

*IBM WebSphere Business Integration Collaborations
for Consumer Products
Item Synchronization, Version 44.0*



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 39.

Eighth Edition (July 2004)

This edition applies to Version 4, Release 4, *IBM WebSphere Business Integration Collaborations for Consumer Products Item Synchronization, Version 4.4.0* (5724-H62)

IBM welcomes your comments. You can send them to the following address:

IBM Corporation
Department 6R4A
11501 Burnet Road
Austin, TX 78578

Include the title of this book, and the page number or topic related to your comment.

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

© Copyright International Business Machines Corporation 2002, 2004. All rights reserved.

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

Contents

Installation guide 1

Obtaining solution components	1
Solution components	1
Supported operating environments	2
Hardware prerequisites	3
Software prerequisites	3
Installing and configuring the solution.	4
Planning the configuration.	4
Configuring the system environment	5
Installing the components	6
Creating the ItemSync database and tables	7
Configuring the metaobjects	8

Creating the database connection pool	11
Creating and configuring the connectors.	12
Creating and configuring the collaboration objects	31
Deploying the solution	38

Notices and Trademarks 39

Notices	39
Programming interface information	40
Trademarks and service marks	41

Installation guide

This document describes how to install and configure components of the Item Synchronization for Suppliers solution. In order to implement the solution, you need to refer to and be familiar with the following core documents:

- 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

Deployment of the Item Synchronization for Suppliers solution involves a series of product installations and configurations. Several elements are common to all engagements; other elements depend on your individual topology, set of installed enterprise applications, and on the marketplaces in which you participate.

Obtaining solution components

Before following the installation procedure, obtain the IBM WebSphere Business Integration Collaboration for Consumer Products Item Synchronization by downloading it from Passport Advantage® (www.lotus.com/passportadvantage). Refer to Passport Advantage for downloading instructions.

Solution components

The following tables list the components used to develop the Item Synchronization for Suppliers solution. Note that there are many other existing components that can be utilized in your customized solution.

Repository files

The following repository files are included:

BIA_Retail_BO_ItemSync.jar
BIA_Retail_BO_ItemSyncSamples.jar
BIA_Retail_CT_ItemSync.jar
BIA_Retail_NM_ItemSync.jar
BIA_Retail_NM_ItemSyncSamples.jar

Collaboration templates

The following collaborations, in the BIA_Retail_CT_ItemSync.jar repository file, are included in the solution:

UCCnet_ItemSync
UCCnet_requestWorklist
UCCnet_processWorklist
Notify_by_eMail
CommandRouter
CI_Sync
CIN_CIP_Dispatcher
UCCnet_Price

Business objects

The following business objects are included in the solution.

Business objects in the BIA_Retail_BO_ItemSync.jar repository file:

UCCnetGBO_envelope
UCCnetXSD_envelope
UCCnetTPIXSD_envelope
UCCnetJMSXSD_envelope
Retail_Item
Retail_Price

Business objects in the BIA_Retail_BO_ItemSyncSamples.jar repository file:

Retail_Item_ASBO
Retail_Price_ASBO

Maps

The following maps are included in the solution.

Maps located in the BIA_Retail_NM_ItemSyncSamples.jar:

Retail_Item_ASBO_to_Retail_Item
Retail_Price_ASBO_to_Retail_Price

Maps located in the BIA_Retail_NM_ItemSync.jar:

Retail_Item_to_UCCnetXSD_envelope_publicationCommand_catalogueItem
Retail_Item_to_UCCnetJMSXSD_envelope_publicationCommand_catalogueItem
Retail_Item_to_UCCnetTPIXSD_envelope_publicationCommand_catalogueItem
Retail_Item_to_UCCnetGBO_env_publicationCommand_CIP
Retail_Item_to_UCCnetGBO_envelope_notifyCommand_catalogueItem
Retail_Item_to_UCCnetXSD_envelope_registerCommand_itemAddChange
Retail_Item_to_UCCnetJMSXSD_envelope_registerCommand_itemAddChange
Retail_Item_to_UCCnetTPIXSD_envelope_registerCommand_itemAddChange
Retail_Price_to_UCCnetJMSXSD_envelope
Retail_Price_to_UCCnetTPIXSD_envelope
Retail_Price_to_UCCnetXSD_envelope

Data base scripts

The following data base scripts are included for DB2

- Create_ItemSync_database.sql
- Create_ItemSync_tables.sql

Microsoft® SQL IServer and Oracle use the Create_ItemSync_tables.sql script to create the tables. However, you must create the database.

Library file for collaboration templates

The CatalogueItemUtility.jar library file for collaboration templates is included.

Supported operating environments

The following operating environments are supported:

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

Hardware prerequisites

The Item Synchronization for Suppliers 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. To download, expand, and install this solution, you require 56.2 MB of free space. After the installation, you can remove the downloaded installation files and leave only the solution on your machine. The installed solution takes up 6.6 MB of space.

Software prerequisites

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

- IBM WebSphere InterChange Server V4.2.2.

Note:

The IBM WebSphere Interchange Server installer includes Retail collaborations, but the version included might not be the latest available. The latest version of the Item Synchronization for Suppliers collaboration can be downloaded from Passport Advantage (www.lotus.com/passportadvantage). See “Obtaining solution components” on page 1 for more information. If you use the version provided by the IBM WebSphere Interchange Server installer, make sure that the version of documentation you use is correct for the version of the Item Synchronization for Suppliers collaboration.

Additionally, you must not install the IBM WebSphere Business Integration Collaboration for Product Information Management or the IBM WebSphere Business Integration Collaboration for UCCnet Message Manager.

- IBM WebSphere Business Integration Toolset, version 4.2.2.2
- IBM DB2 Universal Database™ Enterprise Server Edition, version 8.1.2 (Fix Pack 2) (Oracle and Microsoft SQL Server data bases are also supported)
- IBM WebSphere MQ, version 5.3.0.2 with CSD 7
- IBM WebSphere Business Integration Data Handler for XML, version 2.5.0
- IBM WebSphere Business Integration Adapters, version 2.5, which include the following:
 - IBM WebSphere Business Integration Adapter for e-Mail, version 5.3
 - IBM WebSphere Business Integration Adapter for Java™ Message Service (JMS), version 2.6
 - IBM WebSphere Business Integration Adapter for JText, version 5.5
 - IBM WebSphere Business Integration Adapter for Trading Partner Interchange, version 3.4.0
 - IBM WebSphere Business Integration Adapter for iSoft, version 1.4
 - Any specific ERP to be used.

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 Business Integration Data Handler for XML, the

database, and individual IBM WebSphere Business Integration Adapters, are detailed in their respective installation documents.

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).

Installing and configuring the solution

This installation guide provides step-by-step instructions for installing and configuring the Item Synchronization for Suppliers solution. Install and configure the solution in the order listed below:

1. “Planning the configuration” details the connectivity and messaging decisions.
2. “Configuring the system environment” on page 5 details setting up the system environment to run the Item Synchronization for Suppliers on all supported platforms.
3. “Installing the components” on page 6 instructs how to load artifacts and install them in the InterChange Server repository, ensuring that the solution code is accessible to your system.
4. “Creating the ItemSync database and tables” on page 7 describes how to create the relational tables required by the solution.
5. “Configuring the metaobjects” on page 8 instructs how to create and configure the business objects required to properly process UCCnet messages.
6. “Creating the database connection pool” on page 11 describes how to create the connection to the Item Sync database.
7. “Creating and configuring the connectors” on page 12 describes how to create and configure the connectors.

Note: The term *connector* used throughout 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.

8. “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 the port connections and configuration properties for each collaboration object.
9. “Deploying the solution” on page 38 details how to deploy the solution.

Planning the configuration

Before you install and configure the Item Synchronization for Suppliers 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 TPICConnector, 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.

Messages

Messages are exchanged with UCCnet in Extensible Markup Language (XML) documents. The XML document format and the protocol that you select for communication with UCCnet significantly impact the way that you set up your solution. The format of the XML documents exchanged with UCCnet is defined by an XML Schema Definition (XSD). The XSD mode of operation supports the following command protocols:

CIN operation

The supplier implements its own subscriber data pool. Catalogue_Item_Notification (CIN) messages are sent from the supplier directly to trading partners subscribed to the product categories.

CIP operation

The supplier uses UCCnet as the subscriber data pool. Catalogue_Item (CI) messages containing additional item information that is not included in the UCCnet registry data are sent from the supplier to UCCnet. Catalogue_Item_Publication messages are then sent to UCCnet to identify the subscribers to whom UCCnet needs to send CIN messages.

Configuring the system environment

To configure the system environment, complete the following steps. Be sure to follow the instructions appropriate for your platform where indicated.

1. Edit the CWCLASSES path, as follows:

Windows operating systems:

Edit the `<WebSphereICS_installation_dir>\bin\start_server.bat` file by appending the end of the CWCLASSES path to include the CatalogueItemUtility.jar file, as shown in the following example:

```
CWCLASSES=...;%CROSSWORLDS%\lib\CatalogueItemUtility.jar
```

UNIX operating systems:

Edit the `<WebSphereICS_installation_dir>/bin/CWSharedEnv.sh` file and alter the CWCLASSES path to include the CatalogueItemUtility.jar file, as shown in the following example:

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

Note: The CatalogueItemUtility.jar file must be copied from the Windows system to the UNIX system.

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 CatalogueItemUtility.jar file and its appropriate path information to the line `classpath=` in this file, as in the following example:

```
classpath=<directory_location_of_CatalogueItemUtility.jar_file>  
-file>CatalogueItemUtility.jar
```

3. Configure the InterChange Server with e-mail information by doing the following:
 - a. Open the System Manager.
 - b. Connect to the InterChange Server.
 - c. Right-click on the InterChange Server 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 InterChange Server.

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. Start the InterChange Server in design mode (-design parameter).
2. Install the artifacts downloaded from www.lotus.com/passportadvantage into the same directory in which the InterChange Server is installed by performing the following steps (see "Obtaining solution components" on page 1 for information on how to obtain the artifacts from Passport Advantage):
 - a. Move the downloaded .exe file to a temporary directory.
 - b. Run the executable file. The following files are extracted:
 - 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 executable file to install the components.
3. Verify that the CatalogueItemUtility.jar file is in the Windows directory specified in the classpath on the machine where the System Manager is installed. This ensures that the templates can be compiled successfully in the System Manager before deployment to the InterChange Server.
4. On UNIX systems, transfer the file `<WebSphereICS_installation_dir>\lib\CatalogueItemUtility.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.
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. Browse to the repository .jar files and select one of the files.
 - d. Click Open.
 - e. Click Finish.
 - f. Repeat this process for each of the remaining repository files.

Creating the ItemSync database and tables

Do the following steps to create the ItemSync database and PROCESSED_GTIN and AUDIT_LOG tables.

Be sure to follow the instructions appropriate for your platform.

Creating the data base on a Windows system

Do the following steps, as appropriate for your data base system.

DB2: The script you run creates a database called ITEMSYNC. You can edit the script to change the name of the database. The database name must match the name specified in the database connection pool created later in this installation.

The script also contains a drive and path to a DB2CLI.LST file which must be edited to match your DB2® installation.

1. Start a DB2 session by opening a DB2 command window. Click **Start > Programs > IBMDB2 > Command Line Tools > Command Window**.
2. Change to the directory:
`<ICS_installation_path>\collaborations\dependencies\db2`
3. To run the script, enter the command
`db2 < Create_ItemSync_Database.sql`

You can ignore the error messages from the DROP commands in the script.

4. To connect to the ITEMSYNC database, enter the command:
`db2 < connect to ITEMSYNC`
5. Enter the command
`db2 < Create_ItemSync_Tables.sql`

to run the script to create the tables.

Oracle:

1. Create a database named ITEMSYNC.
2. Enter the command:
`sqlplus dbuser/passwd@SID @Create_ItemSync_Tables.sql`

to run the script to create the tables.

Microsoft SQL Server:

1. Create a database named ITEMSYNC.
2. Use the Query Analyzer tool to run the Create_ItemSync_Tables.sql script to create the tables.

Creating the data base on a UNIX system

Do the following steps, as appropriate for your data base system.

DB2:

1. Transfer the files

```
<ICS_installation_path>/collaborations/dependencies/UCCnet/db2/  
Create_ItemSync_Database.sql  
<ICS_installation_path>/collaborations/dependencies/UCCnet/db2/  
Create_ItemSync_tables.sql
```

to the UNIX system using FTP in ASCII mode. Place the file in the
<ICS_installation_path>/collaborations/dependencies/UCCnet/db2 directory.

2. At the UNIX system, cd to the directory:
<ICS_installation_path>/collaborations/dependencies/UCCnet/db2
3. Start the DB2 session. Enter the command
db2
4. When the session has started, enter the command
< Create_ItemSync_Database.sql

to run the script.

5. Enter the command
< Create_ItemSync_Dables.sql

to run the script to create the tables.

Oracle:

1. Transfer the files:

```
<ICS_installation_path>/collaborations/dependencies  
/UCCnet/oracle/Create_ItemSync_Database.sql  
<ICS_installation_path>/collaborations/dependencies  
/UCCnet/oracle/Create_ItemSync_Tables.sql
```

to the UNIX system using FTP in ASCII mode. Place the file in the
<ICS_installation_path>/collaborations/dependencies/UCCnet/oracle
directory.

2. At the UNIX system, move to the directory:
<ICS_installation_path>/collaborations/dependencies/UCCnet/oracle
3. Create a database named ITEMSYNC.
4. Enter the command:
sqlplus dbuser/passwd@SID @Create_ItemSync_Tables.sql

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.

Table 1. Selected attribute values for MO_DataHandler_UCCnetXMLConfig metaobject

Attribute name	Column	Setting
BOPrefix	Default	<ul style="list-style-type: none"> • UCCnetXSD (for iSoft connectivity) • UCCnetTPIXSD (for TPI connectivity) • UCCnetJMSXSD (For WebSphere Business Integration Connect-Java Message Service connectivity)
DefaultEscapeBehavior	Default	true
DTDPath	Default	Path to Envelope.xsd file. For example, <code><ICS_installation_path>\UCCnet\XSDs\uccnet\2.2\Envelope.xsd</code> Notes: <ol style="list-style-type: none"> 1. The attribute values shown are examples only. The actual value must be the fully qualified path to the Envelope.xsd file on your system. 2. The Envelope file is available from the UCCnet eRoom. You must have a valid UCCnet eRoom user ID and password to obtain them.
IgnoreUndefinedElements	Default	true
Validation	Default	false (This must be set to false to keep the DataHandler from validating the messages against the UCCnetGBO envelope since this business object has been extended beyond the envelope.xsd file.)

2. Edit the MO_DataHandler_Default metaobject by setting the following attribute, then save it as MO_DataHandler_UCCnet_envelope.

Table 2. Selected attribute value for MO_DataHandler_UCCnet_envelope metaobject

Attribute name	Column	Setting
text_xml	Type	MO_DataHandler_UCCnetXMLConfig

3. Edit the MO_JTextConnector_Default metaobject by setting the following attributes, then save it as MO_JTextRWLConnector_Default.

Table 3. Selected attribute values for MO_JTextRWLConnector_Default metaobject

Attribute name	Column	Setting
ArchiveDir	Default	<Name of the directory where archive XML files will be written> (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	<Name of the directory to obtain input XMLs (Events)> (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	<Name of the directory where the XML files will be 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 testing your installation, follow the instructions to create a JTextConnector meta object

-

Edit the MO_JTextConnector_Default metaobject by setting the following attributes, then save it as MO_JTextxxxConnector_Default Where xxx is to be replaced with TPI if using TPI connection, JMS if JMS connection, or omit the xxx characters if using an iSoft connector.

Table 4. Selected attribute values for MO_JTextxxxConnector_Default metaobject

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

If you are planning to use the schema-supported Catalogue Item Publication functions, you must create a second instance of the JTextxxx connector metaobjects. Edit the MO_JTextxxxConnector_Default by setting the following attributes, then save it as MO_JTextxxx2Connector_Default.

Table 5. Selected attribute values for MO_JTextxxx2Connector_Default metaobject

Attribute name	Column	Setting
EventDataHandler	Type	MO_DataHandler_UCCnetXMLConfig
OutputDataHandler	Type	MO_DataHandler_UCCnetXMLConfig
OutputDir	Default	<Name of the directory where the XML files will be written> (for example, C:\IBM\WebSphereICS\UCCnet\JTextxxx2\out). Create this directory if it does not already exist.
OutputExt	Default	xml
OutputFileName	Default	Native
ArchiveDir	Default	<Name of the directory where archive XML files will be written> (for example, C:\IBM\WebSphereICS\UCCnet\JTextxxx2\archive). Create this directory if it does not already exist.
EventDir	Default	<Name of the directory to obtain input XML (Events)> (for example, C:\IBM\WebSphereICS\UCCnet\JTextxxx2\event). Create this directory if it does not already exist.
EventExt	Default	xml
EndBODElimiter	Default	EOF

5. If you are exchanging messages with UCCnet through an AS2/EDIINT interface protocol, follow the instructions that apply to the connectivity that you are using:

iSoft connectivity:

Use the Business Object Designer to create a metaobject called MO_iSoftAdapterConfig with the attributes shown in the following table. If you are using XSD CIP operation, you need two instances of the iSoft connector. However, because both instances use the same metaobject, only one metaobject is needed.

Table 6. 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
UCCnetXSD_envelope_Create (for schema support)	String		If the application specific information provided for the Default attribute is insufficient for your installation, additional information on this metaobject is available in the Adapter for iSoft Peer-to-Peer Agent User Guide.

TPI connectivity:

If TPI connectivity is used, No configuration metaobject 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 7. 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.
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 Java Message Service Agent User Guide.

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

Table 8. Selected attribute values for EmailNotification business object

Name	Column	Setting
RecipientName	Default	E-mail address of recipient.
FromAddress	Default	E-mail address of sender.

Creating the database connection pool

The Item Sync collaborations use a database connection pool to access the ITEMSYNC database PROCESSED_GTIN and AUDIT_LOG tables. See IBM

WebSphere InterChange Server System Administration Guide for instructions on creating a connection pool. Ensure the following values are set when you define the database connection pool:

- Use the appropriate Database Driver, Login ID, and Password. The Login ID and password used in the connection pool must match those used when creating the database and tables.
- The database name must match the name used in the database creation script. The default value in the script is ITEMSYNC.
- The Connection Pool name must match the value set in the collaboration object's DB_CONN_POOL_NAME collaboration property. The default value of the property is ITEMSYNC_DATA.

Creating and configuring the connectors

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

- Create and configure the JTextRWLConnector in every installation. See “Creating and configuring the JTextRWLConnector.”
- 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:

Table 9. Connectors and connector information

Connectivity type and protocol	Connector to configure
Exchanging messages with UCCnet through an AS2/EDIINT interface protocol and using iSoft connectivity	“Creating and configuring the iSoftConnector” on page 14
Exchanging messages with UCCnet through an AS2/EDIINT interface protocol and using TPI connectivity	“Creating and configuring the TPIConnector” on page 18
Exchanging messages with UCCnet through an AS2/EDIINT interface protocol and are using the WebSphere Business Integration Connect interface	“Creating and configuring the first instance of the JMSConnector” on page 24
Testing your installation, and are using iSoft connectivity	“Creating and configuring the JTextConnector” on page 21

- Configure the PortConnector in every installation, as detailed in the section “Configuring the PortConnector” on page 30.
- If you want to use the e-mail capabilities of the Item Synchronization for Suppliers solution, configure the EmailConnector, as detailed in the section “Configuring the EmailConnector” on page 30.

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, a scheduler operation performed by an operating system must be

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, complete the following steps:

1. Open the System Manager.
2. Save the JTextRWLConnector by doing the following:
 - 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 appropriate business objects. Use the values shown in the following tables.

Table 10. Standard property

Property name	Value
ApplicationName	JTextRWLConnector

Table 11. Supported business objects

Business object name	Agent support required?
MO_JTextRWLConnector_Default	Yes
MO_DataHandler_UCCnet_envelope	Yes
One of the following: <ul style="list-style-type: none"> • 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 bindings.

Table 12. Associated maps

Business object name	Map name
UCCnetXSD_envelope for iSoft connectivity	UCCnetXSD_envelope_to_UCCnetGBO_envelope
UCCnetTPIXSD_envelope for TPI connectivity	UCCnetTPIXSD_envelope_to_UCCnetGBO_envelope
UCCnetJMSXSD_envelope for WebSphere Business Integration Connect	UCCnetJMSXSD_envelope_to_UCCnetGBO_envelope
UCCnetGBO_envelope	<ul style="list-style-type: none"> • iSoft connectivity: UCCnetGBO_envelope_to_UCCnetXSD_envelope • TPI connectivity: UCCnetGBO_envelope_to_UCCnetTPIXSD_envelope • WebSphere Business Integration Connect: UCCnetGBO_envelope_to_UCCnetJMSXSD_envelope

3. Save the configuration file (**File > Save > To Project**), then close the Connector Configurator.
4. Create the JTextRWLConnector Agent, as follows:

Windows operating systems:

- 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 InterChange Server, as shown in the following example:

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

UNIX operating systems:

(There is nothing to perform at this point. Use these steps later when you actually want to start the Jtext Connector Agent after the connector has been 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 InterChange Server:

```
start_JText.sh JTextRWL ICS_server_name
```

5. 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, 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/JTEXTRWLCONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/JTEXTRWLCONNECTOR) USAGE (NORMAL)
END
```

Creating and configuring the iSoftConnector

Open the iSoft connector in the Connector Configurator and 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.

Creating and configuring the first instance of the iSoftConnector: Perform the following steps to configure the first instance of 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 example, C:\Program Files\IBM\WebSphere MQ\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.

Table 13. Connector-specific properties

Property name	Value
ArchiveQueue	Queue to which copies of successfully processed messages are sent (for example, 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 example, 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	Semicolon-delimited list of message queues that are polled by the connector for new messages (for example, 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 example, 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 example, queue://my_p2p_agent_queue_manager_name/unsubscribed)
UseDefaults	true (add this property if it does not appear in the list of connector-specific properties)

Table 14. Supported business objects

Business object name	Agent support required?
MO_DataHandler_UCCnet_envelope	Yes
MO_iSoftAdapterConfig	Yes
UCCnetXSD_envelope	Yes
UCCnetGBO_envelope	No
Retail_Item	No
Retail_Price	No

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

Table 15. Associated maps

Business object name	Map name
Retail_Item	Retail_Item_to_UCCnetXSD_envelope_registerCommand_itemAddChange
Retail_Price	Retail_Price_to_UCCnetXSD_envelope
UCCnetGBO_envelope	UCCnetGBO_envelope_to_UCCnetXSD_envelope
UCCnetXSD_envelope	UCCnetXSD_envelope_to_UCCnetGBO_envelope

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

- 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, 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/ISOFTCONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/ISOFTCONNECTOR) USAGE (NORMAL)
END
```

Creating and configuring the second instance of the iSoftConnector: If you are setting up for XSD CIP operation, you need two instances of the iSoft connector. Do the following steps to configure the second instance:

- Open the System Manager.
- Save the iSoft2Connector by doing the following:
 - Open the iSoftConnector.
 - Click **File > Save As > To Project**.
 - Save the connector as iSoft2Connector.
- Open the iSoft2Connector in the Connector Configurator and configure this connector to include the connector-specific configuration properties and appropriate business objects. Use the values shown in the following tables.

Table 16. Connector-specific properties

Property name	Value
ArchiveQueue	Queue to which copies of successfully processed messages are sent (for example, queue://my_p2p_agent_queue_manager_name/archive2)
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 example, queue://my_p2p_agent_queue_manager_name/error2)
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 example, queue://my_p2p_agent_queue_manager_name/inbox3; queue://my_p2p_agent_queue_manager_name/inbox4) Note: The inbox for the second instance of the iSoft connector must not be the same as the inbox for the first instance of the iSoft connector. This instance of the connector is not configured to receive business objects.
InProgressQueue	Message queue where messages are held during processing (for example, queue://my_p2p_agent_queue_manager_name/in_progress2)
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 example, queue://my_p2p_agent_queue_manager_name/unsubscribed2)
UseDefaults	true (add this property if it does not appear in the list of connector-specific properties)

Table 17. Supported business objects

Business object name	Agent support required?
MO_DataHandler_UCCnet_envelope	Yes
MO_iSoftAdapterConfig	Yes
UCCnetXSD_envelope	Yes
Retail_Item	No

4. Create the iSoft2Connector Agent, as follows:

Windows operating systems:

- 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 iSoftConnector short cut and rename it to iSoft2Connector.
- d. Right-click the iSoft2Connector short cut and select **Properties**.
- e. Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to iSoft2, where *ICS_server_name* refers to the name of the InterChange Server, as shown in the following example:

```
<install_path>\connectors\JText\start_iSoft.bat iSoft2 ICS_server_name
```

UNIX operating systems:

(There is nothing to perform at this point. Use these steps later when you actually want to start the Jtext Connector Agent after the connector has been deployed)

- a. Change to the *<install_path>/IBM/WebSphereICS/connectors/iSoft/* directory.
 - b. Run iSoft2. by entering the following command, where *ICS_server_name* refers to the name of the InterChange Server:

```
start_jText.sh iSoft2 ICS_server_name
```
5. Save the configuration (**File > Save > To Project**), then go back to the **Associated Map** tab and set the explicit bindings.

Table 18. Associated maps

Business object name	Map name
Retail_Item	Retail_Item_to_UCCnetXSD_envelope_publicationCommand_catalogueItem

6. Save the configuration (**File > Save > To Project**) then close the Connector Configurator.
7. If you are using WebSphere MQ as your connector transport, create the iSoft2Connector 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/ISOFT2CONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/ISOFT2CONNECTOR) USAGE (NORMAL)
END
```

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

Creating and 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. If you are setting up for XSD CIP operation, you need two instances of the TPI connector. Otherwise, you require only one. Perform the following steps to configure the TPIConnector:

Creating and configuring the first instance of the TPIConnector:

1. Create the trading partner configuration file **tpcfg.txt**, which includes one tab-delimited line for each trading partner formatted as follows:

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 example, 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 19. Connector-specific properties

Property name	Value
ArchiveProcessedDocDir	Directory where processed document metaevents are archived (for example, C:\TPISolo\data\uccnet2\archive).
DataHandlerConfigMO	MO_DataHandler_UCCnet_envelope
DefaultXMLMimeType	text/xml
DocumentOutDir	Directory location where outbound documents are written temporarily before TPI processes them (for example, C:\TPISolo\data\uccnet2\xmlout).
MetaEventDir	Directory used to persist the TPI event information for recovery purposes (for example, 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 example, C:\IBM\WebSphereICS\connectors\TPI\tpcfg.txt).
WaitForMDN	false (MDNs are not supported by this solution).

Table 20. Supported business objects

Business object name	Agent support required?
MO_DataHandler_UCCnet_envelope	Yes
UCCnetTPIXSD_envelope	Yes
Retail_Item	No
Retail_Price	No
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 21. Associated maps

Business object name	Map name
Retail_Item	Retail_Item_to_UCCnetTPIXSD_envelope_registerCommand_itemAddChange
Retail_Price	Retail_Price_to_UCCnetXSD_envelope
UCCnetGBO_envelope	UCCnetGBO_envelope_to_UCCnetTPIXSD_envelope
UCCnetTPIXSD_envelope	UCCnetTPIXSD_envelope_to_UCCnetGBO_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 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/TPICONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/TPICONNECTOR) USAGE (NORMAL)
END
```

When a message is sent to UCCnet, the **SenderId** and **ReceiverId** values in the UCCnetTPIXSD_envelope business object are set by the maps when this business object is created.

Creating and configuring the second instance of the TPIConnector: This connector is required only if you are setting up for XSD CIP operation.

1. Create the TPI2Connector using the System Manager, as follows:
 - a. Open the System Manager.
 - b. Save the TPI2Connector by doing the following:
 - 1) Open the TPIConnector.
 - 2) Click **File > Save As > To Project**.
 - 3) Save the connector as TPI2Connector.
 - c. Create the TPI2Connector Agent, as follows for your operating system:

Windows operating systems:

- 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 TPIConnector short cut and rename it to TPI2Connector.
- d. Right-click the TPIConnector short cut and select **Properties**.
- e. Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to TPI2, where *ICS_server_name* refers to the name of the InterChange Server, as shown in the following example:

```
<install_path>\connectors\JText\start_TPI.bat TPI2 ICS_server_name
```

UNIX operating systems: (There is nothing to perform at this point. Use these steps later when you actually want to start the Jtext Connector Agent after the connector has been deployed)

- a. Change to the `<install_path>/IBM/WebSphereICS/connectors/TPI/` directory.
- b. Run TPI by entering the following command, where `ICS_server_name` refers to the name of the InterChange Server:

```
start_TPI.sh TPI2 ICS_server_name
```

Configure the TPI2Connector as follows:

1. Open the TPI2Connector in the Connector Configurator and configure this connector to include the connector-specific configuration properties and appropriate business objects. Use the values shown in the following tables.

Table 22. Connector-specific properties

Property name	Value
ArchiveProcessedDocDir	Directory where processed document metaevents are archived (for example, <code>C:\TPISolo\data\uccnet2\archive2</code>).
DataHandlerConfigMO	MO_DataHandler_UCCnet_envelope
DefaultXMLMimeType	text/xml
DocumentOutDir	Directory location where outbound documents are written temporarily before TPI processes them (for example, <code>C:\TPISolo\data\uccnet2\xmlout</code>). Note: The output directory for this instance of the connector is the same as the output directory for the first instance of the connector.
MetaEventDir	Directory used to persist the TPI event information for recovery purposes (for example, <code>C:\TPISolo\data\uccnet2\xmlin2</code>).
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 example, <code>C:\IBM\WebSphereICS\connectors\TPI\tpcfg.txt</code>).
WaitForMDN	false (MDNs are not supported by this solution).

Table 23. Supported business objects

Business object name	Agent support required?
MO_DataHandler_UCCnet_envelope	Yes
UCCnetTPIXSD_envelope	Yes
Retail_Item	No

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

Table 24. Associated maps

Business object name	Map name
Retail_Item	Retail_Item_to_UCCnetTPIXSD_envelope_publicationCommand_catalogueItem

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 TPI2Connector 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/TPI2CONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/TPI2CONNECTOR) USAGE (NORMAL)
END
```

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

Creating and configuring the JTextConnector

Create and configure this connector only if you are using the iSoft Peer-to-Peer Agent and are communicating with UCCnet through the UCCnet CLU or testing your installation. This is a copy of the JTextConnector. The JTextConnector uses the IBM WebSphere Business Integration Data Handler for XML and generates the same XML output.

If you are setting up for XSD CIP operation, you need to configure two instances of the JText connector. Otherwise, you need to create and configure only the first instance.

Creating and configuring the first instance of the JTextxxxConnector:

1. Create the JTextxxxConnector using the System Manager, as follows:

- a. Open the System Manager.
- b. Save the JTextxxxConnector by doing the following:
 - 1) Open the JTextConnector.
 - 2) Click **File > Save As > To Project**.
 - 3) Save the connector as JTextxxxConnector.
- c. Create the JTextxxxConnector Agent, as follows:

Windows operating systems:

- 1) In the Windows task bar, right-click **Start** and select **Open All Users**.
- 2) Navigate to the folder on your system that contains the installed connectors by clicking **Programs > IBM WebSphere Business Integration Adapters > Adapters > Connectors**.
- 3) Copy the JText Connector short cut and rename it to JTextxxxConnector.
- 4) Right-click the JTextxxxConnector short cut and select **Properties**.
- 5) Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to JTextxxx, where *ICS_server_name* refers to the name of the InterChange Server, as shown in the following example:

```
<install_path>\connectors\JText\start_JText.bat JTextxxx ICS_server_name
```

UNIX operating systems:

- 1) Change to the <install_path>/IBM/WebSphereICS/connectors/JText/ directory.
 - 2) Run JTextxxx by entering the following command, where *ICS_server_name* refers to the name of the InterChange Server:
- ```
start_JText.sh JTextxxx ICS_server_name
```

Perform the following steps to configure the JTextxxxConnector:

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

Table 25. Standard property

| Property name   | Value             |
|-----------------|-------------------|
| ApplicationName | JTextxxxConnector |

Table 26. Supported business objects

| Business object name           | Agent support required? |
|--------------------------------|-------------------------|
| MO_JTextxxxConnector_Default   | Yes                     |
| MO_DataHandler_UCCnet_envelope | Yes                     |
| UCCnetXSD_envelope             | Yes                     |
| Retail_Item                    | No                      |
| Retail_Price                   | No                      |
| UCCnetGBO_envelope             | No                      |

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

Table 27. Associated maps

| Business object name | Map name                                                        |
|----------------------|-----------------------------------------------------------------|
| Retail_Item          | Retail_Item_to_UCCnetXSD_envelope_registerCommand_itemAddChange |
| UCCnetGBO_envelope   | UCCnetGBO_envelope_to_UCCnetXSD_envelope                        |
| UCCnetXSD_envelope   | UCCnetXSD_envelope_to_UCCnetGBO_envelope                        |
| Retail_Price         | Retail_Price_to_UCCnetXSD_envelope                              |

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 JTextxxxConnector 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
```

**Creating and configuring the second instance of the JTextxxxConnector:** The second instance of the JTextxxxConnector is used only for XSD CIP operation. If it is required, create it as follows:

1. Create the JTextxxx2Connector using the System Manager, as follows:
  - a. Open the System Manager.
  - b. Save the JTextxxx2Connector by doing the following:
    - 1) Open the JTextxxxConnector.
    - 2) Click **File > Save As > To Project**.
    - 3) Save the connector as JTextxxx2Connector.
  - c. Create the JTextxxx2Connector Agent, as follows:
 

**Windows operating systems:**

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

- 2) Navigate to the folder on your system that contains the installed connectors by clicking **Programs > IBM WebSphere Business Integration Adapters > Adapters > Connectors**.
- 3) Copy the JTextxxxConnector short cut and rename it to JTextxxx2Connector.
- 4) Right-click the JTextxxx2Connector short cut and select **Properties**.
- 5) Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to JTextxxx2, where *ICS\_server\_name* refers to the name of the InterChange Server, as shown in the following example:  
`<install_path>\connectors\JText\start_JText.bat JTextxxx2 ICS_server_name`

**UNIX operating systems:**

- 1) Change to the `<install_path>/IBM/WebSphereICS/connectors/JText/` directory.
- 2) Run JTextxxx2. by entering the following command, where *ICS\_server\_name* refers to the name of the InterChange Server:  
`start_JText.sh JTextxxx2 ICS_server_name`

Perform the following steps to configure the JTextxxx2Connector:

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

*Table 28. Standard property*

| Property name   | Value              |
|-----------------|--------------------|
| ApplicationName | JTextxxx2Connector |

*Table 29. Supported business objects*

| Business object name           | Agent support required? |
|--------------------------------|-------------------------|
| MO_JTextxxx2Connector_Default  | Yes                     |
| MO_DataHandler_UCCnet_envelope | Yes                     |
| UCCnetXSD_envelope             | Yes                     |
| Retail_Item                    | No                      |

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

*Table 30. Associated map*

| Business object name | Map name                                                           |
|----------------------|--------------------------------------------------------------------|
| Retail_Item          | Retail_Item_to_UCCnetXSD_envelope_publicationCommand_catalogueItem |

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 JTextiSoft2Connector 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/JTEXTxxx2CONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/JTEXTxxx2CONNECTOR) USAGE (NORMAL)
END
```

## Creating and configuring the JMSConnector

Create and configure this connector only if you are communicating with UCCnet through a WebSphere Business Integration Connect interface. If you are setting up your system to use the XSD CIP operation, you must create and configure two instances of the JMSConnector connector.

**Creating and configuring the first instance of the JMSConnector:** To create and configure the JMSConnector, complete the following steps:

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 (Enter the following on one line. It is formatted for readability.)

```
DEFINE CHANNEL(channel_name) CHLTYPE(SDR) XMITQ(xmit q_name) CONNAME
('IP_address_of_Business_Integration_Conenct_machine(listener_port)%csq;) 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.sun.jndi.ldap.LdapCtxFactory
```

- 2) Uncomment the following line:

```
INITIAL_CONTEXT_FACTORY=com.sun.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.jndi 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)

```

If you are setting up for setting up for XSD CIP operation, you must also include the following additional queue declarations:

```

DEFINE Q(INPUTQ2) +
QUEUE(INPUTQ2) +
TARGCLIENT(JMS)
*
DEFINE Q(REPLYQ2) +
QUEUE(REPLYQ2) +
TARGCLIENT(JMS)
*
DEFINE Q(REMOTEQ2) +
QUEUE(REMOTEQ2) +
TARGCLIENT(JMS)
*
DEFINE Q(ERRORQ2) +
QUEUE(ERRORQ2) +
TARGCLIENT(JMS)
*
DEFINE Q(IN_PROGRESSQ2) +
QUEUE(IN_PROGRESSQ2) +
TARGCLIENT(JMS)
*
DEFINE Q(UNSUBSCRIBEQ2) +
QUEUE(UNSUBSCRIBEQ2) +
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. Open the JMSConnector. 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 31. Connector-specific properties*

| 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 24.                                                                                                   |
| ReplyToDestination         | The name of the queue, in full URL format, to which replies are sent. For example,<br>queue://my_queue_manager_name/REPLYQ                                                                                                                   |
| UnsubscribedQueue          | The name of the queue, in full URL format, that receives unsubscribed messages. For example,<br>queue://my_queue_manager_name/UNSUBSCRIBEQ                                                                                                   |
| 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,<br>file:/C:/IBM/WebSphereICS/connectors/JMS |
| InProgressQueue            | The name of the queue, in full URL format, that holds messages during processing. For example,<br>queue://my_queue_manager_name/IN_PROGRESSQ                                                                                                 |
| Error Queue                | The name of the queue, in full URL format, that receives messages that cannot be processed. For example,<br>queue://my_queue_manager_name/ERROR                                                                                              |
| DataHandlerConfigMO        | MO_DataHandler_UCCnet_envelope                                                                                                                                                                                                               |
| ConfigurationMetaObject    | MO_JMSAdapterConfig                                                                                                                                                                                                                          |
| DataHandlerMimeType        | text/xml                                                                                                                                                                                                                                     |
| QueueConnectionFactoryName | The name of the queue connection factory. For example, WBIC_JMSQCF.                                                                                                                                                                          |

Table 31. Connector-specific properties (continued)

| Property name | Value                                                                                                                                      |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| DefaultVerb   | Create                                                                                                                                     |
| InputQueue    | The name of the queue, in full URL format, that the connector polls for new messages. For example,<br>queue://my_queue_manager_name/INPUTQ |
| PollQuantity  | 1                                                                                                                                          |

**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 32. Supported business objects

| Business object name           | Agent support required? |
|--------------------------------|-------------------------|
| MO_DataHandler_UCCnet_envelope | Yes                     |
| MO_JMSAdapterConfig            | Yes                     |
| UCCnetGBO_envelope             | No                      |
| UCCnetJMSXSD_envelope          | Yes                     |
| Retail_Item                    | No                      |
| Retail_Price                   | No                      |

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

Table 33. Associated maps

| Business object name  | Map name                                                           |
|-----------------------|--------------------------------------------------------------------|
| UCCnetJMSXSD_envelope | UCCnetJMSXSD_envelope_to_UCCnetGBO_envelope                        |
| UCCnetGBO_envelope    | UCCnetGBO_envelope_to_UCCnetJMSXSD_envelope                        |
| Retail_Item           | Retail_Item_to_UCCnetJMSXSD_envelope_registerCommand_itemAddChange |
| Retail_price          | Retail_Price_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 second instance of the JMSConnector:** If you are setting up for XSD CIP operation, you need two instances of the JMS connector. Perform the following steps to configure the second instance:

1. Open the System Manager.
2. Save the JMS2Connector by doing the following:
  - a. Open the JMSConnector.
  - b. Click **File > Save As > To Project**.
  - c. Save the connector as JMS2Connector.
3. Create the JMS2Connector Agent, as follows:
 

**On Windows operating systems:**

  - 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 JMSConnector short cut and rename it to JMS2Connector.
  - d. Right-click the JMS2Connector short cut and select **Properties**.
  - e. Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to JMS2, where *ICS\_server\_name* refers to the name of the InterChange Server, as shown in the following example:  
`<install_path>\connectors\JText\start_JMS.bat JMS2 ICS_server_name`

**On UNIX operating systems:**

  - a. Change to the `<install_path>/IBM/WebSphereICS/connectors/JMS/` directory.
  - b. Run JMS2. by entering the following command, where *ICS\_server\_name* refers to the name of the InterChange Server:  
`start_JText.sh JMS2 ICS_server_name`
4. Open the JMS2Connector in the Connector Configurator and configure it to include the connector-specific configuration properties and appropriate business objects. Use the values shown in the following tables.

Table 34. Connector-specific properties

| Property name             | Value                                                                                                                                                                                                                                        |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CTX_InitialContextFactory | com.sun.jndi.fscontext.ReffFSContextFactory (This is the same name that you uncommented in the JMSAdmin.config file in step 2a2 on page 24).                                                                                                 |
| ReplyToQueue              | The name of the queue, in full URL format, to which replies are sent. For example,<br>queue://my_queue_manager_name/REPLYQ2                                                                                                                  |
| UnsubscribedQueue         | The name of the queue, in full URL format, that receives unsubscribed messages. For example,<br>queue://my_queue_manager_name/UNSUBSCRIBEQ2                                                                                                  |
| 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,<br>file:/C:/IBM/WebSphereICS/connectors/JMS |
| InProgressQueue           | The name of the queue, in full URL format, that holds messages during processing. For example,<br>queue://my_queue_manager_name/IN_PROGRESSQ2                                                                                                |
| Error Queue               | The name of the queue, in full URL format, that receives messages that cannot be processed. For example,<br>queue://my_queue_manager_name/ERROR2                                                                                             |



Table 34. Connector-specific properties (continued)

| Property name              | Value                                                                                                                                       |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| DataHandlerConfigMO        | MO_DataHandler_UCCnet_envelope                                                                                                              |
| ConfigurationMetaObject    | MO_JMSAdapterConfig                                                                                                                         |
| DataHandlerMimeType        | text/xml                                                                                                                                    |
| QueueConnectionFactoryName | The name of the queue connection factory. For example, WBIC_JMSQCF.                                                                         |
| InputQueue                 | The name of the queue, in full URL format, that the connector polls for new messages. For example,<br>queue://my_queue_manager_name/INPUTQ2 |
| PollQuantity               | 1                                                                                                                                           |

**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 35. Messaging properties

| Property name    | Value                                                                              |
|------------------|------------------------------------------------------------------------------------|
| MessagingType    | MQSERIES                                                                           |
| HostName         | The name of the machine running the WebSphere Interchange Server.                  |
| QueueManagerName | The name of the queue manager defined for the WebSphere Interchange Server.        |
| ClientChannel    | The name of the MQ channel defined for WebSphere Interchange Server communication. |
| Port             | The port number that the WebSphere Interchange Server listens on.                  |

Table 36. Supported business objects

| Business object name           | Agent support required? |
|--------------------------------|-------------------------|
| MO_DataHandler_UCCnet_envelope | Yes                     |
| MO_JMSAdapterConfig            | Yes                     |
| UCCnetJMSXSD_envelope          | Yes                     |
| Retail_Item                    | No                      |

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

Table 37. Associated map

| Business object name | Map name                                                              |
|----------------------|-----------------------------------------------------------------------|
| Retail_Item          | Retail_Item_to_UCCnetJMSXSD_envelope_publicationCommand_catalogueItem |

6. Save the configuration (**File > Save > To Project**) then close the Connector Configurator.
7. If you are using WebSphere MQ as your connector transport, create the JMS2Connector 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/JMS2CONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/JMS2CONNECTOR) USAGE (NORMAL)
END
```

Refer to the *Adapter for JMS 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 38. Supported business object

| Business object name | Agent support required? |
|----------------------|-------------------------|
| Retail_Item          | Yes                     |
| UCCnetGBO_envelope   | Yes                     |
| Retail_Price         | Yes                     |

2. Save the configuration file (**File > Save > To Project**).
3. 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, 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/PORTCONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/PORTCONNECTOR) USAGE (NORMAL)
END
```

## Configuring the EmailConnector

To create the EmailConnector, complete the following steps:

1. In the Windows task bar, right-click **Start** and select **Open All Users**.
2. Navigate to the folder on your system that contains the installed connectors by clicking **Programs > IBM WebSphere Business Integration Adapters > Adapters > Connectors**.
3. Right-click the Email Connector and select **Properties > Modify**.
4. Click the **Short cut** tab, edit the **Target** field, and set the first command line argument to EMail, where *ICS\_server\_name* refers to the name of the InterChange Server, as shown in the following example:  

```
<install_path>\connectors\email\start_email.bat
 EMail ICS_server_name
```
5. Configure the EmailConnector to include the following configuration property value.

Table 39. Connector-specific property

| Property name | Property value                           |
|---------------|------------------------------------------|
| SMTP_MailHost | Hostname or IP address of a mail server. |

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

## 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.

### Creating and configuring a UCCnet\_ItemSync collaboration object and making its port connections

1. Create and configure the following collaboration object based on the UCCnet\_ItemSync collaboration template.

Table 40. Collaboration object name: UCCnet\_ItemSyncObj

| Port                  | Type          | Bind to                      |
|-----------------------|---------------|------------------------------|
| From                  | Collaboration | CommandRouterObj.ToRCIR      |
| To                    | Connector     | AS2 channel connector        |
| Error                 | Connector     | Supplier choice of connector |
| DestinationAppReceive | Connector     | PortConnector                |

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

Table 41. UCCnet\_ItemSyncObj collaboration object configuration properties

| Property name      | Property description                                                                                                                                                                                                                                                                                                                                     | Required                                                                                                 |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| SEND_EMAIL_TO      | Defines the e-mail address to which problems detected during execution of collaboration object processing are sent. This entry is for InterChange Server administrators. Do not confuse this property with the Email notification address field, which is configured from the Collaboration General Properties tab when creating a collaboration object. | No. Requires the Email Connector to run if email is entered. The SEND_EMAIL property must be set also.   |
| SEND_EMAIL         | Specifies whether e-mail is sent to the address set in the SEND_EMAIL_TO property.                                                                                                                                                                                                                                                                       | Yes, if you want to send e-mail. To send e-mail, set the value to all. Otherwise, set the value to none. |
| DB_CONN_POOL_NAME  | The name of the connection data pool. The default value is ITEMSYNC_DATA..                                                                                                                                                                                                                                                                               | Yes                                                                                                      |
| FAILURE_EXCEPTIONS | Specifies exception message numbers that would cause the collaboration object to fail flow                                                                                                                                                                                                                                                               | Yes, if a particular exception number should cause a failed flow situation                               |

### Creating and configuring the CommandRouter collaboration object and making its port connections

1. Create and configure the following collaboration object based on the CommandRouter collaboration template:

Table 42. Collaboration object name: CommandRouterObj

| Port     | Type                 | Bind to                      |
|----------|----------------------|------------------------------|
| From     | Connector            | Supplier's choice of adapter |
| ToCINCIP | Collaboration object | CIN_CIP_DispatcherObj.From   |

Table 42. Collaboration object name: *CommandRouterObj* (continued)

| Port   | Type                 | Bind to                      |
|--------|----------------------|------------------------------|
| ToRCIR | Collaboration object | UCCnet_ItemSyncObj.From      |
| ToCI   | Collaboration object | CI_SyncObj.From              |
| Error  | Connector            | Supplier's choice of adapter |

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

Table 43. *CommandRouterObj* collaboration object configuration properties

| Property name      | Property description                                                                                                                                                                                                                                                                                                                                     | Required                                                                                                 |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| SEND_EMAIL_TO      | Defines the e-mail address to which problems detected during execution of collaboration object processing are sent. This entry is for InterChange Server administrators. Do not confuse this property with the Email notification address field, which is configured from the Collaboration General Properties tab when creating a collaboration object. | No. Requires the Email Connector to run if email is entered. The SEND_EMAIL property must be set also.   |
| SEND_EMAIL         | Specifies whether e-mail is sent to the address set in the SEND_EMAIL_TO property.                                                                                                                                                                                                                                                                       | Yes, if you want to send e-mail. To send e-mail, set the value to all. Otherwise, set the value to none. |
| FAILURE_EXCEPTIONS | Specifies exception message numbers that would cause the collaboration object to fail flow                                                                                                                                                                                                                                                               | Yes, if a particular exception number should cause a failed flow situation                               |

### Creating and configuring a *CI\_Sync* collaboration object and making its port connections

1. Create and configure the following collaboration object based on the *CI\_Sync* collaboration template.

Table 44. Collaboration object name: *CI\_SyncObj*

| Port                   | Type          | Bind to                                 |
|------------------------|---------------|-----------------------------------------|
| From                   | Collaboration | CommandRouterObj.ToCI                   |
| To                     | Connector     | AS2 channel connector (second instance) |
| Error                  | Connector     | Supplier's choice of adapter            |
| DestinationAppRetrieve | Connector     | PortConnector                           |

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

Table 45. *CI\_SyncObj* collaboration properties

| Property name      | Description                                                                                                                                                                                                                                                                                                                                              | Required                                                                                                 |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| SEND_EMAIL_TO      | Defines the e-mail address to which problems detected during execution of collaboration object processing are sent. This entry is for InterChange Server administrators. Do not confuse this property with the Email notification address field, which is configured from the Collaboration General Properties tab when creating a collaboration object. | No. Requires the Email Connector to run if email is entered. The SEND_EMAIL property must also be set.   |
| SEND_EMAIL         | Specifies whether e-mail is sent to the address set in the SEND_EMAIL_TO property.                                                                                                                                                                                                                                                                       | Yes, if you want to send e-mail. To send e-mail, set the value to all. Otherwise, set the value to none. |
| DB_CONN_POOL_NAME  | The name of the connection data pool (default value = ITEMSYNC_DATA)                                                                                                                                                                                                                                                                                     | Yes                                                                                                      |
| FAILURE_EXCEPTIONS | Specifies exception message numbers that would cause the collaboration object to fail flow                                                                                                                                                                                                                                                               | Yes, if a particular exception number should cause a failed flow situation                               |

### Creating and configuring the CIN\_CIP\_Dispatcher collaboration objects and making its port connections

1. Create and configure the following collaboration object based on the CIN\_CIP\_Dispatcher collaboration template:

Table 46. Collaboration object name: *CIN\_CIP\_DispatcherObj*. Used for both CIN and CIP operation

| Port                   | Type                 | Bind to                      |
|------------------------|----------------------|------------------------------|
| From                   | Collaboration object | CommandRouterObj.ToCINCIP    |
| To                     | Connector            | AS2 channel connector        |
| DestinationAppRetrieve | Connector            | PortConnector                |
| Error                  | Connector            | Supplier's choice of adapter |

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

Table 47. *CIN\_CIP\_DispatcherObj* collaboration properties

| Property name     | Description                                                          | Required |
|-------------------|----------------------------------------------------------------------|----------|
| DB_CONN_POOL_NAME | The name of the connection data pool (default value = ITEMSYNC_DATA) | Yes      |

Table 47. CIN\_CIP\_DispatcherObj collaboration properties (continued)

| Property name               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Required                                                                   |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| GLN_ATTRIBUTE               | <p>Identifies where in the UCCnetGBO business object the GLN retrieved from the Dispatcher GLN file should be placed. This property is required.</p> <p><b>CIN operation</b> Default:<br/> <code>ROOT.body[0].transaction.command[0].notifyCommand.<br/> notifyCommandOperand<br/> .catalogueItemNotification.<br/> catalogueItem.dataRecipient</code></p> <p><b>CIP operation</b> Default:<br/> <code>ROOT.body[0].transaction.command[0].publicationCommand.<br/> publicationCommandOperand<br/> .catalogueItemPublication.<br/> publishToGLN</code></p> | Yes                                                                        |
| TO_UCCNETGBO_PROCESSING_MAP | <p>Called by the CIN_CIP_Dispatcher object to map incoming Retail_Item objects to the outgoing Catalogue Item Notification or Catalogue Item Publication message.</p> <p><b>CIN operation</b> Default:<br/> <code>Retail_Item_to_UCCnetGBO_envelope<br/> _notifyCommand_catalogueItem</code></p> <p><b>CIP operation</b> Default:<br/> <code>Retail_Item_to_UCCnetGBO_envelope<br/> _publicationCommand_CIP</code></p>                                                                                                                                     |                                                                            |
| FAILURE_EXCEPTIONS          | Specifies exception message numbers that would cause the collaboration object to fail flow                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Yes, if a particular exception number should cause a failed flow situation |

### Creating and configuring a UCCnet\_requestWorklist collaboration object and making its port connections

1. Create and configure the following collaboration object based on the UCCnet\_requestWorklist collaboration template.

Table 48. Collaboration object name: UCCnet\_requestWorklistObj

| 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 49. UCCnet\_requestWorklistObj collaboration object configuration properties

| Property name  | Property description                                                                                                                                                                                                                                                                                                   | Required                                                                                           |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| SET_UNIQUE_IDS | Controls whether unique IDs (messageIdentifier and uniqueCreateorIdentification) are set in the output XML messages. Possible values for this property are:<br><br><b>ALL</b> Set all three unique IDs. The default value.<br><b>NONE</b> No unique IDs<br><b>BLANK</b> Set unique ID only if it is blank in the input | No, unless the default value needs to be changed.                                                  |
| DTD_URL        | Value to be copied into the DTD xml DOCTYPE attribute of the output UCCnetGBO_envelope. The default value is DOCTYPE envelope SYSTEM "http://www.uccnet.net/xmlschema/2.2/Envelope.dtd"                                                                                                                                | Yes, if configured for UCCnet DTD message format. No, if configured for UCCnet XSD message format. |

### Creating and configuring a UCCnet\_processWorklist collaboration object and making its port connections

1. Create and configure the following collaboration object based on the UCCnet\_processWorklist collaboration template:

Table 50. Collaboration object name: UCCnet\_processWorklistObj

| Port                | Type      | Bind to                        |
|---------------------|-----------|--------------------------------|
| From                | Connector | AS2 channel connector          |
| To                  | Connector | Supplier's connector of choice |
| Failure             | Connector | Supplier's connector of choice |
| UnsolicitedResponse | Connector | Supplier's connector of choice |

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

Table 51. UCCnet\_processWorklistObj collaboration object configuration properties

| Property name      | Property description                                                                                                                                                                                                                                                                                                                                            | Required                                                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| SEND_EMAIL_TO      | Defines the e-mail address to which problems detected during execution of the collaboration object are sent. This entry is for InterChange Server administrators. Do not confuse this property with the <b>Email notification address</b> field, which is configured from the <b>Collaboration General Properties</b> tab when creating a collaboration object. | No. Requires the Email Connector and a valid entry in this field to send e-mail. If this field is left blank, no e-mail is sent. |
| DB_CONN_POOL_NAME  | The name of the connection data pool (default value = ITEMSYNC_DATA)                                                                                                                                                                                                                                                                                            | Yes                                                                                                                              |
| BATCH_RESPONSE     | Indicates whether the collaboration should send all CatalogueItemConfirmation notifications in one batch to the To port or individually. Default = true.                                                                                                                                                                                                        | Yes                                                                                                                              |
| FAILURE_EXCEPTIONS | Specifies exception message numbers that would cause the collaboration object to fail flow                                                                                                                                                                                                                                                                      | Yes, if a particular exception number should cause a failed flow situation                                                       |

Table 51. UCCnet\_processWorklistObj collaboration object configuration properties (continued)

| Property name | Property description                                                               | Required                                                                                                 |
|---------------|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| SEND_EMAIL    | Specifies whether e-mail is sent to the address set in the SEND_EMAIL_TO property. | Yes, if you want to send e-mail. To send e-mail, set the value to all. Otherwise, set the value to none. |

## Creating and configuring a UCCnet\_Price collaboration object and making its port connections

Do the following steps:

1. Create and configure the following collaboration object based on the UCCnet\_Price collaboration template:

Table 52. Collaboration object name: UCCnet\_Price

| Port                   | Type      | Bind to                      |
|------------------------|-----------|------------------------------|
| From                   | Connector | Supplier choice of connector |
| To                     | Connector | AS2 channel connector        |
| Error                  | Connector | Supplier choice of connector |
| DestinationAppRetrieve | Connector | PortConnector                |

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

Table 53. UCCnet\_Price collaboration object configuration properties

| Property name      | Property description                                                                                                                                                                                                                                                                                                                                            | Required                                                                                                 |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| SEND_EMAIL_TO      | Defines the e-mail address to which problems detected during execution of the collaboration object are sent. This entry is for InterChange Server administrators. Do not confuse this property with the <b>Email notification address</b> field, which is configured from the <b>Collaboration General Properties</b> tab when creating a collaboration object. | No. Requires the Email Connector to run if email is entered. The SEND_EMAIL property must be set also.   |
| SEND_EMAIL         | Specifies whether e-mail is sent to the address set in the SEND_EMAIL_TO property.                                                                                                                                                                                                                                                                              | Yes, if you want to send e-mail. To send e-mail, set the value to all. Otherwise, set the value to none. |
| DB_CONN_POOL_NAME  | The name of the connection data pool (default value = ITEMSYNC_DATA)                                                                                                                                                                                                                                                                                            | Yes                                                                                                      |
| PERFORM_VALIDATION | Specifies whether to check Price Documents for a gtn, seller gln, and target market country code to validate if the item has already been registered.<br><br>For PriceBracket documents, checks for a seller gln value.                                                                                                                                         | Yes, if you want to perform this check                                                                   |
| FAILURE_EXCEPTIONS | Specifies exception message numbers that would cause the collaboration object to fail flow                                                                                                                                                                                                                                                                      | Yes, if a particular exception number should cause a failed flow situation                               |



## Creating and configuring Notify\_by\_eMail collaboration objects and making their port connections

You can receive responses and error messages as e-mail by using the Notify\_byEmail template. The following is an example of creating a collaboration object based on the Notify\_by\_eMail template. In this example, a collaboration object is configured for use with the UCCnet\_processWorklist Failure port

1. Create and configure the following collaboration objects based on the Notify\_by\_eMail collaboration template:

Table 54. Collaboration object name: UPW\_FAILURE\_RESPONSEObj

| Port     | Type                 | Bind to                           |
|----------|----------------------|-----------------------------------|
| From     | Collaboration object | UCCnet_processWorklistObj:Failure |
| FromItem | Connector            | PortConnector                     |

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

Table 55. Notify\_by\_eMail collaboration object configuration properties. Use the values in this table for the UPW\_FAILURE\_RESPONSEObj collaboration object.

| Property name            | Property description                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Required |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| EMAIL_MESSAGE            | Body of the e-mail. The processing of this field uses the same variables as the EMAIL_SUBJECT property. It also processes values beginning with the @ sign in the same manner. For example, in the following, the first character of the string is an @ sign, so the collaboration object loads the text from the filename following the @ sign:<br><code>@c:\IBM\WebSphereICS\UCCnet\collaborations\eMail\ \ UCCnet_processWorklist_AUTHORIZATION_RESPONSES.mail</code> | Yes      |
| EMAIL_NOTIFICATION_RCPTS | E-mail address of the recipients.                                                                                                                                                                                                                                                                                                                                                                                                                                        | Yes      |
| EMAIL_SUBJECT            | Subject line of the e-mail. This value can contain variables in the form <code>\${variable_name}</code> into which the collaboration object substitutes data from the business object dynamically. See "E-mail examples."                                                                                                                                                                                                                                                | Yes      |

**Note:** Several sample e-mail files are in the following directory:

### Windows operating systems

`<ICS_installation_path>\Samples\UCCnet\collaborations\eMail`

### UNIX operating systems

You must FTP (ASCII mode) the e-mail files to a directory on your UNIX system.

Each of the collaboration objects based on the Notify\_by\_eMail collaboration template must have its properties updated to point to the correct location of these sample files or to your own custom e-mail files.

**E-mail examples:** Type text using the following escape codes:

- `${getRoot}` - substitutes the entire triggering business object.
- `${getDate}` - substitutes the current date and time.
- `${getName}` - substitutes the name of the triggering business object.
- `${getVerb}` - substitutes the verb of the triggering business object.

- `${attribute}` - substitutes the value of the named attribute from the triggering business object. If the value for the *variable\_name* does not match one of the specific values above, the collaboration object interprets it as the name of a business object attribute.

If the first character of the string for this value is an @ sign, the collaboration object loads the text from a filename following the @ sign.

See the Solution development guide for information about how the solution handles sending e-mail and for a sample value for this property.

## 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 lets you choose what parts of the solution to deploy. Deploy the solution in stages, as follows:
  - a. Deploy the business objects.
  - b. Deploy the maps. Ensure that you have checked the **Compile** option.
  - c. Deploy the connectors.
  - d. Deploy the collaboration templates. Ensure that you have checked the **Compile** option.
  - e. Deploy the database connection pool.
  - f. Restart the WebSphere InterChange Server.
  - g. Deploy the collaboration objects.

---

## 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.