_JTextConnector_Default business object by setting the following attributes:

Table 14. Selected attribute values for MO_JTextConnector_Default metaobject

Attribute name	Column	Setting
ArchiveDir	Default	\<Name of the directory where the files will be written> (for example, `C:\IBM\WebSphereICS\connectors\JText\archive`). Create this directory if it does not already exist.
EndBODelimiter	Default	EOF
EventDataHandler	Type	MO_DataHandler_DefaultXMLConfig
EventDir	Default	\<Name of the directory where the files will be written> (for example, `C:\IBM\WebSphereICS\connectors\JText\event`). Create this directory if it does not already exist.
EventExt	Default	xml
OutputDataHandler	Type	MO_DataHandler_DefaultXMLConfig
OutputDir	Default	\<Name of the directory where the files will be written> (for example, `C:\IBM\WebSphereICS\connectors\JText\output`). Create this directory if it does not already exist.
OutputExt	Default	xml
OutputFileName	Default	Native

Edit the MO_JText Connector_Default metaobject by setting the following attributes, then save it as MO_JTextRWLConnector_Default.

Table 15. Selected attribute values for MO_JTextRWLConnector_Default metaobject

Attribute name	Column	Setting
ArchiveDir	Default	The name of the directory to archive XML files in. For example, `C:\IBM\WebSphereICS\ UCCnet\JTextRWL\archive` Create this directory if it does not already exist.
EndBODelimiter	Default	EOF
EventDataHandler	Type	MO_DataHandler_UCCnetXMLConfig
EventDir	Default	The name of the directory to obtain input XML events in. For example, `C:\IBM\WebSphereICS\ UCCnet\JTextRWL\event` Create this directory if it does not already exist.
EventExt	Default	xml
OutputDataHandler	Type	MO_DataHandler_UCCnetXMLConfig
OutputDir	Default	The name of the directory where XML events are written. For example, `C:\IBM\WebSphereICS\ UCCnet\JTextRWL\out` Create this directory if it does not already exist.
OutputExt	Default	xml
OutputFileName	Default	Native

4. If you are exchanging messages through the UCCnet CLU, WebSphere Business Integration Connect Express, or are testing your installation, follow the instructions to create a JtextConnector metaobject:

- Edit the following attributes of the MO_JTextConnector_Default metaobject and save it as MO_JText*xxx*Connector_Default where *xxx* has the following value:

  TPI If Trading Partner connector is used

  JMS If WebSphere Business Integration Connect Advanced/Enterprise Edition is used

  iSOFT If you communicate with UCCnet through an iSoft Peer-to-Peer Agent.

  xxx Not used, if you are testing or if the Command Line Utility is used.

Table 16. Selected attribute values for MO_JTextxxxConnector_Default metaobject

Attribute name	Column	Setting
ArchiveDir	Default	<Name of the directory where archive XML files will be written> (for example, `C:\IBM\WebSphereICS\connectors\JTextxxx\archive`). Create this directory if it does not already exist.
EndBODelimiter	Default	EOF
EventDataHandler	Type	MO_DataHandler_UCCnetXMLConfig
EventDir	Default	<Name of the directory to obtain input XMLs (Events)> (for example, `C:\IBM\WebSphereICS\connectors\JTextxxx\event`). Create this directory if it does not already exist.
EventExt	Default	xml
OutputDataHandler	Type	MO_DataHandler_UCCnetXMLConfig
OutputDir	Default	<Name of the directory where the XML files will be written> (for example, `C:\IBM\WebSphereICS\connectors\JTextxxx\output`). Create this directory if it does not already exist.
OutputExt	Default	xml
OutputFileName	Default	Native

- If you are exchanging messages with UCCnet through an AS2/EDIINT interface protocol, do one of the following things, depending on the connectivity type used:

  iSoft connectivity:

  Use the Business Object Designer to create a metaobject called MO_ISoftAdapterConfig with the attributes shown in the following table. The variable *my_p2p_agent_queue_manager_name* represents the name of the iSoft Peer-to-Peer Agent queue manager.

  Note: Spaces have been inserted in some entries in the following table to enable the entries to fit in the table cells. The actual entries do not include spaces.

Table 17. Attribute values for MO_ISoftAdapterConfig

Attribute name	Type	Key	Application specific information
Default	String	x	OutputQueue=queue:// *my_p2p_agent_queue_manager_name*/ *my_outbox_queue_name*; DataEncoding=Text

Table 17. Attribute values for MO_ISoftAdapterConfig (continued)

Attribute name	Type	Key	Application specific information
UCCnet_envelope_Create	String		The application specific information provided for the Default attribute might be sufficient for your installation. For additional information on this metaobject, refer to the Adapter for iSoft Peer-to-Peer Agent User Guide.

TPI connectivity:

No action is required.

Websphere Business Integration Connect-Java Message Service connectivity:

Use the Business Object Designer to create a metaobject called MO_JMSAdapterConfig with the attributes shown in the following table.

Table 18. Attribute values for MO_JMSAdapterConfig

Attribute name	Type	Key	Application specific information
Default	String	x	OutputQueue=*my_output_queue_name*; where *my_output_queue_name* specifies where to deliver the messages.
UCCnetJMSDTD_envelope (for DTD support) or UCCnetJMSXSD_envelope_create (for XSD support)	String		The application specific information provided for the Default attribute might be sufficient for your installation. For additional information on this metaobject, refer to the Adapter for JMS Agent User Guide.

5. Edit the MO_DataHandler_DefaultXMLConfig metaobject by setting the following attribute, then save it as MO_DataHandler_XMLDataStoreConfig.

Table 19. Selected attribute values for MO_DataHandler_XMLDataStoreConfig metaobject

Attribute name	Column	Setting
BOPrefix	Default	Leave blank.

6. Configure the MO_Server_DataHandler metaobject by doing the following:
   a. Replace the default attribute Dummy with an attribute that contains the following:
   ```
   Name = text_xml_datastore
   Type = MO_DataHandler_XMLDataStoreConfig
   ```
   b. Make sure that the Key field is checked for the `text_xml_datastore` attribute and the Cardinality is set to 1.
7. Edit the MO_DataHandler_Default metaobject by setting the following attribute, then save it as MO_DataHandler_UCCnet_envelope.

Table 20. Selected attribute values for MO_DataHandler_UCCnet_envelope metaobject

Attribute name	Column	Setting
text_xml	Type	MO_DataHandler_UCCnetXMLConfig

8. Set the following attributes in the EmailNotification business object:

Table 21. Selected attribute values for EmailNotification business object

Name	Column	Setting
RecipientName	Default	Email address of recipient.
FromAddress	Default	Email address of sender.

Creating and configuring the connectors

The connectors that must be created and/or configured depend on the individual installation, as follows:

- Configure the JTextRWLConnector in every installation, as detailed in "Creating and configuring the JTextRWLConnector."
- Configure the JTextConnector in every installation, as detailed in "Configuring the JTextConnector" on page 17.
- Configure, or if necessary, create and configure, one of the following connectivity connectors depending on the connectivity type you are using and the protocol you are using to exchange messages:
  - If you are exchanging messages with UCCnet through an AS2/EDIINT interface protocol and are using iSoft connectivity, configure the iSoftConnector, as detailed in "Configuring the iSoftConnector" on page 17.
  - If you are exchanging messages with UCCnet through an AS2/EDIINT interface protocol and are using TPI connectivity, configure the TPIConnector, as detailed in "Configuring the TPIConnector" on page 19.
  - If you are exchanging messages with UCCnet through an AS2/EDIINT interface protocol and are using the WebSphere Business Integration Connect interface, configure the JMSConnector, as detailed in the section "Configuring the JMSConnector" on page 21
  - If you are exchanging messages through the UCCnet CLU, WebSphere Business Integration Connect Express, or are testing your installation, create and configure the JTextISoftConnector, as detailed in the section "Creating and configuring the JText*xxx*Connector" on page 24.
- Configure the JDBCConnector in every installation, as detailed in the section "Configuring the JDBCConnector" on page 26.
- If you want to use the email capabilities of the Product Information Management for Retailers solution, configure the EmailConnector, as detailed in the section "Configuring the EmailConnector" on page 28.
- Configure two instances of the WebSphereMQWorkflowConnector, as detailed in the section "Creating and Configuring the WebSphereMQWorkflowConnectors" on page 28.
- Configure the PortConnector in every installation, as detailed in the section "Configuring the PortConnector" on page 30.

Note: The connector configuration procedures defined in the following sections assume the connector configuration information is saved to the project, where it is accessed by the connector at startup time. As an alternative, the connector configuration information can be saved to a file and the connector startup procedure can be altered to access that file. For additional information on options for starting your connectors, refer to the System Administration Guide.

Creating and configuring the JTextRWLConnector

The JTextRWLConnector is the input connector for a UCCnet_requestWorklist collaboration object. The UCCnet_requestWorklist collaboration object passes a

UCCnet worklist request to UCCnet from a file system folder. To poll for the worklist, the operating system must have a scheduler operation configured to periodically drop a fresh copy of the request .xml into the event file system folder for the JTextRWLConnector.

Creating the JTextRWLConnector: To create the JTextRWLConnector, do the following steps:

1. Open the System Manager.
2. Save the JTextRWLConnector by doing the following steps:
   a. Open the JTextConnector.
   b. Click **File > Save As > To Project**.
   c. Save the connector as JTextRWLConnector.

Configuring the JTextRWLConnector: Perform the following steps to configure the JTextRWLConnector:

1. Configure this connector to include the standard configuration property and the appropriate business objects. Use the values shown in the following tables.

Table 22. JTextRWLConnector standard property

Property name	Value
ApplicationName	JTextRWLConnector

Table 23. JTextRWLConnector Supported business objects

Business object name	Agent support required?
MO_JTextRWLConnector_Default	Yes
MO_DataHandler_UCCnet_envelope	Yes
One of the following: • UCCnetDTD_envelope (when iSoft connectivity and DTD XML definition type are used) • UCCnetTPIDTD_envelope (when TPI connectivity and DTD XML definition type are used) • UCCnetJMSDTD_envelope (when WebSphere Business Integration Connect connectivity and DTD XML definition type are used) • UCCnetXSD_envelope (when iSoft connectivity and XSD XML definition type are used) • UCCnetTPIXSD_envelope (when TPI connectivity and XSD XML definition type are used) • UCCnetJMSXSD_envelope (when WebSphere Business Integration Connect connectivity and XSD XML definition type are used)	Yes
UCCnetGBO_envelope	No

2. Save the configuration (**File > Save > To Project**), then go back to the **Associated Map** tab and set the explicit binding.

Table 24. JTextRWLConnector associated maps

Business object name	Map name
For iSoft Connectivity: • UCCnetDTD_envelope (when the DTD XML definition type is used) • UCCnetXSD_envelope (when the XSD XML definition type is used)	UCCnetDTD_envelope_to_UCCnetGBO_envelope UCCnetXSD_envelope_to_UCCnetGBO_envelope
For TPI Connectivity: • UCCnetTPIDTD_envelope (when the DTD XML definition type is used) • UCCnetTPIXSD_envelope (when the XSD XML definition type is used)	UCCnetTPIDTD_envelope_to_UCCnetGBO_envelope UCCnetTPIXSD_envelope_to_UCCnetGBO_envelope
For WebSphere Business Integration Connect: • UCCnetJMSDTD_envelope (when the DTD XML definition type is used) • UCCnetJMSXSD_envelope (when the XSD XML definition type is used)	UCCnetJMSXSD_envelope_to_UCCnetGBO_envelope UCCnetJMSXSD_envelope_to_UCCnetGBO_envelope

3. If you are using WebSphere MQ Series, ensure that the messaging properties reflect what is needed for this InterChange Server .

4. Save the configuration (**File > Save > To Project**), then close the Connector Configurator.

5. Create the JTextRWLConnector Agent as follows:

   • **Windows operating system**:

      a. In the Windows task bar, right-click **Start** and select **Open All Users**.

      b. Navigate to the folder on your system that contains the installed connectors by clicking **Programs > IBM WebSphere Business Integration Adapters > Adapters > Connectors.**

      c. Copy the JText Connector short cut and rename it to JTextRWL Connector.

      d. Right-click the JTextRWL Connector short cut and select **Properties**.

      e. Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to JTextRWL, where *ICS_server_name* refers to the name of the ICS server, as shown in the following example:

      ```
      <install_path>\IBM\WebSphereICS\connectors\JText\start_JText.bat   \
      JTextRWL ICS_server_name
      ```

   • **UNIX operating system** You can do these steps later when you actually start the Connector Agent after all the components are deployed.

      a. Change to the *<install_path>*/IBM/WebSphereICS/connectors/JText/ directory: .

      b. Run JTextRWL by entering the following command, where *ICS_server_name* refers to the name of the ICS server:

      ```
      start_JText.sh JTextRWL ICS_server_name
      ```

6. If you are using WebSphere MQ as your connector transport, create the JTextRWLConnector queues in WebSphere MQ. You must create the following queues as local queues and accept the defaults. Enter the following at a command prompt:

   ```
   runmqsc local_queue_manager_name
   DEFINE QLOCAL (AP/JTEXTRWLCONNECTOR/server_name) USAGE (NORMAL)
   DEFINE QLOCAL (IC/server_name/JTEXTRWLCONNECTOR) USAGE (NORMAL)
   END
   ```

where *local_queue_manager_name* is the name of the queue manager used by the IBM WebSphere Interchange Server and *server_name* is the InterChange Server's name.

Refer to the Adapter for JText User Guide for more information about this connector.

Configuring the JTextConnector

Perform the following steps to configure the JTextConnector:

1. Configure this connector to include the appropriate business objects. Use the values shown in the following table.

Table 25. Supported business objects

Business object name	Agent support required?
MO_DataHandler_Default	No
MO_DataHandler_DefaultXMLConfig	Yes
MO_JTextConnector_Default	Yes
Retail_Item_ASBO	Yes
Retail_Item	No

2. Save the configuration (**File > Save > To Project**), then go back to the **Associated Map** tab and set the explicit binding.

Table 26. Associated map

Business object name	Map name
Retail_Item	Retail_Item_to_Retail_Item_ASBO

3. Save the configuration (**File > Save > To Project**), then close the Connector Configurator.

4. If you are using WebSphere MQ as your connector transport, create the JTextConnector queues in WebSphere MQ. You must create the following queues as local queues and accept the defaults. Enter the following at a command prompt:

```
runmqsc local_queue_manager_name
DEFINE QLOCAL (AP/JTEXTCONNECTOR/server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/server_name/JTEXTCONNECTOR) USAGE (NORMAL)
END
```

where *local_queue_manager_name* is the name of the queue manager used by the IBM WebSphere Interchange Server and *server_name* is the interchange server's name.

Refer to the Adapter for JText User Guide for more information about this connector.

Configuring the iSoftConnector

Configure this connector only if you are using the iSoft Peer-to-Peer Agent and are communicating with UCCnet through an AS2/EDIINT interface protocol. Perform the following steps to configure the iSoftConnector:

1. Set the value of the MQSERIES_JAVA_LIB attribute in the connector startup file (`start_ISoft.bat` on Windows, `start_ISoft.sh` on UNIX) to the location of your WebSphere MQ Java™ client libraries (for instance, `C:\Program Files\IBM\MQSeries\Java\lib`).

2. Configure this connector to include the connector-specific configuration properties and appropriate business objects. Use the values shown in the following tables. The variable *my_p2p_agent_queue_manager_name* represents the name of the iSoft Peer-to-Peer Agent queue manager.

Table 27. Connector-specific properties

Property name	Value
ArchiveQueue	Queue to which copies of successfully processed messages are sent (for instance, `queue://`*my_p2p_agent_queue_manager_name*`/archive`).
Channel	WebSphere MQ server connector channel for your iSoft Peer-to-Peer Agent queue manager.
ConfigurationMetaObject	MO_ISoftAdapterConfig
DataHandlerConfigMO	MO_DataHandler_UCCnet_envelope
DefaultVerb	Create (add this property if it does not appear in the list of connector-specific properties).
ErrorQueue	Queue to which messages that could not be processed are sent (for instance, `queue://`*my_p2p_agent_queue_manager_name*`/error`).
HostName	The name of the host running the iSoft Peer-to-Peer Agent WebSphere MQ queue manager.
InputQueue	Semi-colon-delimited list of message queues that are polled by the connector for new messages (for instance, `queue://`*my_p2p_agent_queue_manager_name*`/inbox1;` `queue://`*my_p2p_agent_queue_manager_name*`/inbox2`).
InProgressQueue	Message queue where messages are held during processing (for instance, `queue://`*my_p2p_agent_queue_manager_name*`/in_progress`).
Port	Port established for the WebSphere MQ listener of the iSoft Peer-to-Peer Agent's queue manager.
UnsubscribedQueue	Queue to which messages that are not subscribed are sent (for instance, `queue://`*my_p2p_agent_queue_manager_name*e`/unsubscribed`).
UseDefaults	true (add this property if it does not appear in the list of connector-specific properties).

Table 28. Supported business objects

Business object name	Agent support required?
MO_DataHandler_UCCnet_envelope	Yes
MO_ISoftAdapterConfig	Yes
• UCCnetDTD_envelope (when the DTD XML definition type is used) • UCCnetXSD_envelope (when the XSD XML definition type is used)	Yes
UCCnetGBO_envelope	No

3. Save the configuration (**File > Save > To Project**), then go back to the **Associated Map** tab and set the explicit bindings.

Table 29. Associated maps

Business object name	Map name
UCCnetDTD_envelope (when DTD XML definition type is used)	UCCnetDTD_envelope_to_UCCnetGBO_envelope

Table 29. Associated maps (continued)

Business object name	Map name
UCCnetXSD_envelope (when XSD XML definition type is used)	UCCnetXSD_envelope_to_UCCnetGBO_envelope
UCCnetGBO_envelope	• When DTD XML definition type is used: UCCnetGBO_envelope_to_UCCnetDTD_envelope • When XSD XML definition type is used: UCCnetGBO_envelope_to_UCCnetXSD_envelope

4. Save the configuration (**File > Save > To Project**), then close the Connector Configurator.

5. If you are using WebSphere MQ as your connector transport, create the iSoftConnector queues in WebSphere MQ. You must create the following queues as local queues and accept the defaults. Enter the following at a command prompt:

```
runmqsc local_queue_manager_name
DEFINE QLOCAL (AP/ISOFTCONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/server_name/ISOFTCONNECTOR) USAGE (NORMAL)
END
```

where *local_queue_manager_name* is the name of the queue manager used by the IBM WebSphere Interchange Server and *server_name* is the InterChange Server's name.

Refer to the Adapter for iSoft Peer-to-Peer Agent User Guide for more information about this connector.

Configuring the TPIConnector

Configure this connector only if you are using the TPI server and are communicating with UCCnet through an AS2/EDIINT interface protocol. Perform the following steps to configure the TPIConnector:

1. Create the trading partner configuration file `tpcfg.txt`, which includes one tab-delimited line for each trading partner formatted as follows, and stored, for example, in `C:\IBM\WebSphereICS\connectors\TPI\tpcfg.txt`:

```
Trading Partner ID (tab) XML MIME type
```

A sample file might look like the following example:

```
#Comment lines start with #
TP1 text/xml
TP2 text/xml
```

2. Set the value of the CYCLONEHOMEDIR attribute in the connector startup file (`start_TPI.bat` on Windows, `start_TPI.sh` on UNIX) to the location of the home directory for your TPI server installation (for instance, `C:\TPISolo\`).

3. Configure this connector to include the connector-specific configuration properties and appropriate business objects. Use the values shown in the following tables.

Table 30. Connector-specific properties

Property name	Value
ArchiveProcessedDocDir	Directory where processed document metaevents are archived (for instance, `C:\TPISolo\data\uccnet2\archive`).
DataHandlerConfigMO	MO_DataHandler_UCCnet_envelope

Table 30. Connector-specific properties (continued)

Property name	Value
DefaultXMLMimeType	text/xml
DocumentOutDir	Directory location where outbound documents are written temporarily before TPI processes them (for instance, `C:\TPISolo\data\uccnet2\xmlout`).
MetaEventDir	Directory used to persist the TPI event information for recovery purposes (for instance, `C:\TPISolo\data\uccnet2\xmlin`).
PollQuantity	1 (add this property if it does not appear in the list of connector-specific properties).
TradingPartnerConfigurationFile	Fully qualified name of the trading partner configuration file created in Step 1 above (for instance, `C:\IBM\WebSphereICS\connectors\TPI\tpcfg.txt`).
WaitForMDN	false (MDNs are not supported by this solution).

Table 31. Supported business objects

Business object name	Agent support required?
MO_DataHandler_UCCnet_envelope	Yes
• UCCnetTPIDTD_envelope (when the DTD XML definition type is used) • UCCnetTPIXSD_envelope (when the XSD XML definition type is used)	Yes
UCCnetGBO_envelope	No

4. Save the configuration (**File > Save > To Project**), then go back to the **Associated Map** tab and set the explicit bindings.

Table 32. Associated maps

Business object name	Map name
UCCnetTPIDTD_envelope (when DTD XML definition type is used)	UCCnetTPIDTD_envelope_to_UCCnetGBO_envelope
UCCnetTPIXSD_envelope (when XSD XML definition type is used)	UCCnetTPIXSD_envelope_to_UCCnetGBO_envelope
UCCnetGBO_envelope	• When DTD XML definition type is used: UCCnetGBO_envelope_to_UCCnetTPIDTD_envelope • When XSD XML definition type is used: UCCnetGBO_envelope_to_UCCnetTPIXSD_envelope

5. Save the configuration (**File > Save > To Project**), then close the Connector Configurator.

6. If you are using WebSphere MQ as your connector transport, create the TPIConnector queues in WebSphere MQ. You must create the following queues as local queues and accept the defaults, where *local_WebSphere_ICS_queue_manager_name* refers to the queue manager used by the IBM WebSphere InterChange Server *ICS_server_name* refers to the name of the InterChange Server. Enter the following at a command prompt:

```
runmqsc local_WebSphere_ICS_queue_manager_name
DEFINE QLOCAL (AP/TPICONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/TPICONNECTOR) USAGE (NORMAL)
END
```

Refer to the Adapter for Trading Partner Interchange User Guide for more information about this connector.

Configuring the JMSConnector

Configure this connector only if you are communicating with UCCnet through a WebSphere Business Integration Connect interface. Perform the following steps to configure the JMSConnector:

1. Create the following channels and queues using MQ Explorer, or the runmqsc tool:

   a. Start runmqsc:

      ```
      runmqsc local_queue_manager name
      ```

   b. Define a transmission queue

      ```
      DEFINE QLOCAL(xmit q_name)  USUAGE(XMITQ) MAXMSGL(10485760)
      ```

   c. Define a sender channel for sending messages to WebSphere Business Integration Connect

      ```
      DEFINE CHANNEL(channel_name) CHLTYPE(SDR) XMITQ(xmit q_name) CONNAME
          ('IP_address_of_Business_Integration_Conenct_machine(listener_port)') MAXMSGL(10485760)
      ```

      Note: On the Business Integration Connect machine, you must have a channel of type "Server Connection" defined with the same name you define for the channel here.

   d. Define a receiver channel for receiving messages from WebSphere Business Integration Connect

      ```
      DEFINE CHANNEL (channel_name) CHLTYPE (RCVR)  MAXMSGL(10485760)
      ```

   e. Define a Remote queue

      ```
      DEFINE QREMOTE(q_name) RNAME(name_of_input_queue_for_Business_Integration_Connect)
          RQMNAME(QM_name_used_by_Business_Integration_Connect)  XMITQ(name_of_transmission_queue)
      ```

   f. Define a local queue for receiving messages

      ```
      DEFINE QLOCAL(Input_qname) MAXMSGL(10485760) DEFPSIST(YES)
      ```

2. Create the JMS bindings files:

   a. Modify the *<MQ_JAVA_INSTALL_PATH>*\bin\JMSAdmin.config file as follows:

      1) Comment out the following line:

         ```
         INITIAL_CONTEXT_FACTORY=com.sub.jndi.ldap.LdapCtxFactory
         ```

      2) Uncomment the following line:

         ```
         INITIAL_CONTEXT_FACTORY=com.sub.jndi.fscontext.RefFSContextFactory
         ```

      3) Set the PROVIDER_URL variable to point to the directory where you want the JMS bindings file to be placed. For example:

         ```
         PROVIDER_URL=file:/C:/IBM/WebSphereICS/connectors/JMS
         ```

         Note: This directory must exist, and you must have write permission for it.

      4) Leave the rest of the file as it is.

   b. Save your changes.

   c. Create a new file called WBIRetail.jdni with the following content:

      ```
      *
      *  JDNI Configured Objects for Information Pipeline
      *  Used by MQSeries JMS*
      *
      *
      DEFINE QCF(WBIC_JMSQCF)  +
      TRAN(client) HOST(HOST_NAME) PORT(1414)  +
      CHANNEL(CHANNEL_NAME)  CLIENTID(' ')
      ```

```
       *
       *  Queues
       *
       DEFINE Q(INPUTQ)   +
       QUEUE(INPUTQ)   +
       TARGCLIENT(JMS)
       *
       DEFINE Q(REPLYQ)   +
       QUEUE(REPLYQ)   +
       TARGCLIENT(JMS)
       *
       DEFINE Q(REMOTEQ) +
       QUEUE(REMOTEQ) +
       TARGCLIENT(JMS)
       *
       DEFINE Q(ERRORQ) +
       QUEUE(ERRORQ) +
       TARGCLIENT(JMS)
       *
       DEFINE Q(IN_PROGRESSQ) +
       QUEUE(IN_PROGRESSQ) +
       TARGCLIENT(JMS)
       *
       DEFINE Q(UNSUBSCRIBEQ) +
       QUEUE(UNSUBSCRIBEQ) +
       TARGCLIENT(JMS)
```

Where *HOST_NAME* is the name of the host machine, *CHANNEL_NAME* is the name of the communications channel, and *INPUTQ, REPLYQ, REMOTEQ* , and the other indicated variables are the queue names.

d. Save the file. The location that you save the file to does not matter, as long as it is available when you run the batch file in step 2f.

e. Move to the *<MQ_JAVA_INSTALL_PATH>*\bin directory.

f.

On UNIX operating systems:

Run the JMSAdmin.bat file with the following syntax:

```
./JMSAdmin.sh < WBIRetail.jndi
```

where WBIRetail.jndi is the file that you created in step 2d.

On Windows operating systems:

Run the JMSAdmin.bat file with the following syntax:

```
JMSAdmin.bat < WBIRetail.jndi
```

Where WBIRetail.jndi is the file that you created in step 2d.The JMSAdmin.bat and JMSAdmin.sh files create the JMS bindings files, called .bindings, and puts it in the directory specified by the PROVIDER_URL variable of the JMSAdmin.config file.

3. Configure this connector to include the connector-specific configuration properties and appropriate business objects. Use the values shown in the following tables. The variable *my_queue_manager_name* represents the name of the name of the queue manager defined for the WebSphere Interchange Server.

Table 33. Connector-specific properties

Property name	Value
ConfigurationMetaObject	MO_JMSAdapterConfig

Table 33. Connector-specific properties (continued)

Property name	Value
CTX_InitialContextFactory	com.sun.jndi.fscontext.RefFSContextFactory (This is the same name that you uncommented in the JMSAdmin.config file in step 2a2 on page 21.
CTX_ProviderURL	The fully-qualified path, in full URL format, to the directory that contains the JMS bindings file. This name must match the value of the PROVIDER_URL in the JMSAdmin.config file. For example, `file:/C:/IBM/WebSphereICS/connectors/JMS`
DataHandlerConfigMO	MO_DataHandler_UCCnet_envelope
DataHandlerMimeType	text/xml
Error Queue	The name of the queue, in full URL format, that receives messages that cannot be processed. For example, `queue://my_queue_manager_name/ERROR`
InProgressQueue	The name of the queue, in full URL format, that holds messages during processing. For example, `queue//my_queue_manager_name/IN_PROGRESSQ`
InputQueue	The name of the queue, in full URL format, that the connector polls for new messages. For example, `queue://my_queue_manager_name/INPUTQ`
PollQuantity	1
QueueConnectionFactoryName	The name of the queue connection factory. For example, `WBIC_JMSQCF`.
ReplyToQueue	The name of the queue, in full URL format, to which replies are sent. For example, `queue://my_queue_manager_name/REPLYQ`
UnsubscribedQueue	The name of the queue, in full URL format, that receives unsubscribed messages. For example, `queue://my_queue_manager_name/UNSUBSCRIBEQ`

Notes:

a. Add the PollQuantity property if it does not appear in the list of connector-specific properties.

b. All the queue names defined by the connector-specific properties must be listed in the WBI_Retail.jndi file

Table 34. Supported business objects

Business object name	Agent support required?
MO_DataHandler_UCCnet_envelope	Yes
MO_JMSAdapterConfig	Yes
UCCnetGBO_envelope	No
• UCCnetJMSDTD_envelope (when the DTD XML definition type is used) • UCCnetJMSXSD_envelope (when the XSD XML definition type is used)	Yes

4. Save the configuration (**File > Save > To Project**), then go back to the **Associated Map** tab and set the explicit bindings.

Table 35. Associated maps

Business object name	Map name
UCCnetJMSDTD_envelope (when the DTD XML definition type is used)	UCCnetJMSDTD_envelope_to_UCCnetGBO_envelope
UCCnetJMSXSD_envelope (when the XSD XML definition type is used)	UCCnetJMSXSD_envelope_to_UCCnetGBO_envelope
UCCnetGBO_envelope	• When the DTD XML definition type is used: UCCnetGBO_envelope_to_UCCnetJMSDTD_envelope • When the XSD XML definition type is used: UCCnetGBO_envelope_to_UCCnetJMSXSD_envelope

5. Save the configuration (**File > Save > To Project**), then close the Connector Configurator.

6. If you are using WebSphere MQ as your connector transport, create the JMSConnector queues in WebSphere MQ. To do this, type the following statements at a command prompt:

```
runmqsc local_queue_manager_name
DEFINE QLOCAL (AP/JMSCONNECTOR/server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/server_name/JMSCONNECTOR) USAGE (NORMAL)
END
```

where *local_queue_manager* is the name of the queue manager used by the IBM WebSphere Interchange Server, and *server_name* is the interchange server's name.

Refer to the *Adapter for JMS User Guide* for more information about this connector.

Creating and configuring the JText*xxx*Connector

Create and configure this connector only if you are using WebSphere Business Integration Connect Express, communicating with UCCnet through the UCCnet CLU or are testing your installation. This is a copy of the JTextConnector. The JText*xxx*Connector uses the IBM WebSphere Business Integration Data Handler for XML and generates the same XML output.

Replace *xxx* with TPI, iSoft, or JMS, depending on the connection you are using.

Creating the JText*xxx*Connector: To create the JText*xxx*Connector, complete the following steps:

1. Create the JText*xxx*Connector Agent, as follows:
   - **Windows**:
     a. In the Windows task bar, right-click **Start** and select **Open All Users**.
     b. Navigate to the folder on your system that contains the installed connectors by clicking **Programs > IBM WebSphere Business Integration Adapters > Adapters > Connectors.**
     c. Copy the JText Connector short cut and rename it to JText*xxx* Connector.
     d. Right-click the JText*xxx* Connector short cut and select **Properties**.
     e. Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to JText*xxx*, where *ICS_server_name* refers to the name of the ICS server, as shown in the following example:

        ```
        <install_path>\IBM\WebSphereICS\connectors\JText\start_JText.bat   \
        JTextxxx ICS_server_name
        ```
   - **UNIX**:

    a. Access the JText*xxx*Connector Agent program located in the following directory: *<install_path>*/IBM/WebSphereICS/connectors/JText/.

    b. Run JText*xxx* by switching to this directory and entering the following command, where *ICS_server_name* refers to the name of the ICS server:

```
start_JText.sh JTextxxx ICS_server_name
```

2. Create the JText*xxx*Connector using the System Manager, as follows:

    a. Open the System Manager.

    b. Save the JText*xxx*Connector by doing the following:

        1) Open the JTextConnector.

        2) Click **File > Save As > To Project**.

        3) Save the connector as JText*xxx*Connector.

3. Create the JText*xxx*Connector queues in WebSphere MQ. You must create the following queues as local queues and accept the defaults, where *local_WebSphere_ICS_queue_manager_name* refers to the queue manager used by the IBM WebSphere InterChange Server and *ICS_server_name* refers to the name of the InterChange Server. Enter the following at a command prompt:

```
runmqsc local_WebSphere_ICS_queue_manager_name
DEFINE QLOCAL (AP/JTEXTxxxCONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/JTEXTxxxCONNECTOR) USAGE (NORMAL)
END
```

Configuring the JText*xxx*Connector: Perform the following steps to configure the JText*xxx*Connector:

1. Configure this connector to include the standard configuration property and appropriate business objects. Use the values shown in the following tables.

Table 36. Standard property

Property name	Value
ApplicationName	JText*xxx*Connector

Table 37. Supported business objects

Business object name	Agent support required?
UCCnetGBO_envelope	No
One of the following: • UCCnetDTD_envelope (when the DTD XML definition type is used) • UCCnetXSD_envelope (when the XSD XML definition type is used) • UCCnetDTD_envelope (when iSoft connectivity and the DTD XML definition type is used) • UCCnetXSD_envelope (when iSoft connectivity and the XSD XML definition type is used) • UCCnetTPIDTD_envelope (when TPI connectivity and the DTD XML definition type is used) • UCCnetTPIXSD_envelope (when TPI connectivity and the XSD XML definition type is used) • UCCnetJMSDTD_envelope (when WebSphere Business Integration Connect and the DTD XML definition type is used) • UCCnetJMSXSD_envelope (when WebSphere Business Integration Connect and the XSD XML definition type is used)	Yes
MO_DataHandler_UCCnet_envelope	Yes

Table 37. Supported business objects (continued)

Business object name	Agent support required?
MO_JText*xxx*Connector_Default	Yes

2. Save the configuration (**File > Save > To Project**), then go back to the **Associated Map** tab and set the explicit bindings.

Table 38. Associated maps

Business object name	Map name
UCCnetDTD_envelope (when the DTD XML definition type is used)	UCCnetDTD_envelope_to_UCCnetGBO_envelope
UCCnetXSD_envelope (when the XSD XML definition type is used)	UCCnetXSD_envelope_to_UCCnetGBO_envelope
UCCnetTPIDTD_envelope (when TPI connectivity and the DTD XML definition type is used)	UCCnetTPIDTD_envelope_to_UCCnetGBO_envelope
UCCnetTPIXSD_envelope (when TPI connectivity and the XSD XML definition type is used)	UCCnetTPIXSD_envelope_to_UCCnetGBO_envelope
UCCnetJMSDTD_envelope (when WebSphere Business Integration Connect and the DTD XML definition is used)	UCCnetJMSDTD_envelope_to_UCCnetGBO_envelope
UCCnetJMSXSD_envelope (when WebSphere Business Integration Connect and the XSD XML definition type is used)	UCCnetJMSXSD_envelope_to_UCCnetGBO_envelope
UCCnetGBO_envelope	One of the following: • When iSoft connectivity and DTD XML definition type used: UCCnetGBO_envelope_to_UCCnetDTD_envelope • When iSoft connectivity and XSD XML definition type used: UCCnetGBO_envelope_to_UCCnetXSD_envelope • When TPI connectivity and DTD XML definition type used: UCCnetGBO_envelope_to_UCCnetTPIDTD_envelope • When TPI connectivity and XSD XML definition type used: UCCnetGBO_envelope_to_UCCnetTPIXSD_envelope • When WebSphere Business Integration Connect and DTD XML definition type used: UCCnetGBO_envelope_to_UCCnetJMSDTD_envelope • When WebSphere Business Integration Connect and XSD XML definition type used: UCCnetGBO_envelope_to_UCCnetJMSXSD_envelope

3. Save the configuration (**File > Save > To Project**), then close the Connector Configurator.

Configuring the JDBCConnector

Perform the following steps to configure the JDBCConnector:

1. Assuming use of DB2 on a Windows system, edit the `start_JDBC.bat` file, as follows:

   a. Copy the `db2java.zip` file from the DB2 SQLLIB\java directory into the WebSphere Interchange Server\lib directory.

   b. Modify the SET JDBCDRIVERPATH line, as follows:

   ```
   SET JDBCDRIVERPATH="%CROSSWORLDS%"\lib\db2java.zip
   ```

   c. Add the following line:

```
SET DB2BIN=C:\Program Files\IBM\SQLLIB\BIN
```

   d. Add the following to the parameter -Djava.library.path near the end of
      the file:

```
%DB2BIN%
```

2. Configure this connector to include the standard configuration property,
   connector-specific configuration properties, and appropriate business objects.
   Use the values shown in the following tables.

Table 39. Standard property

Property name	Value
PollFrequency	No

Table 40. Connector-specific properties

Property name	Property value
Application User Name	Database user ID
Application Password	Database user password
ArchiveProcessed	false
EventTableName	null
Database URL	jdbc:db2:*database_name*; **Note**: The value provided assumes the database used is DB2. The database name represents the Retail database and its tables that are specified in the section "Creating the Retail database and tables" on page 9. If the value of the connector's AutoCommit attribute is false, the database URL must be appended with SelectMethod=cursor (for instance, jdbc:db2:*database_name*;SelectMethod=cursor).
RDBMS vendor	IBMDB2
JDBCDriverClass	COM.ibm.db2.jdbc.app.DB2Driver
PollQuantity	1 (add this property if it does not appear in the list of connector-specific properties)

Table 41. Supported business objects

Business object name	Agent support required?
SerialItem	Yes
SerialMessage	Yes
SerialObject	No
SerialIdentifier	Yes

3. Save the configuration (**File > Save > To Project**).

4. If you are using WebSphere MQ as your connector transport, create the
   JDBCConnector queues in WebSphere MQ. You must create the following
   queues as local queues and accept the defaults. Enter the following at a
   command prompt:

```
runmqsc local_queue_manager_name
DEFINE QLOCAL (AP/JDBCCONNECTOR/server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/server_name/JDBCCONNECTOR) USAGE (NORMAL)
END
```

where *local_queue_manager_name* is the name of the queue manager used by the IBM WebSphere Interchange Server and *server_name* is the name interchange server's name.

See the Adapter for JDBC User Guide for more information on configuring a JDBCConnector.

Configuring the EmailConnector

The EmailConnector requires that the E-mail collaboration template be active on the server. The E-mail collaboration template is installed by default during installation of the ICS, and must always be active, although it might not appear in any Component Library, and does not appear in the System Manager System View window. If it is necessary to drop the ICS repository and redeploy the Product Information Management for Retailers solution from the System Manager, you must restore the E-mail collaboration template to the repository by entering the following command:

- **Windows:**

```
repos_copy -sICS_server_name -uICS_admin_ID -pICS_admin_password \
-ai -i<ICS_installation_path>\repository\Email.jar -xcompilepackage
```

- **UNIX:**

```
repos_copy -sICS_server_name -uICS_admin_ID -pICS_admin_password \
-ai -i<ICS_installation_path>/repository/Email.jar -xcompilepackage
```

This command activates the E-mail collaboration template and does not require a restart of the ICS.

Configure the EmailConnector to include the following connector-specific configuration properties. Use the values shown in the following table.

Table 42. Connector-specific properties

Property name	Property value
SMTP_MailHost	SMTP Mail Server
DataHandlerConfigMO	Leave this field blank

Save the configuration after you make the updates (**File > Save > To Project**).

Refer to the *Adapter for e-Mail User Guide* for more information about this connector.

Creating and Configuring the WebSphereMQWorkflowConnectors

The following steps detail how to create and to configure the two instances of the WebSphereMQWorkflowConnector. Configure one instance first. Then make a copy of it to create the second instance.

1. Configure this connector to include the connector-specific configuration properties and appropriate business objects. Use the values shown in the following tables.

Table 43. Connector-specific properties

Property name	Property value
MQSeriesHostname	*system_hostname* of the host where the WebSphere MQ Workflow queue manager is running.
MQSeriesPort	The listener port that is defined for the WebSphere MQ Workflow queue manager.
ApplicationUserID	User ID for the WebSphere MQ Workflow user.

Table 43. Connector-specific properties (continued)

Property name	Property value
Application Password	Password for the WebSphere MQ Workflow user.

Table 44. Supported business objects

Business object name	Agent support required?
Retail_Item	No
MQWF_Retail_Item	Yes
MO_DataHandler_Default	Yes

Note: Be sure to keep the MO_DataHandler_xxx and MO_MQWorkflow_*xxxx* business objects in the list of supported business objects.

2. Save the configuration (**File > Save > To Project**), then go back to the **Associated Map** tab and set the explicit bindings.

Table 45. Associated maps

Business object name	Map name
Retail_Item	Retail_Item_to_MQWF_Retail_Item
MQWF_Retail_Item	MQWF_Retail_Item_to_Retail_Item

3. Save the configuration (**File > Save > To Project**)

4. Create the second instance of the WebSphereMQWorkflowConnector Agent as follows:

Windows operating system:

a. In the Windows task bar, right-click **Start** and select **Open All Users**.

b. Navigate to the folder on your system that contains the installed connectors by clicking **Programs > IBM WebSphere Business Integration Adapters > Adapters > Connectors.**

c. Copy the WebSphereMQWorkflowConnector short cut and rename it to WebSphereMQWorkflow2Connector.

d. Right-click the WebSphereMQWorkflow2Connector short cut and select **Properties**.

e. Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to WebSphereMQWorkflow2, as shown in the following example:

```
<install_path>\IBM\WebSphereICS\connectors\WebSphereMQWorkflow\start_WebSphereMQWorkflow.bat \
WebSphereMQWorkflow2 server_name
```

where *server_name* refers to the name of the WebSphere Interchange Server.

UNIX operating system:

a. Access the WebSphereMQWorkflow2Connector Agent program located in the following directory:
```
<install_path>/IBM/WebSphereICS/connectors/WebSphereMQWorkflow/.
```

b. Run WebSphereMQWorkflow2Connector by switching to this directory and entering the following command:
```
start_WebSphereMQWorkflow.sh WebSphereMQWorkflow2 server_name
```

where *server_name* refers to the name of the WebSphere Interchange Server

5. Create a folder called
   \IBM\WebSphereICS\connectors\WebSphereMQWorkflow2Connector and
   copy CWWebSphereMQWorkflow.jar from the
   WebSphereMQWorkflowConnector folder into this new folder.
6. Create the second instance of the WebSphereMQWorkflowConnector as follows:
   a. Open the System Manager.
   b. Open the WebSphereMQWorkflowConnector.
   c. Click **File > Save As > To Project**.
   d. Save the connector as be WebSphereMQWorkflow2Connector.
7. Modify the properties for WebSphereMQWorkflow2Connector with the values
   shown in the following tables:

Table 46. Standard Properties

Property name	Property value
ApplicationName	MQWorkflow2Connector

Table 47. Connector-Specific Properties

Property name	Property value
ArchiveQueue	MQWFCONN.ARCHIVE2
ErrorQueue	MQWFCONN.ERROR2
InProgressQueue	MQWFCONN.IN_PROGRESS2
InputQueue	CWLDINPUTQ2
OutputQueue	FMC.FMCGRP.EXE.XML2
ReplyToQueue	MQWFCONN.REPLYTO2
UnsubscribedQueue	MQWFCONN.UNSUBSCRIBED2

8. If you are using WebSphere MQ as your connector transport, create the
   WebSphereMQWorkflowConnector queues in WebSphere MQ. You must create
   the following queues as local queues and accept the defaults. Enter the
   following at a command prompt:

```
runmqsc local_queue_manager_name
DEFINE QLOCAL (AP/WEBSPHEREMQWORKFLOWCONNECTOR/server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/server_name/WEBSPHEREMQWORKFLOWCONNECTOR) USAGE (NORMAL)
DEFINE QLOCAL (AP/WEBSPHEREMQWORKFLOW2CONNECTOR/server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/server_name/WEBSPHEREMQWORKFLOW2CONNECTOR) USAGE (NORMAL)
END
```

   where *local_queue_manager_name* is the name of the queue manager used by the
   WebSphere Interchange Server and *server_name* is the interchange server's
   name.
9. Update the file start_WebSphereMQWorkflow.bat with WebSphere MQ and
   WebSphere MQ Workflow Java client library paths.

Refer to the Adapter for WebSphere MQ Workflow User Guide for more
information about this connector.

Configuring the PortConnector

Perform the following steps to configure the PortConnector:

1. Configure this connector to include the appropriate business object. Use the
   value shown in the following table.

Table 48. Supported business objects

Business object name	Agent support required?
Retail_Audit	Yes
Retail_Item	Yes
Retail_Price	Yes

2. If you are using WebSphere MQ as your connector transport, create the PortConnector queues in WebSphere MQ. You must create the following queues as local queues and accept the defaults. Enter the following at a command prompt:

```
runmqsc local_WebSphere_ICS_queue_manager_name
DEFINE QLOCAL (AP/PORTCONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/PORTCONNECTOR) USAGE (NORMAL)
END
```

where *local_queue_manager_name* is the name of the queue manager used by the WebSphere Interchange Server and *server_name* is the interchange server's name.

Creating and configuring the collaboration objects

Use the information provided in this section to bind the ports and set the attribute values of various collaboration objects.

Note: In the tables in this section, some values might include spaces to allow them to fit in the table cells. The actual values do not include spaces.

Creating and configuring a UCCnetMessageReceive collaboration object and making its port connections

To create and configure a collaboration object based on the UCCnetMessageReceive collaboration template, complete the following steps:

1. Name the collaboration object and bind the ports using the values from the following table.

Table 49. Collaboration object name: UMR

Port	Type	Bind to
FromAS2	connector	AS2 channel connector
ToMessage_Store	collaboration object	MS1:From
ToIdentifier_Store	collaboration object	IDS1:From
ToRetail_Processing	collaboration object	IV1:From
ToRetail_Response	collaboration object	UMS1:FromRetail
ToPrice_Processing	User defined, either collaboration object or connector	User Defined, if not needed then bind to PortConnector
Audit	User defined, either collaboration object or connector	User Defined, if not enabled then bind to PortConnector
Error	User defined, either collaboration object or connector	User Defined, if not enabled then bind to PortConnector

2. Set the tracing level for the object.
3. Configure the collaboration object properties using the values from the following table. Use the default property values unless otherwise listed in this table.

Table 50. UMR collaboration object configuration properties

Collaboration property	Value
TORETAIL_PROCESSING_MAP	UCCnetGBO_envelope_to_Retail_Item
TORETAIL_RESPONSE_MAP	UCCnetGBO_envelope_to_Retail_Item
TOMESSAGE_STORE_MAP	UCCnetGBO_envelope_to_UCCnetGBO_storable
TOIDENTIFIER_STORE_MAP	UCCnetGBO_envelope_to_UCCnetGBO_identifier
FILTER_FAIL_RESPONSE	Review
FILTER_DUPLICATE	true (when using the IdentifierStore)
REQUIRED_ATTRIBUTE_FILE	Your fully qualified filename for the required attribute list. Leave blank if there are no attributes to check.
VENDOR_FILE	Your fully qualified filename for the acceptable vendors. Leave blank if all vendors are acceptable.
CATEGORY_FILE	Your fully qualified filename for the acceptable categories. Leave blank if all categories are acceptable.
CATEGORYMAP_FILE	Your fully qualified filename of the UDEX category conversions. Leave blank if there are no UDEX category conversions.
DEBUG	false
COMPLEX_FILTER_FILE	Your fully qualified filename of the complex filtering file. Leave blank if there are no complex filtering rules.
UTILITY_CLASS	com.ibm.wbi.retail.utils.RetailUtility
ENVELOPE_CORRELATIONID_FIELDS	List of fully qualified field names from the UCCnetGBO_envelope business object to be included in the correlationID attribute. The default value is ROOT.messageHeader.messageIdentifier.Value. Do not change the default value, as it affects the key used for the item store and the message store.
NOTIFICATION_CORRELATIONID_FIELDS	List of fully qualified field names from the UCCnetXSD_envelope_notification business object to be included in the correlationID attribute. The default value is sequenceId. Do not change the default value, as it affects the key used for the item store and the message store.
TOPRICE_PROCESSING_MAP	UCCnetGBO_envelope_to_Retail_Price
AUDIT_PORT_ENABLE	false - Unless you want to send audit data out the Audit port. Otherwise, leave the value as false. Setting this to true causes a Map to create a Retail_Audit BO from a price or item notification
ERROR_PORT_ENABLE	false – Unless you want to send Error information out the Error port when certain problems occur within the collaboration
PRICE_AUDIT_MAP	Map used to build the Retail_Audit record sent out the Audit port for price notifications: UCCnet_price_notification_to_Retail_Audit
ITEM_AUDIT_MAP	Map used to build the Retail_Audit record sent out the Audit port for item notifications: UCCnet_item_notification_to_Retail_Audit

Creating and configuring UCCnetMessageSend collaboration objects and making their port connections

To create and configure collaboration objects based on the UCCnetMessageSend collaboration template, complete the following steps to create two instances:

1. Name the collaboration objects and bind the ports using the values from the following table.

Table 51. Collaboration object name: UMS1

Port	Type	Bind to
ToAS2_Response	connector	AS2 channel connector
FromRetail	collaboration object	UMR:ToRetail_Response
ToMessage_Store	collaboration object	MS2:From
ToIdentifier_Store	collaboration object	IDS2:From
Audit	User defined, either collaboration object or connector	User Defined, if not enabled then bind to PortConnector
Error	User defined, either collaboration object or connector	User Defined, if not enabled then bind to PortConnector

Table 52. Collaboration object name: UMS2

Port	Type	Bind to
ToAS2_Response	connector	AS2 channel connector
FromRetail	collaboration object	PRI:respond_to
ToMessage_Store	collaboration object	MS3:From
ToIdentifier_Store	collaboration object	IDS3:From
Audit	User defined, either collaboration object or connector	User Defined, if not enabled then bind to PortConnector
Error	User defined, either collaboration object or connector	User Defined, if not enabled then bind to PortConnector

2. Set the tracing level for each object.
3. Configure the collaboration object properties using the values from the following table. Use the default property values unless otherwise listed in this table. Both collaboration objects use the same property values.

Table 53. UMS1 and UMS2 collaboration object configuration properties

Collaboration property	Value
TOAS2_RESPONSE_MAP	UCCnetGBO_RI_S_to_UCCnetGBO_envelope
SEND_REVIEW	true (Review messages are sent)
SEND_REJECT	true (Rejected messages are sent)
SEND_ACCEPT	true (Accepted messages are sent)
FILTER_DUPLICATE	true (when using the IdentifierStore)
DEBUG	false
AUDIT_REJECTED	false - Unless you want to send audit data out the Audit port for Item Status=Rejected. Otherwise, leave the value as false.
AUDIT_ACCEPTED	false - Unless you want to send audit data out the Audit port for Item Status=Accepted. Otherwise, leave the value as false.

Table 53. UMS1 and UMS2 collaboration object configuration properties (continued)

Collaboration property	Value
AUDIT_SYNCHRONIZED	false - Unless you want to send audit data out the Audit port for Item Status=Approved. Otherwise, leave the value as false.
AUDIT_REVIEW	false - Unless you want to send audit data out the Audit port for Item Status=Review. Otherwise, leave the value as false.
AUDIT_OUTPUT	false - Unless you want to send audit data out the (normal) Audit port. Otherwise, leave the value as false.
ERROR_PORT_ENABLE	false - Unless you want to send error information out the Error port. Otherwise, leave the value as false.

Creating and configuring ItemValidation collaboration objects and making their port connections

To create and configure collaboration objects based on the ItemValidation collaboration template, complete the following steps to create two instances:

1. Name the collaboration objects and bind the ports using the values from the following table.

Table 54. Collaboration object name: IV1

Port	Type	Bind to
From	collaboration object	UMR:ToRetail_Processing
To	collaboration object	ID1:From
Notify	collaboration object	RE1:From
ToMissingData		*Custom* (If none, use PortConnector)
LocalItemStore	collaboration object	IS1:From
DestinationAppRetrieve	connector	PortConnector
Audit	User defined, either collaboration object or connector	User defined, if not enabled, then bind to the PortConnector
Error	User defined, either collaboration object or connector	User defined, if not enabled, then bind to the PortConnector

Table 55. Collaboration object name: IV2

Port	Type	Bind to
Audit	User defined, either collaboration object or connector	User defined, if not enabled, then bind to the PortConnector
DestinationAppRetrieve	connector	PortConnector
Error	User defined, either collaboration object or connector	User defined, if not enabled, then bind to the PortConnector
From	collaboration object	PRI:reprocess
LocalItemStore	collaboration object	IS2:From
Notify	collaboration object	RE2:From
To	collaboration object	ID2:From
ToMissingData		*Custom* (If none, use PortConnector)

2. Set the tracing level for each object.

3. Configure the collaboration object properties using the values from the following table. Use the default property values unless otherwise listed in this table. Both collaboration objects use the same property values.

Table 56. IV1 and IV2 collaboration object configuration properties

Collaboration property	Value
AUDIT_PORT_ON_MISSING_DATA	false - Enable/Disable sending of audit data to the Audit port when sending the item to the ToMissingData port
AUDIT_PORT_ON_APPROVAL	false - Enable/Disable sending of audit data to the Audit port when sending the item to the To port (for items to be approved)
BUSINESS_POLICY_CMDS	Create
CUST_DATA_MISS_ATTR	internals.customer_data_missing_attributes
ERROR_PORT_ENABLE	false - Enable/Disable sending of error information to the Error port
ITEM_COMMAND_ATTRIBUTE	internals.item_command
ITEM_STATUS_ATTRIBUTE	internals.item_status
ITEM_IDENTIFICATION_ATTRIBUTE	item.catalogueItem.tradeItem.tradeItemIdentification.gtin
LOG_REVIEW_ITEM	true
LOG_REJECTED_ITEM	true
LOG_ERROR_ITEM	true
REQUIRED_ATTRIBUTE_CMDS	Create
REQUIRED_ATTRIBUTE_FILE	Your fully qualified attribute file name. Leave blank if there are no required attributes to check.
RETAIL_MISS_ATTR_TYPE	Retail_Missing_Attributes
RETAIL_MISS_ATTR_NAME	attribute_name
RETAIN_ITEM_IN_LOCAL_STORE	true (when ItemStore is used)
MESSAGE_TYPE_PROCESSING_CMDS	Create
UTILITY_CLASS	com.ibm.wbi.retail.utils.RetailUtility
EMAIL_MSG_ATTRIBUTE	internals.message_text
EMAIL_SUBJECT_ATTRIBUTE	internals.message_subject
EMAIL_ROLE_ATTRIBUTE	internals.message_recipient_role
REJECT_EMAIL_MSG	Set the message text for notifying that an item was rejected by business processing. If left blank, the default message as supplied by the collaboration object will be used as the message text.
REJECT_EMAIL_SUBJECT	Retail Item Rejected by ItemValidation
REJECT_EMAIL_ROLE	Your administrator's mail ID
ERROR_EMAIL_MSG	Set the message text for notifying that an error was detected. If left blank, the default message as supplied by the collaboration object will be used as the message text.
ERROR_EMAIL_SUBJECT	ItemValidation Error
ERROR_EMAIL_ROLE	Your administrator's mail ID
SEND_MAIL_ON_ERROR	true
SEND_MAIL_ON_REJECTION	true
TEST	false

Creating and configuring ItemDispatcher collaboration objects and making their port connections

To create and configure collaboration objects based on the ItemDispatcher collaboration template, complete the following steps to create two instances:

1. Name the collaboration objects and bind the ports using the values from the following table.

Table 57. Collaboration object name: ID1

Port	Type	Bind to
From	collaboration object	IV1:To
To	connector	WebSphereMQWorkflowConnector
Notify	collaboration object	RE5:From
LocalItemStore	collaboration object	IS5:From
Audit	User defined, either collaboration object or connector	User Defined, if not enabled, then bind to PortConnector

Table 58. Collaboration object name: ID2

Port	Type	Bind to
Audit	User defined, either collaboration object or connector	User Defined, if not enabled, then bind to PortConnector
From	collaboration object	IV2:To
To	connector	WebSphereMQWorkflowConnector
Notify	collaboration object	RE6:From
LocalItemStore	collaboration object	IS6:From

2. Set the tracing level for each object.
3. Configure the collaboration object properties using the values from the following table. Use the default property values unless otherwise listed in this table. Both collaboration objects use the same property values.

Table 59. ID1 and ID2 collaboration object configuration properties

Collaboration property	Value
UTILITY_CLASS	com.ibm.wbi.retail.utils.RetailUtility
GLN_CASCADE_GROUPING_FILE	Your fully qualified name of the GLN Cascade Grouping file. Leave blank if there are no cascaded groups.
GLN_CASCADE_GROUPING_DEFAULT	Possible values are: • Error - Error condition if no grouping file found); • Split - Split each GLN into a separate message • Together- Send all GLNs together in one group
EMAIL_MSG_ATTRIBUTE	internals.message_text
EMAIL_SUBJECT_ATTRIBUTE	internals.message_subject
EMAIL_ROLE_ATTRIBUTE	internals.message_recipient_role
ERROR_EMAIL_ROLE	Your administrator's mail ID
ERROR_EMAIL_MSG	Set the message text for notifying that an error was detected. If left blank, the default message as supplied by the collaboration object is used as the message text.
ERROR_EMAIL_SUBJECT	ItemDispatcher error
DEBUG	false

Table 59. ID1 and ID2 collaboration object configuration properties (continued)

Collaboration property	Value
SEND_MAIL_ON_ERROR	true
AUDIT_PORT_ENABLE	false - Unless you want to send audit data out the Audit port. Otherwise, leave the value as false

Creating and configuring an ItemCollector collaboration object and making its port connections

To create and configure a collaboration object based on the ItemCollector collaboration template, complete the following steps:

1. Name the collaboration object and bind the ports using the values from the following table.

Table 60. Collaboration object name: IC

Port	Type	Bind to
From	connector	WebSphereMQWorkflow2Connector
To	collaboration object	PRI:From
local_store	collaboration object	IS3:From
email	collaboration object	RE3:From
Audit	User defined, either collaboration object or connector	User Defined, if not enabled then bind to PortConnector

2. Set the tracing level for the object.
3. Configure the collaboration object properties using the values from the following table. Use the default property values unless otherwise listed in this table.

Table 61. IC collaboration object configuration properties

Collaboration property	Value
AUDIT_PORT_ENABLE	false - Unless you want to send audit data out the Audit port. Otherwise, leave the value as false.
LOG_ERROR_ITEM	true
ITEM_IDENTIFICATION_ATTRIBUTE	item.catalogueItem.tradeItem.tradeItemIdentification.gtin
ITEM_STATUS_ATTRIBUTE	internals.item_status
MISSING_DATA_CHILD_ATTRIBUTE	internals.customer_data_missing_attributes
MISSING_DATA_NAME_ATTRIBUTE	attribute_name
MISSING_DATA_VALUE_ATTRIBUTE	attributeValue
1_COPY_ATTRIBUTE	internals.item_status
2_COPY_ATTRIBUTE	internals.date_processed
3_COPY_ATTRIBUTE	internals.time_processed
4_COPY_ATTRIBUTE	internals.responder_name
EMAIL_MSG_ATTRIBUTE	internals.message_text
EMAIL_SUBJECT_ATTRIBUTE	internals.message_subject
EMAIL_ROLE_ATTRIBUTE	internals.message_recipient_role

Table 61. IC collaboration object configuration properties (continued)

Collaboration property	Value
ERROR_RETRIEVE_EMAIL_MSG	Set the message text for notifying that an error was detected when retrieving an item from the local item store. If left blank, the default message as supplied by the collaboration object will be used as the message text.
ERROR_SEND_EMAIL_MSG	Set the message text for notifying that an error was detected while sending the merged item to the To port. If left blank, the default message as supplied by the collaboration object will be used as the message text.
ERROR_EMAIL_SUBJECT	Internal error occurred
ERROR_EMAIL_ROLE	Your administrator's mail ID
SEND_MAIL_ON_ERROR	true

Creating and configuring IdentifierStore collaboration objects and making their port connections

To create and configure collaboration objects based on the IdentifierStore collaboration template, complete the following steps to create *three* instances:

1. Name the collaboration objects and bind the ports using the values from the following table.

Table 62. Collaboration object name: IDS1

Port	Type	Bind to
From	collaboration object	UMR:ToIdentifier_Store
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

Table 63. Collaboration object name: IDS2

Port	Type	Bind to
From	collaboration object	UMS1:ToIdentifier_Store
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

Table 64. Collaboration object name: IDS3

Port	Type	Bind to
From	collaboration object	UMS2:ToIdentifier_Store
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

2. Set the tracing level for each object.
3. Configure the collaboration object properties using the values from the following table. Use the default property values unless otherwise listed in this table. All three collaboration objects use the same property values.

Table 65. IDS1, IDS2, and IDS3 collaboration object configuration properties

Collaboration property	Value
OBJECT_KEY	• When XSD XML definition type is used: gtin, topic, dataRecipientGLN, dataSourceGLN, targetMarket, uniqueCreatorID • When DTD XML definition type is used: gtin, version, topic
GENERATE_KEY	false
MIME_TYPE	text/xml.datastore
TEST	false

Creating and configuring MessageStore collaboration objects and making their port connections

To create and configure collaboration objects based on the MessageStore collaboration template, complete the following steps to create three instances:

1. Name the collaboration objects and bind the ports using the values from the following tables.

Table 66. Collaboration object name: MS1

Port	Type	Bind to
From	collaboration object	UMR:ToMessage_Store
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

Table 67. Collaboration object name: MS2

Port	Type	Bind to
From	collaboration object	UMS1:ToMessage_Store
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

Table 68. Collaboration object name: MS3

Port	Type	Bind to
From	collaboration object	UMS2:ToMessage_Store
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

2. Set the tracing level for each object.
3. Configure the collaboration object properties using the values from the following table. Use the default property values unless otherwise listed in this table. All these collaboration objects use the same property values.

Table 69. MS1, MS2, MS3 collaboration object configuration properties

Collaboration property	Value
OBJECT_KEY	correlationID
GENERATE_KEY	false
MIME_TYPE	text/xml.datastore

Table 69. MS1, MS2, MS3 collaboration object configuration properties (continued)

Collaboration property	Value
TEST	false

Creating and configuring ItemStore collaboration objects and making their port connections

To create and configure collaboration objects based on the ItemStore collaboration template, complete the following steps to create *six* instances:

1. Name the collaboration objects and bind the ports using the values from the following table.

Table 70. Collaboration object name: IS1

Port	Type	Bind to
From	collaboration object	IV1:LocalItemStore
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

Table 71. Collaboration object name: IS2

Port	Type	Bind to
From	collaboration object	IV2:LocalItemStore
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

Table 72. Collaboration object name: IS3

Port	Type	Bind to
From	collaboration object	IC:local_store
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

Table 73. Collaboration object name: IS4

Port	Type	Bind to
From	collaboration object	PRI:local_store
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

Table 74. Collaboration object name: IS5

Port	Type	Bind to
From	collaboration object	ID1:LocalItemStore
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

Table 75. Collaboration object name: IS6

Port	Type	Bind to
From	collaboration object	ID2:LocalItemStore

Table 75. Collaboration object name: IS6 (continued)

Port	Type	Bind to
To	connector	JDBCConnector
DestinationAppRetrieve	connector	JDBCConnector

2. Set the tracing level for each object.
3. Configure the collaboration object properties using the values from the following table. Use the default property values unless otherwise listed in this table. All these collaboration objects use the same property values.

Table 76. IS1, IS2, IS3, IS4, IS5 and IS6 collaboration object configuration properties

Collaboration property	Value
OBJECT_KEY	internals.correlationID
GENERATE_KEY	false
MIME_TYPE	text/xml.datastore
TEST	false

Creating and configuring a Process_Reviewed_Item collaboration object and making its port connections

To create and configure a collaboration object based on the Process_Reviewed_Item collaboration template, complete the following steps:

1. Name the collaboration object and bind the ports using the values from the following table.

Table 77. Collaboration object name: PRI

Port	Type	Bind to
From	collaboration object	IC:To
Sync	connector	JTextConnector
mail	collaboration object	RE4:From
respond_to	collaboration object	UMS2:FromRetail
reprocess	collaboration object	IV2:From
local_store	collaboration object	IS4:From
Audit	User defined, either collaboration object or connector	User Defined, if not enabled then bind to PortConnector

2. Set the tracing level for the object.
3. Configure the collaboration object properties using the values from the following table. Use the default property values unless otherwise listed in this table.

Table 78. PRI collaboration object configuration properties

Collaboration property	Value
AUDIT_PORT_ENABLE	false - Unless you want to send audit data out the Audit port. Otherwise, leave the value as false.
ITEM_STATUS_ATTRIBUTE	internals.item_status
ITEM_IDENTIFICATION_ATTRIBUTE	item.catalogueItem.tradeItem.tradeItemIdentification.gtin
EMAIL_MSG_ATTRIBUTE	internals.message_text
EMAIL_SUBJECT_ATTRIBUTE	internals.message_subject

Table 78. PRI collaboration object configuration properties (continued)

Collaboration property	Value
EMAIL_ROLE_ATTRIBUTE	internals.message_recipient_role
REJECT_EMAIL_MSG	Set the message text for notifying that an item has a status of Rejected. If left blank, the default message as supplied by the collaboration object will be used as the message text.
REJECT_EMAIL_SUBJECT	item rejected
REJECT_EMAIL_ROLE	Your administrator's mail ID
ERROR_EMAIL_MSG	Set the message text for notifying that an error was detected. If left blank, the default message as supplied by the collaboration object will be used as the message text.
ERROR_EMAIL_SUBJECT	internal error occurred
ERROR_EMAIL_ROLE	Your administrator's mail ID
ACCEPTED_EMAIL_MSG	Set the message text for notifying that an item has a status of Accepted. If left blank, the default message as supplied by the collaboration object will be used as the message text.
ACCEPTED_EMAIL_SUBJECT	item accepted
ACCEPTED_EMAIL_ROLE	Your administrator's mail ID
APPROVED_EMAIL_MSG	Set the message text for notifying that an item has a status of Approved. If left blank, the default message as supplied by the collaboration object will be used as the message text.
APPROVED_EMAIL_SUBJECT	item approved
APPROVED_EMAIL_ROLE	Your administrator's mail ID
SEND_MAIL_ON_ERROR	true
SEND_MAIL_ON_REJECTION	true
SEND_MAIL_ON_ACCEPTED	true
SEND_MAIL_ON_APPROVAL	true
LOG_ACCEPTED_ITEM	true
LOG_APPROVED_ITEM	true
LOG_REJECTED_ITEM	true
LOG_ERROR_ITEM	true
DELETE_FROM_LOCAL_STORE	true
RETAIN_ITEM_IN_LOCAL_STORE	false
QUALIFICATION_FAILED_EMAIL_MSG	Set the message text for notifying that the status of the Retail_Item returned from the validating collaboration object is Rejected. If left blank, the default message as supplied by the collaboration object will be used as the message text.
SYNC_TO_BACKEND	true

Creating and configuring UCCnet_requestWorklist collaboration object and making its port connections

To create and configure collaboration object based on the UCCnet_requestWorklist collaboration template, complete the following steps:

1. Name the collaboration objects and bind the ports using the values from the following table.

Table 79. Collaboration object name: URW

Port	Type	Bind to
From	Connector	JTextRWLConnector
To	Connector	AS2 channel connector

2. Set the tracing level for the object.
3. Configure the collaboration object properties using the values from the following table.

Table 80. URW collaboration object configuration properties

Property name	Value
SET_UNIQUE_IDS	Controls whether unique IDs (messageIdentifier and uniqueCreateorIdentification) are set in the output XML messages. Possible values for this property are:
	ALL Set all three unique IDs. The default value.
	NONE No unique IDs
	BLANK Set unique ID only if it is blank in the input

Creating and configuring Role_Email collaboration objects and making their port connections

To create and configure collaboration objects based on the Role_Email collaboration template, complete the following steps to create six instances:

1. Name the collaboration objects and bind the ports using the values from the following tables.

Table 81. Collaboration object name: RE1

Port	Type	Bind to
From	collaboration object	IV1:Notify

Table 82. Collaboration object name: RE2

Port	Type	Bind to
From	collaboration object	IV2:Notify

Table 83. Collaboration object name: RE3

Port	Type	Bind to
From	collaboration object	IC:email

Table 84. Collaboration object name: RE4

Port	Type	Bind to
From	collaboration object	PRI:mail

Table 85. Collaboration object name: RE5

Port	Type	Bind to
From	collaboration object	ID1:Notify

Table 86. Collaboration object name: RE6

Port	Type	Bind to
From	collaboration object	ID2:Notify

2. Set the tracing level for each object.

3. Configure the collaboration object properties using the values from the following table. Use the default property values unless otherwise listed in this table. All six collaboration objects use the same property values.

Table 87. RE1, RE2, RE3, RE4, RE5 and RE6 collaboration object configuration properties

Collaboration property	Value
MSG_RECIPIENT_ATTRIBUTE	internals.message_recipient_role
MSG_TEXT_ATTRIBUTE	internals.message_text
MSG_SUBJECT_ATTRIBUTE	internals.message_subject
LOG_ERROR	true
LOG_ALL_MAIL	false
SUBSTITUTION_VARIABLE_PREFIX	${ (These characters might have to be changed to meet National Language requirements.)
SUBSTITUTION_VARIABLE_SUFFIX	} (This character might have to be changed to meet National Language requirements.)
FILE_NAME_PREFIX	@ (This character might have to be changed to meet National Language requirements.)

Deploying the solution

Deploy the solution, as follows:

1. After all of the components of the solution have been configured, compile the maps and collaboration templates in the System Manager.

2. Create a User Project that contains all of the solution components.

3. Use the Deploy wizard to deploy the solution to the WebSphere Interchange Server (running in design mode). The wizard allows you to choose what parts of the solution to deploy. It is recommended that you deploy the solution in stages as follows:

   a. Deploy the business objects.

   b. Deploy the maps. Ensure that you have checked the **Compile** option.

      Note: Ensure that the system PATH statement contains an entry for the jdk131 installed products \bin directory for the WebSphere Interchange Server to find the Java compiler.

   c. Deploy the relationships. Ensure that you have checked the **Create schema** option.

      Note: If you are using the Microsoft Visual C++ v6.0 compiler, be sure that you have updated the database software as needed. For example, IBM DB2 requires the following db2set command:

      ```
      DB2SET DB2_SQLROUTINE_COMPILER_PATH=
              "C:\Program Files\Microsoft Visual Studio\V98\Bin\VCVARS32.BAT"
      ```

   d. Deploy the connectors.

   e. Deploy the collaboration templates. Ensure that you have checked the **Compile** option.

   f. Restart the WebSphere InterChange Server.

   g. From the Component Management window of the System Manager, start the relationships.

   h. Deploy the collaboration objects.

 4. Restart the ICS.

Configuring the relationship

Edit the CMDTOWPN relationship using the System Manager, as follows:

1. Open the System Manager.
2. Double-click the CMDTOWPN Static Relationship to open it in the Relationship Designer.
3. Click **Edit > Advanced Settings** and set the following DBMS Settings:
   - URL = URL for the database
   - Login = Database user's name
   - Password = Database user's password
   - Type = Oracle, SQL Server, or DB2
4. Click **OK** to save the settings.
5. In the Relationship Designer, select **Tools > Relationship Manager**.
6. In the Relationship Manager, connect to the ICS, as follows:
   a. Select **Server > Connect to Server**.
   b. Specify the **Relationship** as CMDTOWPN.
   c. Click **Connect** to connect to the ICS.
7. Click **Get Relationships**. **Note:** CMDTOWPN must be deployed and active.
8. To add the Retail_ItemDelete relationship, Click the **New Relationship** icon in the Relationship Manager tool bar. Then do the following steps:
   a. Expand the newly added row by clicking the plus sign (+).
   b. Right click **WFPGNAME** and select **Add Participant**.
   c. Click the plus sign (+) next to **WFPGNAME** and enter `Retail_ItemDelete` in the **Value** column of the newly added row.
   d. Right click **COMMAND** and select **Add Participant**.
   e. Click the plus sign (+) next to **COMMAND** and enter `DELETE` in the **Value** column of the newly added row.
   f. Right click **COMMAND** and select **Add Participant** again.
   g. Enter `WITHDRAW` in the **Value** column of the newly added row.
   h. Add the Retail_ItemCreate and Retail_ItemUpdate WFPGNames and their commands listed in the table below in a similar fashion by following the same steps used for the Retail_ItemDelete relationship..

Table 88. Relationships

WFPGName	Command
Retail_ItemDelete	DELETE WITHDRAW
Retail_ItemCreate	CREATE LOAD
Retail_ItemUpdate	UPDATE

Configuring WebSphere MQ Workflow and WebSphere MQ

To configure WebSphere MQ Workflow and WebSphere MQ, complete the following steps:

1. You might have to modify the Workflow process definitions to match the queue manager specified in the network information to the IBM WebSphere ICS queue manager. Use the WebSphere MQ Workflow Buildtime user interface:

    a. Import the Workflow process definitions .fdl file into the Workflow Buildtime by doing the following:

        1) Open Workflow Buildtime.
        2) From the menu bar, select **Buildtime**.
        3) Click **Import**.
        4) From the Buildtime menu, locate your .fdl file (that is, `Retail.fdl`), select **Overwrite**, and click **OK**.

    b. Update the network information by doing the following:

        1) In the left pane of the Workflow Buildtime window, click the **Network** tab.
        2) Expand the DOMAIN, FMCGRP, and FMCSYS sections.
        3) Right-click **CWLDSVR** and click **Properties**.
        4) In the dialog box, click the **Message Queuing** tab, change the queue name to CWLDINPUTQ2 and change the queue manager to the appropriate value (that is, to your local IBM WebSphere MQ Workflow queue manager name).
        5) Click **OK**.

    c. Update the process by doing the following:

        1) Click the **Processes** tab.
        2) Expand the **WSBI-Retail** node.
        3) Double-click **Retail_ItemCreate**.
        4) Right-click the **Item_Approve** process and click **Properties**.
        5) Click the **Start** tab and click **Manual**, then close the Properties window.
        6) Right-click the **Approval_Reply** process and click **Properties**.
        7) Click the **Start** tab and click **Manual**, then close the Properties window.
        8) Double-click **Retail_ItemUpdate**.
        9) Right-click the **Item_Approve** process and click **Properties**.
        10) Click the **Start** tab and click **Manual**, then close the Properties window.
        11) Right-click the **Approval_Reply** process and click **Properties**.
        12) Click the **Start** tab and click **Manual**, then close the Properties window.
        13) If you want to use an ID other than ADMIN for logging into the WebSphere MQ Workflow Client, add the following WorkFlow process definition user IDs to the system and make them members of role CategoryManager:
            - CTGMGR_1
            - CTGMGR_2
            - CTGMGR_3
        14) Select **Buildtime** from the menu and click **Export**.
        15) Select **Export deep** in the **Export flags** section.
        16) Click **OK** and save the .fdl file when prompted.

2. Import the WebSphere MQ Workflow process definitions using the .fdl file just exported. Enter the following command:

```
fmcibie -i<fdl_filename> -uadmin -ppassword -o -f -t
```

3. Define the WebSphere MQ channels required on the WebSphere MQ Workflow queue manager to communicate with the IBM WebSphere ICS queue manager. Enter the following at a command prompt. (The commands shown in this step assume that the IBM WebSphere ICS queue manager's name is *local_WebSphere_ICS_queue_manager_name*, and the WebSphere MQ Workflow queue manager's name is *FMCQM*. Enter the appropriate names for your queue managers.)

```
runmqsc FMCQM
DEFINE QLOCAL ('local_WebSphere_ICS_queue_manager_name') USAGE (XMITQ)
DEFINE CHANNEL ('FMCQM.TO.CW') CHLTYPE (SDR) CONNAME
    ('local_hostname(Listener port for local_WebSphere_ICS_queue_manager)')
    XMITQ ('local_WebSphere_ICS_queue_manager_name')
DEFINE CHANNEL ('CW.TO.FMCQM') CHLTYPE (RCVR)

END
```

4. Define the WebSphere MQ channels required on the IBM WebSphere ICS queue manager to communicate with the WebSphere MQ Workflow queue manager. Enter the following at a command prompt. (The commands shown in this step assume that the IBM WebSphere ICS queue manager's name is *local_WebSphere_ICS_queue_manager_name*, and the WebSphere MQ Workflow queue manager's name is *FMCQM*. Enter the appropriate names for your queue managers.)

```
runmqsc local_WebSphere_ICS_queue_manager_name
DEFINE QLOCAL (FMCQM) USAGE (XMITQ)
DEFINE CHANNEL ('CW.TO.FMCQM') CHLTYPE (SDR) CONNAME
    ('local_hostname(Listener port for WebSphere_MQ_Workflow_queue_manager)')
    XMITQ (FMCQM)
DEFINE CHANNEL ('FMCQM.TO.CW') CHLTYPE (RCVR)
END
```

5. Define the following local queues required by WebSphere MQ Workflow queue manager. Enter the following at a command prompt (these commands assume that the WebSphere MQ Workflow queue manager is named *FMCQM*):

```
runmqsc FMCQM
DEFINE QLOCAL (MQWFCONN.ARCHIVE) USAGE (NORMAL)
DEFINE QLOCAL (MQWFCONN.ERROR) USAGE (NORMAL)
DEFINE QLOCAL (MQWFCONN.IN_PROGRESS) USAGE (NORMAL)
DEFINE QLOCAL (MQWFCONN.REPLYTO) USAGE (NORMAL)
DEFINE QLOCAL (MQWFCONN.UNSUBSCRIBED) USAGE (NORMAL)
DEFINE QLOCAL (CWLDINPUTQ) USAGE (NORMAL)
DEFINE QLOCAL (CWLDRETQ) USAGE (NORMAL)
DEFINE QLOCAL (MQWFCONN.ARCHIVE2) USAGE (NORMAL)
DEFINE QLOCAL (MQWFCONN.ERROR2) USAGE (NORMAL)
DEFINE QLOCAL (MQWFCONN.IN_PROGRESS2) USAGE (NORMAL)
DEFINE QLOCAL (MQWFCONN.REPLYTO2) USAGE (NORMAL)
DEFINE QLOCAL (MQWFCONN.UNSUBSCRIBED2) USAGE (NORMAL)
DEFINE QLOCAL (CWLDINPUTQ2) USAGE (NORMAL)
DEFINE QLOCAL (CWLDRETQ2) USAGE (NORMAL)

END
```

6. Verify that the following WebSphere MQ queue required by the WebSphere MQ Workflow queue manager is defined in the FMCQM queue manager.

   FMC.FMCGRP.EXE.XML

   If it is not defined, you can define by entering the following at a command prompt:

```
runmqsc FMCQM
DEFINE QALIAS(FMC.FMCGRP.EXE.XML) DEFPSIST(YES) TARGQ(EXEXMLINPUTQ) CLUSTER(FMCGRP) DEFBIND(NOTFIXED)
END
```

7. Define a similar queue for the second instance of the WebSphereMQWorkflowConnector by entering the following at a command prompt:

```
runmqsc FMCQM
DEFINE QALIAS(FMC.FMCGRP.EXE.XML2) DEFPSIST(YES) TARGQ(EXEXMLINPUTQ2) CLUSTER(FMCGRP) DEFBIND(NOTFIXED)
END
```

8. If you are running WebSphere MQ Workflow and the IBM WebSphere ICS on the same machine, make sure that the port number used by each instance of the WebSphere MQ queue manager is unique (default for WebSphere MQ queue manager [first instance] is 1414). The default port number for WebSphere MQ Workflow is 5010).

Notices and Trademarks

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

Notices

IBM may not offer the products, services, or features discussed in this document in all 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.

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 program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created

programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Burlingame Laboratory Director
IBM Burlingame Laboratory
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not necessarily tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

COPYRIGHT LICENSE
This information may contain 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.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

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

IBM
the IBM logo
AIX
DB2
DB2 Universal Database
Passport Advantage
WebSphere

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

Microsoft and Windows, are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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