

*IBM WebSphere Business Integration Collaborations
for UCCnet Item Synchronization Version 4.3.1
IBM WebSphere Business Integration Collaborations
Version 4.5*



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 47.

Sixth Edition (December 2003)

This edition applies to

Version 4, Release 3, Modification 1 of the *IBM WebSphere Business Integration Collaborations for UCCnet Item Synchronization* (5724-H62)

Version 4, Release 5, of the *IBM WebSphere Business Integration Collaborations* (5724-C12)

and to all subsequent releases and modifications until otherwise indicated in new editions.

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

IBM Canada Ltd. Laboratory
Information Development
8200 Warden Avenue
Markham, Ontario, Canada L6G 1C7

Include the title and order number 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, 2003. 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	3
Hardware prerequisites	4
Software prerequisites	4
Installing and configuring the solution.	5
Planning the configuration.	6
Configuring the system environment	6
Installing the components	8
Creating the audit_log table	8
Configuring the business objects	10

Creating and configuring the connectors.	14
Creating and configuring the collaboration objects	32
Configuring the relationships before deployment	43
Deploying the solution	44
Populating the relationships after deployment.	44

Notices and Trademarks 47

Notices	47
Programming interface information	48
Trademarks and service marks	49

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 will 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 UCCnet® 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.

Table 1. Collaboration templates

Name	Location
UCCnet_ItemSync	BIA_Retail_CT_ItemSync.jar repository file
UCCnet_requestWorklist	BIA_Retail_CT_ItemSync.jar repository file
UCCnet_processWorklist	BIA_Retail_CT_ItemSync.jar repository file
Notify_by_eMail	BIA_Retail_CT_ItemSync.jar repository file
ItemCommandRouter	BIA_Retail_CT_ItemSync.jar repository file
CIN_CIP_Dispatcher	BIA_Retail_CT_ItemSync.jar repository file

Table 2. Business objects

Name	Location
UCCnetGBO_envelope	BIA_Retail_BO_ItemSync.jar repository file
UCCnetDTD_envelope	BIA_Retail_BO_ItemSync.jar repository file
UCCnetXSD_envelope	BIA_Retail_BO_ItemSync.jar repository file
UCCnetTPIDTD_envelope	BIA_Retail_BO_ItemSync.jar repository file
UCCnetTPIXSD_envelope	BIA_Retail_BO_ItemSync.jar repository file
SAP4_MatlBasic	BIA_Retail_BO_ItemSyncSamples.jar repository file

Table 3. Maps

Name	Location
CwItemBasic_to_UCCnetTPIDTD_envelope_documentCommand_item	BIA_Retail_NM_ItemSync.jar repository file
CwItemBasic_to_UCCnetGBO_envelope_notifyCommand_catalogueItem	BIA_Retail_NM_ItemSync.jar repository file
CwItemBasic_to_UCCnetXSD_envelope_publicationCommand_catalogueItem	BIA_Retail_NM_ItemSync.jar repository file
CwItemBasic_to_UCCnetTPIXSD_envelope_publicationCommand_catalogueItem	BIA_Retail_NM_ItemSync.jar repository file
CwItemBasic_to_UCCnetXSD_envelope_publicationCommand_catalogueItem	BIA_Retail_NM_ItemSync.jar repository file
CwItemBasic_to_UCCnetTPIXSD_envelope_publicationCommand_catalogueItem	BIA_Retail_NM_ItemSync.jar repository file
CwItemBasic_to_UCCnetGBO_env_publicationCommand_catalogueItemPublication	BIA_Retail_NM_ItemSync.jar repository file
CwItemBasic_to_UCCnetDTD_envelope_documentCommand_item	BIA_Retail_NM_ItemSync.jar repository file
CwItemBasic_to_UCCnetDTD_envelope_publishCommand_documentIdentifier	BIA_Retail_NM_ItemSync.jar repository file
CwItemBasic_to_UCCnetTPIDTD_envelope_publishCommand_documentIdentifier	BIA_Retail_NM_ItemSync.jar repository file
CwItemBasic_to_UCCnetXSD_envelope_registerCommand_itemAddChange	BIA_Retail_NM_ItemSync.jar repository file
CwItemBasic_to_UCCnetTPIXSD_envelope_registerCommand_itemAddChange	BIA_Retail_NM_ItemSync.jar repository file
RouterMap_CwItemBasic_to_UCCnetDTD_envelope	BIA_Retail_NM_ItemSync.jar repository file
RouterMap_CwItemBasic_to_UCCnetTPIDTD_envelope	BIA_Retail_NM_ItemSync.jar repository file
RouterMap_UCCnetGBO_envelope_to_UCCnetDTD_envelope	BIA_Retail_NM_ItemSync.jar repository file
RouterMap_UCCnetGBO_envelope_to_UCCnetTPIDTD_envelope	BIA_Retail_NM_ItemSync.jar repository file
UCCnetGBO_envelope_to_UCCnetDTD_envelope	BIA_Retail_NM_ItemSync.jar repository file
UCCnetGBO_envelope_to_UCCnetTPIDTD_envelope	BIA_Retail_NM_ItemSync.jar repository file
UCCnetGBO_envelope_to_UCCnetXSD_envelope	BIA_Retail_NM_ItemSync.jar repository file
UCCnetGBO_envelope_to_UCCnetTPIXSD_envelope	BIA_Retail_NM_ItemSync.jar repository file
UCCnetDTD_envelope_to_UCCnetGBO_envelope	BIA_Retail_NM_ItemSync.jar repository file
UCCnetTPIDTD_envelope_to_UCCnetGBO_envelope	BIA_Retail_NM_ItemSync.jar repository file
UCCnetXSD_envelope_to_UCCnetGBO_envelope	BIA_Retail_NM_ItemSync.jar repository file
UCCnetTPIXSD_envelope_to_UCCnetGBO_envelope	BIA_Retail_NM_ItemSync.jar repository file
UCCnetGBO_envelope_notification_to_UCCnetDTD_envelope_publishCommand	BIA_Retail_NM_ItemSync.jar repository file
UCCnetGBO_envelope_notification_to_UCCnetTPIDTD_envelope_publishCommand	BIA_Retail_NM_ItemSync.jar repository file
CwSa4ItemBasic	BIA_Retail_NM_ItemSyncSamples.jar repository file
Sa4CwItemBasic	BIA_Retail_NM_ItemSyncSamples.jar repository file

Table 4. Relationships

Name	Location
GTIN	BIA_Retail_REL_ItemSync.jar repository file
LiUT	BIA_Retail_REL_ItemSync.jar repository file
PrTyNT	BIA_Retail_REL_ItemSync.jar repository file
SiUT	BIA_Retail_REL_ItemSync.jar repository file
TPTable	BIA_Retail_REL_ItemSync.jar repository file
UDEX	BIA_Retail_REL_ItemSync.jar repository file
UpcTyT	BIA_Retail_REL_ItemSync.jar repository file
VoUT	BIA_Retail_REL_ItemSync.jar repository file
WeUT	BIA_Retail_REL_ItemSync.jar repository file

Table 5. Repository files

Name
BIA_Retail_BO_ItemSync.jar
BIA_Retail_BO_ItemSyncSamples.jar
BIA_Retail_CT_ItemSync.jar

Table 5. Repository files (continued)

Name
BIA_Retail_NM_ItemSync.jar
BIA_Retail_NM_ItemSyncSamples.jar
BIA_Retail_REL_ItemSync.jar

Table 6. Database scripts for DB2®, Microsoft® SQL Server, and Oracle

Name
audit_log.sql (one copy supplied for each database type)
InitializeRelationshipTables.sql (one copy supplied for each database type)
InitializeRelationshipTablesForXSD.sql (one copy supplied for each database type)

Table 7. Library file for collaboration templates

Name
CSIAuditLogger.jar

Table 8. InterchangeSystem.UCCnet file

Name
InterchangeSystem.UCCnet

Table 9. Email sample files

Name
UCCnet_processWorklist_AUTHORIZATION_RESPONSES.mail
UCCnet_processWorklist_CATEGORY_ADD_CHANGE.mail
UCCnet_processWorklist_CATEGORY_ADD_CHANGESubject.mail
UCCnet_processWorklist_CATALOGUE_ITEM_CONFIRMATION.mail
UCCnet_processWorklist_DEAD_LETTER_PUB_RECEIPT.mail
UCCnet_processWorklist_SIMPLE_RESPONSE.mail
UCCnet_processWorklist_UNKNOWN_MESSAGES.mail
UCCnet_processWorklist_UNKNOWN_RESPONSE.mail
UCCnet_processWorklist_CIN_RESPONSE.mail
UCCnet_processWorklist_CI_RESPONSE.mail
UCCnet_processWorklist_CIP_RESPONSE.mail
UCCnet_processWorklist_RCIR_QUERY_RESPONSE.mail
UCCnet_processWorklist_FAILURE_RESPONSE.mail
ItemCommandRouter_DE_LIST_WITHDRAW.mail

Supported operating environments

The following operating environments are supported in this release:

- Windows 2000 (Professional, Server, or Advanced Server) with Service Pack 3
- 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 52.9 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 V4.2.1. Ensure that you select the **IBM WebSphere Business Integration Collaboration for Automotive and Manufacturing - Item Manager** during installation of IBM WebSphere InterChange Server 4.2.1. The Item Manager collaboration contains the ItemBasic business object, which is used by the Item Synchronization for Suppliers solution.

Note:

The IBM WebSphere Interchange Server installer includes Retail collaborations, but the version included may 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 V4.2.1
- IBM DB2 V8.1 (Oracle and Microsoft SQL Server databases are also supported)

Note: When using DB2, a C compiler must be installed on the same server on which the ICS resides so that the stored procedures can be compiled during the relationship deployment. This is not required for Microsoft SQL Server or Oracle databases.

- IBM WebSphere MQ V5.3.0.2 with CSD 3
- IBM WebSphere Business Integration Data Handler for XML V2.4.1

Note: Refer to section “Configuring the system environment” on page 6 for additional setup information for the IBM WebSphere Business Integration Data Handler for XML V2.4.1

- IBM WebSphere Business Integration Adapters V2.3.1, which include the following:
 - IBM WebSphere Business Integration Adapter for e-Mail V5.1.0
 - IBM WebSphere Business Integration Adapter for JText V2.3.1

- IBM WebSphere Business Integration Adapter for Trading Partner Interchange V3.3.0
- IBM WebSphere Business Integration Adapter for iSoft V1.2.0
- Any specific ERP to be used. (the sample configuration in this document uses the IBM WebSphere Business Integration Adapter for mySAP.com, SAP R/3 V. 4.X)

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. It is recommended that you proceed through the sections in order:

1. “Planning the configuration” on page 6 details the set up decisions that must be made regarding connectivity and messaging method.
2. “Configuring the system environment” on page 6 details how to set up the system environment to run the Item Synchronization for Suppliers on all supported platforms.
3. “Installing the components” on page 8 instructs how to load artifacts and install them in the ICS repository, ensuring that the solution code is accessible to your system.
4. “Creating the audit_log table” on page 8 describes how to create the audit_log table, a relational table provided with the solution.
5. “Configuring the business objects” on page 10 instructs how to create and configure the business objects required to properly process UCCnet messages.
6. “Creating and configuring the connectors” on page 14 describes how to create and configure the connectors needed to operate the Item Synchronization for Suppliers solution.

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.

7. “Creating and configuring the collaboration objects” on page 32 details the collaboration objects that must be created from the collaboration templates, and how to set each collaboration object’s port connections and configuration properties.
8. “Configuring the relationships before deployment” on page 43 describes the database configuration entries needed for each relationship.

9. “Deploying the solution” on page 44 details how to deploy the solution.
10. “Populating the relationships after deployment” on page 44 instructs how to populate the relationship table with default values after system deployment.

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 will determine the connector that you use to communicate with it. If you exchange messages with UCCnet using an AS2/EDIINT interface protocol, you can use either a TPIConnector or an IsoftConnector. Use the TPI connector if you communicate with UCCnet through Trading Partner Interchange servers. Use the IsoftConnector if you are communicating with UCCnet through an iSoft Peer-to-Peer Agent. If you exchange messages through the UCCnet Command Line Utility (CLU) or are testing your installation, you can use either a JTextTPIConnector, or a JTextIsoftConnector.

Since the actual connector you use is dependent on your set up, the term “AS2 channel connector” is used throughout this documentation as a general term for any of the TPIConnector, iSoftConnector, JTextTPIConnector, and JTextIsoftConnector.

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 following options are available:

DTD message format

The format of the XML documents exchanged with UCCnet is defined by a Document Type Definition (DTD). The DTD mode of operation has one protocol available.

Schema message format

The format of the XML documents exchanged with UCCnet is defined by an XML Schema Definition (XSD). The XSD mode of operation has two command protocols available:

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 DATAHANDLER path, as follows:

Windows

- a. Edit the `<WebSphereICS_Installation_dir>\bin\start_server.bat` file to ensure that the beginning of the DATAHANDLER path includes the `CwXMLDataHandler.jar` file, as shown in the following example
`DATAHANDLER=%CROSSWORLDS%\DataHandlers\CwXMLDataHandler.jar`
- b. Edit the `WebSphereICS_Installation_dir\bin\CWConnEnv.bat` file to ensure that the beginning of the DATAHANDLER path includes the `CwXMLDataHandler.jar` file, as shown in the following example:
`DATAHANDLER=%CROSSWORLDS%\DataHandlers\CwXMLDataHandler.jar`

UNIX

- a. Edit the `WebSphereICS_Installation_dir/bin/CWSharedEnv.sh` file and alter the DATAHANDLER path to include the `CwXMLDataHandler.jar` file, as shown in the following example:
`DATAHANDLER=${CWCLASSES}:${CROSSWORLDS}/DataHandlers/CwXMLDataHandler.jar`
 - b. Edit the `WebSphereICS_Installation_dir/bin/CWConnEnv.sh` file and alter the DATAHANDLER path to include the `CwXMLDataHandler.jar` file, as shown in the following example:
`DATAHANDLER=${CWCLASSES}:${CROSSWORLDS}/DataHandlers/CwXMLDataHandler.jar`
2. Edit the CWCLASSES path, as follows:
 - **Windows:** Edit the `<WebSphereICS_installation_dir>\bin\start_server.bat` file by appending the end of the CWCLASSES path to include the `CSIAuditLogger.jar` file, as shown in the following example:
`CWCLASSES=...;%CROSSWORLDS%\lib\CSIAuditLogger.jar`
 - **UNIX:** Edit the `<WebSphereICS_installation_dir>/bin/CWSharedEnv.sh` file and alter the CWCLASSES path to include the `CSIAuditLogger.jar` file, as shown in the following example:
`CWCLASSES=${CWCLASSES}:${CROSSWORLDS}/lib/CSIAuditLogger.jar`
- Note:** The `CSIAuditLogger.jar` file must be copied from the Windows system to the UNIX system.
3. 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 `CSIAuditLogger.jar` file and its appropriate path information to the line `classpath=` in this file, as in the following example:
`classpath=<directory_location_of_CSIAuditLogger.jar_file>CSIAuditLogger.jar`
 4. Configure the ICS with email information by doing the following:
 - a. Open the System Manager.
 - b. Connect to the ICS.
 - c. Right-click on the ICS name and select **Edit Configuration**.
 - d. On the **E-mail** tab, select **Connector mail** from the **E-mail send type** menu.
 - e. Close the Edit Configuration window and save it when prompted.

Note: The EmailConnector requires that the E-mail collaboration template be active on the server. The E-mail collaboration template is installed by default during installation of the ICS, and must always be active, although it might not appear in any Component Library, and does not appear on the System Monitor screen. If it is necessary to drop the ICS 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:

- **Windows:**

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

- **UNIX:**

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

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

5. Stop the ICS.

Installing the components

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

1. Start the ICS in design mode (-design parameter).
2. Install the artifacts downloaded from www.lotus.com/passportadvantage into the same directory in which the ICS is installed by performing the following steps (see the section “Obtaining solution components” on page 1 for information on how to obtain the artifacts from Passport Advantage):
 - a. Move the downloaded .exe file to a temporary directory.
 - b. Run the executable file. The following are extracted:
 - media.inf
 - setup.jar
 - setupwin32.exe (installer executable file for Windows operating systems)
 - an English readme file and translated versions of it
 - c. Run the executable file to install the components.
3. Verify that the CSIAuditLogger.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 ICS.
4. On UNIX systems, transfer the file `<WebSphereICS_installation_dir>\lib\CSIAuditLogger.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. Use the Browse button to navigate to where the repository .jar files are located and select one of the files.
 - d. Click Open.
 - e. Click Finish.
 - f. Repeat this process for each of the remaining repository files.

Creating the audit_log table

To create the audit_log table, complete the following steps. Be sure to follow the instructions appropriate for your platform where indicated.

1. Create the audit_log table by running one of the following scripts, depending on the database you are using:
 - **Windows:**

- DB2 —
 - a. Start a DB2 session by opening a DB2 command window by clicking **Start > Programs > IBMDB2 > Command Line Tools > Command Window**.
 - b. Move to the following directory:
`<ICS_installation_path>\collaborations\dependencies\db2`
 - c. At the prompt, enter the following command: **db2 connect to <repository_name> USER <userid> USING <password>** where `<repository_name>` is the name of your WebSphere InterChange Server repository database, and `<userid>` and `<password>` are the ID and password that were used to create it.
 - d. After you are connected, enter the command **db2 < audit_log.sql** to run the script. You will receive error messages from the DROP commands in the script. This is expected and the errors can be ignored.
- Oracle —
 - a. Move to the following directory:
`<ICS_installation_path>\collaborations\dependencies\ \ UCCnet\oracle`
 - b. Enter the following command:
`sqlplus dbuser/passwd@SID @audit_log.sql`

You might have to enable autoextension of the CWRepository datafile. Select the **AUTOEXTEND** option on the **Storage** tab for the datafile in the DBA Studio.
- Microsoft SQL Server —
 - a. Move to the following directory:
`<ICS_installation_path>\collaborations\dependencies\ \ UCCnet\mss`
 - b. Select the CWRepository database in the drop-down in the menu bar.
 - c. Execute the audit_log.sql script by using the Query Analyzer tool.
- **UNIX:**
 - DB2 —
 - a. Transfer the file
`<ICS_installation_path>\collaborations\dependencies\UCCnet\db2 \audit_log.sql` to the UNIX system using FTP in ASCII mode. Place the file in the
`<ICS_installation_path>/collaborations/dependencies/UCCnet/db2` directory.
 - b. Move to the following directory of the UNIX system:
`<ICS_installation_path>/collaborations/dependencies/ \ UCCnet/db2`
 - c. Start a DB2 session by entering the command **db2**.
 - d. At the prompt, enter the command **connect to repository_name**.
 - e. After you are connected, enter the command **< audit_log.sql** to run the script.
 - Oracle —
 - a. Transfer the file
`<ICS_installation_path>\collaborations\dependencies\UCCnet\oracle \audit_log.sql` to the UNIX system using FTP in ASCII mode. Place

the file in the

`<ICS_installation_path>/collaborations/dependencies/UCCnet/oracle` directory.

- b. Move to the following directory of the UNIX system:

`<ICS_installation_path>/collaborations/dependencies/ \`
`UCCnet/oracle`

- c. Enter the following command:

```
sqlplus dbuser/passwd@SID @audit_log.sql
```

You might have to enable autoextension of the CWRepository datafile. Select the **AUTOEXTEND** option on the **Storage** tab for the datafile in the DBA Studio.

2. The UCCnet property provides the collaboration objects with the information they require to access the audit_log table. Add the UCCnet property to the InterchangeSystem.cfg file by doing the following.

- **Windows:**

- a. Move to the directory `<ICS_installation_path>\samples\UCCnet`.
- b. Copy the contents of the InterchangeSystem.UCCnet file
- c. Move to the directory `<ICS_installation_path>`.
- d. Paste the UCCnet property into the InterchangeSystem.cfg file, just before the `cw:ServerConfig` tag at the end of the file.
- e. Modify the values for the following UCCnet properties to match your repository database installation:
 - USER_NAME
 - DRIVER
 - PASSWORD
 - DATA_SOURCE_NAME

- **UNIX:**

- a. Copy the InterchangeSystem.UCCnet from the Windows system to the UNIX system.
- b. Move to the directory `<ICS_installation_path>/UCCnet`.
- c. Copy the contents of the InterchangeSystem.UCCnet file
- d. Move to the directory `<ICS_installation_path>`.
- e. Paste the UCCnet property into the InterchangeSystem.cfg file, just before the `cw:ServerConfig` tag at the end of the file.
- f. Modify the values for the following UCCnet properties to match your repository database installation:
 - USER_NAME
 - DRIVER
 - PASSWORD
 - DATA_SOURCE_NAME
- g. Modify the TARGETHLQ property so that it points to a local directory.

Configuring the business objects

You must configure (and in some cases create) the following business objects 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 10. Selected attribute values for MO_DataHandler_UCCnetXMLConfig metaobject

Attribute name	Column	Setting
BOPrefix	Default	<ul style="list-style-type: none"> • UCCnetDTD (for iSoft connectivity using the DTD XML definition) • UCCnetXSD (for iSoft connectivity using XSD XML definition) • UCCnetTPIDTD (for TPI connectivity using DTD XML definition) • UCCnetTPIXSD (for TPI connectivity using XSD XML definition)
DTDPath	Default	<p>DTD support Path to Envelope.dtd file. For example, <ICS_installation_path>\UCCnet\DTDs\2.2\Envelope.dtd</p> <p>XSD support Path to Envelope.xsd file. For example, <ICS_installation_path>\UCCnet\XSDs\uccnet\2.2\Envelope.xsd</p> <p>Note: The value of this attribute assumes use of the UCCnet 2.2 DTD. The attribute values shown are examples only. The actual value must be the fully qualified path to the Envelope.dtd or Envelope.xsd file on your system. The Envelope files are available from the UCCnet eRoom. You must have a valid UCCnet eRoom user ID and password to obtain them.</p>
Validation	Default	false
DefaultEscapeBehavior	Default	true
IgnoreUndefinedElements	Default	true

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

Table 11. 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 12. Selected attribute values for MO_JTextRWLConnector_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\JTextRWL\out). Create this directory if it does not already exist.
OutputExt	Default	xml
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.
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
EndBODelimiter	Default	EOF

4. If you are exchanging messages through the UCCnet CLU or testing your installation, edit the MO_JTextRWLConnector_Default metaobject in one of the following ways, depending on the connectivity type used:

iSoft connectivity

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

Table 13. Selected attribute values for MO_JTextISoftConnector_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\JTextISoft\out). Create this directory if it does not already exist.
OutputExt	Default	xml
ArchiveDir	Default	<Name of the directory where archive XML files will be written> (for example, C:\IBM\WebSphereICS\UCCnet\JTextISoft\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\JTextISoft\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 functionality, you must create a second instance of both the JTextISoft connector and the JTextISoft connector metaobjects. Edit the MO_JTextISoftConnector_Default by setting the following attributes, then save it as MO_JTextISoft2Connector_Default.

Table 14. Selected attribute values for MO_JTextISoft2Connector_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\JTextISoft2\out). Create this directory if it does not already exist.
OutputExt	Default	xml
ArchiveDir	Default	<Name of the directory where archive XML files will be written> (for example, C:\IBM\WebSphereICS\UCCnet\JTextISoft2\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\JTextISoft2\event). Create this directory if it does not already exist.
EventExt	Default	xml
EndBODelimiter	Default	EOF

TPI connectivity

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

Table 15. Selected attribute values for MO_JTextTPIConnector_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\JTextTPI\out). Create this directory if it does not already exist.
OutputExt	Default	xml
ArchiveDir	Default	<Name of the directory where archive XML files will be written> (for example, C:\IBM\WebSphereICS\UCCnet\JTextTPI\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\JTextTPI\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 functionality, you must create a second instance of both the JTextTPI connector and the JTextTPI connector metaobjects. Edit the MO_JTextTPIConnector_Default by setting the following attributes, then save it as MO_JTextTPI2Connector_Default.

Table 16. Selected attribute values for MO_JTextTPI2Connector_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\JTextTPI2\out). Create this directory if it does not already exist.
OutputExt	Default	xml
ArchiveDir	Default	<Name of the directory where archive XML files will be written> (for example, C:\IBM\WebSphereICS\UCCnet\JTextTPI2\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\JTextTPI2\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, do one of the following, depending on the connectivity type used:
 - If iSoft connectivity is used, 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 will need two instances of the iSoft connector. However, both instances use the same metaobject, so only one metaobject is needed.

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

Table 17. Attribute values for MO_IsoftAdapterConfig

Attribute name	Type	Key	Application Specific Information
Default	String	x	OutputQueue=queue:// my_p2p_agent_queue_manager_name/ my_outbox_queue_name; DataEncoding=Text
UCCnetDTD_envelope_Create (for DTD support) or 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.

- If TPI connectivity is used, no configuration metaobject is required.

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

Table 18. Selected attribute values for EmailNotification business object

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

Creating and configuring the connectors

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

- Create and configure the JTextRWLConnector in every installation, as detailed in the section “Creating and configuring the JTextRWLConnector” on page 15.
- Configure, or if necessary, create and configure one of the following connectivity connectors depending on the connectivity type you are using and the protocol you are using to exchange messages:
 - If you are exchanging messages with UCCnet through an AS2/EDIINT interface protocol and are using iSoft connectivity, configure the iSoftConnector, as detailed in the section “Creating and configuring the iSoftConnector” on page 17.
 - If you are exchanging messages with UCCnet through an AS2/EDIINT interface protocol and are using TPI connectivity, configure the TPIConnector, as detailed in the section “Creating and configuring the TPIConnector” on page 20.
 - If you are exchanging messages through the UCCnet CLU or testing your installation, and are using iSoft connectivity, create and configure the JTextISoftConnector, as detailed in the section “Creating and configuring the JTextISoftConnector” on page 24.
 - If you are exchanging messages through the UCCnet CLU or testing your installation, and are using TPI connectivity, create and configure the JTextTPIConnector, as detailed in the section “Creating and configuring the JTextTPIConnector” on page 27.
- Configure the PortConnector in every installation, as detailed in the section “Configuring the PortConnector” on page 30.
- If you want to use the email capabilities of the Item Synchronization for Suppliers solution, configure the EmailConnector, as detailed in the section “Configuring the EmailConnector” on page 30.

- Configure the connector for the source of your item information. The SAPConnector is included here as an example, as detailed in the section “Configuring the SAPConnector” on page 31. If you want to test the solution without a live connection to the source of your item information, configure a TestConnector. A sample TestConnector configuration is detailed in the section “Configuring the TestConnector” on page 31.

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 19. Standard property

Property name	Value
ApplicationName	JTextRWLConnector

Table 20. 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"> • UCCnetDTD_envelope (when iSoft connectivity and DTD XML definition type are used) • UCCnetTPIDTD_envelope (when TPI connectivity and DTD XML definition type are used) • UCCnetXSD_envelope (when iSoft connectivity and XSD XML definition type are used) • UCCnetTPIXSD_envelope (when TPI connectivity and XSD XML definition type are used) 	Yes

Table 20. Supported business objects (continued)

Business object name	Agent support required?
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 21. Associated maps

Business object name	Map name
UCCnetDTD_envelope (when iSoft connectivity and DTD XML definition type are used)	UCCnetDTD_envelope_to_UCCnetGBO_envelope
UCCnetTPIDTD_envelope (when TPI connectivity and DTD XML definition type are used)	UCCnetTPIDTD_envelope_to_UCCnetGBO_envelope
UCCnetXSD_envelope (when iSoft connectivity and XSD XML definition type are used)	UCCnetXSD_envelope_to_UCCnetGBO_envelope
UCCnetTPIXSD_envelope (when TPI connectivity and XSD XML definition type are used)	UCCnetTPIXSD_envelope_to_UCCnetGBO_envelope
UCCnetGBO_envelope	<ul style="list-style-type: none"> • When iSoft connectivity and DTD XML definition type are used: RouterMap_UCCnetGBO_envelope_to_UCCnetDTD_envelope • When TPI connectivity and DTD XML definition type are used: RouterMap_UCCnetGBO_envelope_to_UCCnetTPIDTD_envelope • When iSoft connectivity and XSD XML definition type are used: UCCnetGBO_envelope_to_UCCnetXSD_envelope • When TPI connectivity and XSD XML definition type are used: UCCnetGBO_envelope_to_UCCnetTPIXSD_envelope

3. Save the configuration file (**File > Save > To Project**), then close the Connector Configurator.
4. Create the JTextRWLConnector Agent, as follows:
 - **Windows:**
 - a. In the Windows taskbar, right-click **Start** and select **Open All Users**.
 - b. Navigate to the folder on your system that contains the installed connectors by clicking **Programs > IBM WebSphere Business Integration Adapters > Adapters > Connectors**.
 - c. Copy the JText Connector short cut and rename it to JTextRWL Connector.
 - d. Right-click the JTextRWL Connector short cut and select **Properties**.
 - e. Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to JTextRWL, where *ICS_server_name* refers to the name of the ICS server, as shown in the following example:


```
<install_path>\IBM\WebSphereICS\connectors\JText\start_JText.bat \
JTextRWL ICS_server_name
```
 - **UNIX:**
 - a. Access the JTextConnector Agent program located in the following directory: *<install_path>/IBM/WebSphereICS/connectors/JText/*.
 - b. Run JTextRWL by switching to this directory and entering the following command, where *ICS_server_name* refers to the name of the ICS server:

```
start_JText.sh 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 ICS and *ICS_server_name* refers to the name of the ICS 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 instance, 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 22. Connector-specific properties

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

Table 23. Supported business objects

Business object name	Agent support required?
MO_DataHandler_UCCnet_envelope	Yes
MO_IsoftAdapterConfig	Yes
UCCnetDTD_envelope (used only for DTD support)	Yes
UCCnetXSD_envelope (used only for XSD support)	
UCCnetGBO_envelope	No
ItemBasic	No

3. 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
UCCnetDTD_envelope (used only for DTD support)	UCCnetDTD_envelope_to_UCCnetGBO_envelope
UCCnetXSD_envelope (used only for XSD support)	UCCnetXSD_envelope_to_UCCnetGBO_envelope
UCCnetGBO_envelope	DTD support RouterMap_UCCnetGBO_envelope_to_UCCnetDTD_envelope XSD support UCCnetGBO_envelope_to_UCCnetXSD_envelope
ItemBasic	DTD support RouterMap_CwItemBasic_to_UCCnetDTD_envelope XSD support CwItemBasic_to_UCCnetXSD_envelope_registerCommand_itemAddChange

4. Save the configuration (**File > Save > To Project**) then close the Connector Configurator.
5. If you are using WebSphere MQ as your connector transport, create the iSoftConnector queues in WebSphere MQ. You must create the following queues as local queues and accept the defaults, where *local_WebSphere_ICS_queue_manager_name* refers to the queue manager used by the IBM WebSphere ICS and *ICS_server_name* refers to the name of the ICS 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. Perform the following steps to configure the second instance:

1. Open the System Manager.
2. Save the ISoft2Connector by doing the following:
 - a. Open the ISoftConnector.
 - b. Click **File > Save As > To Project**.
 - c. Save the connector as ISoft2Connector.

3. Create the ISoft2Connector Agent, as follows:
 - **Windows:**
 - a. In the Windows taskbar, 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 ICS server, as shown in the following example:


```
<install_path>\connectors\JText\start_ISoft.bat ISoft2 ICS_server_name
```
 - **UNIX:**
 - a. Access the ISoft2Connector Agent program located in the following directory: *<install_path>/IBM/WebSphereICS/connectors/ISoft/*.
 - b. Run ISoft2 by switching to this directory and entering the following command, where *ICS_server_name* refers to the name of the ICS server:


```
start_JText.sh ISoft2 ICS_server_name
```
4. 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 25. Connector-specific properties

Property name	Value
ArchiveQueue	Queue to which copies of successfully processed messages are sent (for instance, queue://my_p2p_agent_queue_manager_name/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 instance, 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 instance, 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 instance, 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 instance, queue://my_p2p_agent_queue_manager_name/unsubscribed2)

Table 25. Connector-specific properties (continued)

Property name	Value
UseDefaults	true (add this property if it does not appear in the list of connector-specific properties)

Table 26. Supported business objects

Business object name	Agent support required?
MO_DataHandler_UCCnet_envelope	Yes
MO_ISoftAdapterConfig	Yes
UCCnetXSD_envelope	Yes
ItemBasic	No

5. 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
ItemBasic	CwItemBasic_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 ICS and `ICS_server_name` refers to the name of the ICS 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 TPICConnector

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 will need two instances of the TPI connector. Otherwise, you only require one. Perform the following steps to configure the TPICConnector:

Creating and configuring the first instance of the TPICConnector:

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 instance, c:\TPISolo\).
3. Configure this connector to include the connector-specific configuration properties and appropriate business objects. Use the values shown in the following tables.

Table 28. Connector-specific properties

Property name	Value
ArchiveProcessedDocDir	Directory where processed document meta-events are archived (for instance, 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 instance, C:\TPISolo\data\uccnet2\xmlout).
MetaEventDir	Directory used to persist the TPI event information for recovery purposes (for instance, C:\TPISolo\data\uccnet2\xmlin).
PollQuantity	1 (add this property it does not appear in the list of connector-specific properties).
TradingPartnerConfigurationFile	Fully qualified name of the trading partner configuration file created in Step 1 above (for instance, C:\IBM\WebSphereICS\connectors\TPI\tpcfg.txt).
WaitForMDN	false (MDNs are not supported by this solution).

Table 29. Supported business objects

Business object name	Agent support required?
MO_DataHandler_UCCnet_envelope	Yes
<ul style="list-style-type: none"> • UCCnetTPIDTD_envelope (when the DTD XML definition type is used) • UCCnetTPIXSD_envelope (when the XSD XML definition type is used) 	Yes
UCCnetGBO_envelope	No
ItemBasic	No

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

Table 30. Associated maps

Business object name	Map name
UCCnetTPIDTD_envelope (used only for DTD support)	UCCnetTPIDTD_envelope_to_UCCnetGBO_envelope
UCCnetTPIXSD_envelope (used only for XSD support)	UCCnetTPIXSD_envelope_to_UCCnetGBO_envelope
UCCnetGBO_envelope	<p>DTD support RouterMap_UCCnetGBO_envelope_to_UCCnetTPIDTD_envelope</p> <p>XSD support UCCnetGBO_envelope_to_UCCnetTPIXSD_envelope</p>

Table 30. Associated maps (continued)

Business object name	Map name
ItemBasic	DTD support RouterMap_CwItemBasic_to_UCCnetTPIDTD_envelope XSD support CwItemBasic_to_UCCnetTPIXSD_envelope_registerCommand _itemAddChange

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 ICS and *ICS_server_name* refers to the name of the ICS 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
```

7. When a message is sent to UCCnet, the **SenderId** and **ReceiverId** values in the UCCnetTPIDTD_envelope and UCCnetTPIXSD_envelope business objects are set by the maps when these business objects are created. For the following maps, use the Map Designer to modify the "Set value" rules for the TPIRouteInfo child business object with the correct **SenderId** and **ReceiverId** values for your installation.

```
CwItemBasic_to_UCCnetGBO_envelope_notifyCommand_catalogueItem
CwItemBasic_to_UCCnetGBO_env_publicationCommand_catalogueItemPublication
CwItemBasic_to_UCCnetTPIXSD_envelope_publicationCommand_catalogueItem
CwItemBasic_to_UCCnetTPIXSD_envelope_registerCommand_CIP_itemAddChange
CwItemBasic_to_UCCnetTPIXSD_envelope_registerCommand_itemAddChange
RouterMap_CwItemBasic_to_UCCnetTPIDTD_envelope
RouterMap_UCCnetGBO_envelope_to_UCCnetTPIDTD_envelope
```

To implement multiple trading partners, you need either multiple versions of these maps or you need to set these values in the collaboration object.

Creating and configuring the second instance of the TPIConnector: This connector is only required 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:
 - **Windows:**
 - 1) In the Windows taskbar, 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 TPIConnector short cut and rename it to TPI2Connector.

- 4) Right-click the TPIConnector short cut and select **Properties**.
- 5) 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 ICS server, as shown in the following example:

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

- **UNIX:**

- 1) Access the TPI Agent program located in the following directory:
<install_path>/IBM/WebSphereICS/connectors/TPI/.
- 2) Run TPI by switching to this directory and entering the following command, where *ICS_server_name* refers to the name of the ICS 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 31. Connector-specific properties

Property name	Value
ArchiveProcessedDocDir	Directory where processed document meta-events are archived (for instance, C:\TPISolo\data\uccnet2\archive2).
DataHandlerConfigMO	MO_DataHandler_UCCnet_envelope
DefaultXMLMimeType	text/xml
DocumentOutDir	Directory location where outbound documents are written temporarily before TPI processes them (for instance, C:\TPISolo\data\uccnet2\xmlout). 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 instance, C:\TPISolo\data\uccnet2\xmlin2).
PollQuantity	1 (add this property it does not appear in the list of connector-specific properties).
TradingPartnerConfigurationFile	Fully qualified name of the trading partner configuration file created in Step 1 above (for instance, C:\IBM\WebSphereICS\connectors\TPI\tpcfg.txt).
WaitForMDN	false (MDNs are not supported by this solution).

Table 32. Supported business objects

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

2. 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
UCCnetTPIXSD_envelope	UCCnetTPIXSD_envelope_to_UCCnetGBO_envelope
ItemBasic	CwItemBasic_to_UCCnetTPIXSD_envelope_registerCommand_itemAddChange

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 ICS and *ICS_server_name* refers to the name of the ICS 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 JTextIsoftConnector

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 used to simulate the iSoftConnector. Like the iSoftConnector, the JTextIsoftConnector 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 JTextIsoft connector. Otherwise, you only need to create and configure the first instance.

Creating and configuring the first instance of the JTextIsoftConnector:

1. Create the JTextIsoftConnector using the System Manager, as follows:
 - a. Open the System Manager.
 - b. Save the JTextIsoftConnector by doing the following:
 - 1) Open the JTextConnector.
 - 2) Click **File > Save As > To Project**.
 - 3) Save the connector as JTextIsoftConnector.
 - c. Create the JTextIsoftConnector Agent, as follows:
 - **Windows:**
 - 1) In the Windows taskbar, 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 JTextIsoft Connector.
 - 4) Right-click the JTextIsoft Connector short cut and select **Properties**.
 - 5) Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to JTextIsoft, where *ICS_server_name* refers to the name of the ICS server, as shown in the following example:


```
<install_path>\connectors\JText\start_JText.bat JTextIsoft ICS_server_name
```
 - **UNIX:**
 - 1) Access the JTextIsoftConnector Agent program located in the following directory:


```
<install_path>/IBM/WebSphereICS/connectors/JText/.
```

- 2) Run JTextISoft by switching to this directory and entering the following command, where *ICS_server_name* refers to the name of the ICS server:

```
start_JText.sh JTextISoft ICS_server_name
```

Perform the following steps to configure the JTextISoftConnector:

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

Table 34. Standard property

Property name	Value
ApplicationName	JTextISoftConnector

Table 35. Supported business objects

Business object name	Agent support required?
MO_JTextISoftConnector_Default	Yes
MO_DataHandler_UCCnet_envelope	Yes
<ul style="list-style-type: none"> UCCnetDTD_envelope (when the DTD XML definition type is used) UCCnetXSD_envelope (when the XSD XML definition type is used) 	Yes
UCCnetGBO_envelope	No
ItemBasic	No

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

Table 36. Associated maps

Business object name	Map name
UCCnetDTD_envelope	UCCnetDTD_envelope_to_UCCnetGBO_envelope
UCCnetXSD_envelope	UCCnetXSD_envelope_to_UCCnetGBO_envelope
UCCnetGBO_envelope	DTD support RouterMap_UCCnetGBO_envelope_to_UCCnetDTD_envelope XSD support UCCnetGBO_envelope_to_UCCnetXSD_envelope
ItemBasic	DTD support RouterMap_CwItemBasic_to_UCCnetDTD_envelope XSD support CwItemBasic_to_UCCnetXSD_envelope_to_registerCommand_itemAddChange

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 JTextISoftConnector 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 ICS and *ICS_server_name* refers to the name of the ICS server. Enter the following at a command prompt:

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

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

1. Create the JTextISOFT2Connector using the System Manager, as follows:
 - a. Open the System Manager.
 - b. Save the JTextISOFT2Connector by doing the following:
 - 1) Open the JTextISOFTConnector.
 - 2) Click **File > Save As > To Project**.
 - 3) Save the connector as JTextISOFT2Connector.
 - c. Create the JTextISOFT2Connector Agent, as follows:
 - **Windows:**
 - 1) In the Windows taskbar, 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 JTextISOFTConnector short cut and rename it to JTextISOFT2Connector.
 - 4) Right-click the JTextISOFT2Connector short cut and select **Properties**.
 - 5) Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to JTextISOFT2, where *ICS_server_name* refers to the name of the ICS server, as shown in the following example:
 <install_path>\connectors\JText\start_JText.bat JTextISOFT2 *ICS_server_name*
 - **UNIX:**
 - 1) Access the JTextISOFT2Connector Agent program located in the following directory:
 <install_path>/IBM/WebSphereICS/connectors/JText/.
 - 2) Run JTextISOFT2 by switching to this directory and entering the following command, where *ICS_server_name* refers to the name of the ICS server:
 start_JText.sh JTextISOFT2 *ICS_server_name*

Perform the following steps to configure the JTextISOFT2Connector:

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

Table 37. Standard property

Property name	Value
ApplicationName	JTextISOFT2Connector

Table 38. Supported business objects

Business object name	Agent support required?
MO_JTextISOFTConnector_Default	Yes
MO_DataHandler_UCCnet_envelope	Yes
UCCnetXSD_envelope	Yes

Table 38. Supported business objects (continued)

Business object name	Agent support required?
ItemBasic	No

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

Table 39. Associated maps

Business object name	Map name
UCCnetXSD_envelope	UCCnetXSD_envelope_to_UCCnetGBO_envelope
ItemBasic	CwItemBasic_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 ICS and *ICS_server_name* refers to the name of the ICS server. Enter the following at a command prompt:

```
runmqsc local_WebSphere_ICS_queue_manager_name
DEFINE QLOCAL (AP/JTEXTISOFT2CONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/JTEXTISOFT2CONNECTOR) USAGE (NORMAL)
END
```

Creating and configuring the JTextTPICConnector

Create and configure this connector only if you are using the TPI server and are communicating with UCCnet through the UCCnet CLU or testing your installation. This is a copy of the JTextConnector used to simulate the TPICConnector. Like the TPICConnector, the JTextTPICConnector uses the IBM WebSphere Business Integration Data Handler for XML and generates the same XML output. If you are setting up your system to use the XSD CIP operation, you must create and configure two instances of the JTextTPI connector.

Creating and configuring the first instance of the JTextTPICConnector: To create the JTextTPICConnector, complete the following steps:

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

To configure the JTextTPICConnector, complete the following steps::

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

Table 40. Standard property

Property name	Value
ApplicationName	JTextTPICConnector

Table 41. Supported business objects

Business object name	Agent support required?
MO_JTextTPIConnector_Default	Yes
MO_DataHandler_UCCnet_envelope	Yes
<ul style="list-style-type: none"> UCCnetTPIDTD_envelope (when the DTD XML definition type is used) UCCnetTPIXSD_envelope (when the XSD XML definition type is used) 	Yes
UCCnetGBO_envelope	No
ItemBasic	No

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

Table 42. Associated maps

Business object name	Map name
UCCnetTPIDTD_envelope	UCCnetTPIDTD_envelope_to_UCCnetGBO_envelope
UCCnetTPIXSD_envelope	UCCnetTPIXSD_envelope_to_UCCnetGBO_envelope
UCCnetGBO_envelope	DTD support RouterMap_UCCnetGBO_envelope_to_UCCnetTPIDTD_envelope XSD support UCCnetGBO_envelope_to_UCCnetTPIXSD_envelope
ItemBasic	DTD support RouterMap_CwItemBasic_to_UCCnetTPIDTD_envelope XSD support CwItemBasic_to_UCCnetTPIXSD_envelope_registerCommand_itemAddChange

3. Save to the configuration (**File > Save > To Project**), then close the Connector Configurator.
4. Create the JTextTPIConnector Agent, as follows:
 - **Windows:**
 - a. In the Windows taskbar, 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 JTextTPI Connector.
 - d. Right-click the JTextTPI Connector short cut and select **Properties**.
 - e. Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to JTextTPI, where *ICS_server_name* refers to the name of the ICS server, as shown in the following example:
`<install_path>\connectors\JText\start_JText.bat JTextTPI ics_server_name`
 - **UNIX:**
 - a. Access the JTextTPIConnector Agent program located in the following directory: `<install_path>/IBM/WebSphereICS/connectors/JText/`.
 - b. Run JTextTPI by switching to this directory and entering the following command, where *ICS_server_name* refers to the name of the ICS server:
`start_JText.sh JTextTPI ICS_server_name`

5. If you are using WebSphere MQ as your connector transport, create the JTextTPIConnector 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 ICS and *ICS_server_name* refers to the name of the ICS server. Enter the following at a command prompt:

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

Creating and configuring the second instance of the JTextTPIConnector: To create the JTextTPI2Connector, complete the following steps:

1. Open the System Manager.
2. Save the JTextTPI2Connector by doing the following:
 - a. Open the JTextTPIConnector.
 - b. Click **File > Save As > To Project**.
 - c. Save the connector as JTextTPI2Connector.

Perform the following steps to configure the JTextTPI2Connector:

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

Table 43. Standard property

Property name	Value
ApplicationName	JTextTPI2Connector

Table 44. Supported business objects

Business object name	Agent support required?
MO_JTextTPI2Connector_Default	Yes
MO_DataHandler_UCCnet_envelope	Yes
UCCnetTPIXSD_envelope	Yes
ItemBasic	No

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

Table 45. Associated maps

Business object name	Map name
UCCnetTPIXSD_envelope	UCCnetTPIXSD_envelope_to_UCCnetGBO_envelope
ItemBasic	CwItemBasic_to_UCCnetTPIXSD_envelope_publicationCommand_catalogueItem

3. Save to the configuration (**File > Save > To Project**), then close the Connector Configurator.
4. Create the JTextTPI2Connector Agent, as follows:
 - **Windows:**
 - a. In the Windows taskbar, 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 JTextTPI2 Connector.

- d. Right-click the JTextTPI2Connector short cut and select **Properties**.
- e. Click the **Short cut** tab, edit the **Target field**, and set the first command line argument to JTextTPI2, where *ICS_server_name* refers to the name of the ICS server, as shown in the following example:


```
<install_path>\connectors\JText\start_JText.bat JTextTPI2 ics_server_name
```
- **UNIX:**
 - a. Access the JTextTPIConnector Agent program located in the following directory: *<install_path>/IBM/WebSphereICS/connectors/JText/*.
 - b. Run JTextTPI2 by switching to this directory and entering the following command, where *ICS_server_name* refers to the name of the ICS server:


```
start_JText.sh JTextTPI2 ICS_server_name
```
5. If you are using WebSphere MQ as your connector transport, create the JTextTPI2Connector 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 ICS and *ICS_server_name* refers to the name of the ICS server. Enter the following at a command prompt:


```
runmqsc local_WebSphere_ICS_queue_manager_name
DEFINE QLOCAL (AP/JTEXTTPI2CONNECTOR/ICS_server_name) USAGE (NORMAL)
DEFINE QLOCAL (IC/ICS_server_name/JTEXTTPI2CONNECTOR) USAGE (NORMAL)
END
```

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 46. Supported business object

Business object name	Agent support required?
ItemBasic	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 ICS and *ICS_server_name* refers to the name of the ICS 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:

- Create the EmailConnector Agent, as follows:
 - **Windows:**
 1. In the Windows taskbar, 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**.

- 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 ICS server, as shown in the following example:

```
<install_path>\connectors\email\start_email.bat
EMail ICS_server_name
```
- Configure the EmailConnector to include the following configuration property. Use the value shown in the following table.

Table 47. Connector-specific property

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

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

Configuring the SAPConnector

If you are using SAP as the source for your item information, configure the SAPConnector according to the Adapter for mySAP.com (SAP R/3 V.4.x) User Guide. Configure the connector to include the appropriate business objects. Use the values shown in the following table.

Table 48. Supported business objects

Business object name	Agent support required?
SAP4_MatlBasic	Yes
ItemBasic	No

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

Table 49. Associated maps

Business object name	Map name
SAP4_MatlBasic	Sa4CwItemBasic
ItemBasic	CwSa4ItemBasic

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

A TestConnector can be used to simulate a SAP connection (refer to “Configuring the TestConnector” for details). If you are going to use the TestConnector, you must also save the SAP connector configuration to a file (**File > Save > To File**). However, if you are connecting to a live SAP system, refer to the document Adapter for mySAP.com (SAP R/3 V.4.x) User Guide for additional configuration steps.

Configuring the TestConnector

The SAPConnector Agent can be simulated using a TestConnector if a live SAP installation is not available.

- By referring to the section “Configuring the SAPConnector,” configure the SAPConnector to include the appropriate business objects and save the configuration to a file.
- Start the **Start > Programs > IBM WebSphere InterChange Server > IBM WebSphere Business Integration Toolset > Development > Test Connector** shortcut.
- Select **File > Create/Select Profile**.
- On the Connector Profile window, select **File > New Profile**.

5. In the New Profile window, provide the path to the connector configuration file saved in Step 1 above. The Connector Name for this example is SAPConnector. Choose the Broker Type from the drop-down list, and fill in the ICS server name and password. Finally, click **OK** to save the profile.
6. In the Connector Profile window, select the profile just created.
7. Click **OK** at the bottom of the window, which will return to the main VTC window.
8. To make the connection with the ICS, select **File > Connect**. When the connection is complete, the message SAPConnector Ready and the date and time are displayed at the bottom of the window.

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. When you have completed creating and configuring the collaboration objects, proceed to the section “Configuring the relationships before deployment” on page 43.

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

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

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

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

Table 50. UCCnet_ItemSync collaboration object ports

Collaboration object	Port	Type	Bind to
UCCnet_ItemSyncObject	From	connector	Source application's connector
	To (Using DTD)	connector	AS2 channel connector
	To (Using XSD)	collaboration object	ItemCommandRouter_Obj:From
	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 51. UCCnet_ItemSync collaboration object configuration properties

Collaboration object	Property name	Property description	Required
UCCnet_ItemSyncObject	GtinDB_USER	The user ID used to connect to the database containing the PROCESSED_GTIN and trading_partner tables.	Yes
	GtinDB_PASSWORD	The password for the GtinDB_USER.	Yes

Table 51. UCCnet_ItemSync collaboration object configuration properties (continued)

Collaboration object	Property name	Property description	Required
	JDBC_DRIVER	Pointer to the driver used to connect to the database containing the PROCESSED_GTIN and trading_partner tables.	It is required if the ICS does not register the driver in the Java Virtual Machine (JVM). The Microsoft SQL driver, for example, is already registered in the JVM by the ICS and does not need to be specified.
	JDBC_URL	The JDBC URL corresponding to the JDBC_DRIVER specified.	Yes
	SEND_EMAIL_TO	Defines the email address to which problems detected during execution of collaboration object processing are sent. This entry is for ICS 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	Dictates whether email is sent to the address set in the SEND_EMAIL_TO property.	Yes, if you want to send email. To send email, set it to the value all. Otherwise, set the value to none.
	AUDITLOG_INSTANCE_NAME	Identifies the tag in the UCCnet section within the InterchangeServer.cfg file under which other tags needed for audit logging are contained. The default value is UCCnet. This value must be correct for the CSIAuditLogger class to find the database information to make audit log entries.	Yes
	SUPPLIER_NAME	The name of the partner that is synchronizing item information with UCCnet. This value is logged in audit entries. A sample entry is SAP.	No

Creating and configuring the ItemCommandRouter collaboration object and making its port connections

This object is only required for schema (XSD) support. It is not used for DTD support. There are two possible ways to configure this collaboration object, depending on whether you implement your own data pool (CIN operation) or use UCCnet as a data pool (CIP operation).

If you set up your system for CIN operation, then create and configure a collaboration object based on the ItemCommandRouter collaboration template:

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

Table 52. ItemCommandRouter collaboration object ports (CIN operation)

Collaboration object	Port	Type	Bind to
ItemCommandRouter_Obj	From	collaboration object	UCCnet_ItemSyncObject:To
	ToCIN_CI	collaboration object	CIN_CIP_Dispatcher_forICR_Obj:From
	ToRCIR	connector	AS2 channel connector

- Set the tracing level for the object.

If you set up your system for CIP operation, then create and configure a collaboration object based on the ItemCommandRouter collaboration template as follows:

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

Table 53. ItemCommandRouter collaboration object ports (CIP operation)

Collaboration object	Port	Type	Bind to
ItemCommandRouter_Obj	From	collaboration object	UCCnet_ItemSyncObject:To
	ToCIN_CI	collaboration object	ItemCommandRouter_DE_LIST_WITHDRAWObject:FromItemBasic
	ToRCIR	connector	AS2 channel connector

Note: For CIP operation, the ToCin_CI port is configured to send an email indicating that delist and withdrawal actions are not supported.

- Set the tracing level for the object.

Creating and configuring CIN_CIP_Dispatcher collaboration objects and making their port connections

This collaboration object is only created if you are using schema (XSD) support. If you are setting up for CIN operation, you will need to create two objects. If you are setting up for CIP operation, you only need one. Complete the following steps:

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

Table 54. CIN_CIP_Dispatcher collaboration object ports (CIN operation)

Collaboration object	Port	Type	Bind to
CIN_CIP_Dispatcher_forICR_Obj	From	collaboration object	ItemCommandRouter_Obj:ToCIN_CI

Table 54. CIN_CIP_Dispatcher collaboration object ports (CIN operation) (continued)

Collaboration object	Port	Type	Bind to
	To	connector	AS2 channel connector
	DestinationAppRetrieve	connector	PortConnector
CIN_CIP_Dispatcher_forPWL_Obj	From	collaboration object	UCCnet_processWorklist: RCIR_RESPONSE
	To	connector	AS2 channel connector
	DestinationAppRetrieve	connector	PortConnector

Table 55. CIN_CIP_Dispatcher collaboration object ports (CIP operation)

Collaboration object	Port	Type	Bind to
CIN_CIP_Dispatcher_forPWL_Obj	From	collaboration object	UCCnet_processWorklist: PUBLICATION_CMD_RESPONSE
	To	connector	AS2 channel connector
	DestinationAppRetrieve	connector	PortConnector

- Set the tracing level for each object.
- Configure the collaboration object properties using the values from the following table. Both objects use the same property values.

Table 56. CIN_CIP_Dispatcher collaboration object configuration properties

Collaboration object	Property name	Property description	Required
CIN_CIP_Dispatcher_forICR_Obj CIN_CIP_Dispatcher_forPWL_Obj	AUDIT_LOG_INSTANCE_NAME	Identifies the tag in the UCCnet section the InterchangeServer.cfg file that contains the other tags needed for audit logging. This value must be correct for the CSIAuditLogger class to find the database information to make audit log entries. Default: UCCnet	Yes
	DISPATCHER_GLN_FILE	The fully qualified name of the file that holds the GLNs subscribed to each category code. Default: C:\IBM\WebSphereICS\UCCnet \dispatcher_gln_file.txt	Yes

Table 56. CIN_CIP_Dispatcher collaboration object configuration properties (continued)

Collaboration object	Property name	Property description	Required
	TO_UCCNETGBO_PROCESSING_MAP	Called by the CIN_CIP_Dispatcher object to map incoming item basic objects to the outgoing Catalogue Item Notification or Catalogue Item Publication message. Value: For CIN operation (default): <code>CwItemBasic_to_UCCnetGBO_envel \</code> <code>ope_notifyCommand_catalogue \</code> <code>eItem</code> For CIP operation: <code>CwItemBasic_to_UCCnetGBO_env_pub \</code> <code>licationCommand_catalogueItemPu \</code> <code>blication</code>	Yes
	SUPPLIER_NAME	The name of the partner responsible for synchronizing item information with UCCnet. SAP, for example. This value is logged in audit entries. No default value.	No
	GLN_ATTRIBUTE	Identifies where in the UCCnetGBO business object the GLN retrieved from the Dispatcher GLN file should be placed. Values to use: For CIN operation (default): <code>ROOT.body[0].transaction.command \</code> <code>[0].notifyCommand.notifyCommand \</code> <code>Operand.catalogueItemNotificati \</code> <code>on.catalogueItem.dataRecipient</code> For CIP operation: <code>ROOT.body[0].transaction.command \</code> <code>[0].publicationCommand.publicat \</code> <code>ionCommandOperand.catalogueItem \</code> <code>Publication.publishToGLN</code>	
	TOPIC_ATTRIBUTE	Identifies where in the UCCnetGBO business object the collaboration should get the topic value used for audit logging. Values to use: For CIN operation (default): <code>ROOT.body[0].transaction.command \</code> <code>[0].notifyCommand.notifyComman \</code> <code>dOperand.catalogueItemNotifica \</code> <code>tion.notificationTopic.topic</code> For CIP operation: <code>ROOT.body[0].transaction.comman \</code> <code>d[0].publicationCommand.publi \</code> <code>cationCommandHeader.type</code>	

The GLN_DISPATCHER file contains one or more category codes with a list of subscribing retailers after each category code. For example:

UDEX.01.0020.0855
7789333000026
7789333000027
7789333000028

UDEX.05.0139.0334
7789333000026
7789333000027
7789333000028
7789333000029
7789333000030
7789333000031

Here, UDEX.01.0020.0855 and UDEX.05.0139.0334 are the category codes. The numbers underneath each category code are the GLNs for the subscribing retailers.

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

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

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

Table 57. UCCnet_requestWorklist collaboration object ports

Collaboration object	Port	Type	Bind to
UCCnet_requestWorklistObject	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 58. UCCnet_requestWorklist collaboration object configuration properties

Collaboration object	Property name	Property description	Required
UCCnet_requestWorklistObject	DTD_URL	Sets the value for the DocType line in outgoing XML. The default value is: DOCTYPE envelope SYSTEM "http://www.uccnet.net/ \ \ xmlschema/2.2/Envelope.dtd" For schema support, the value is left blank.	No, unless the default value is incorrect for the environment to which the XML messages are being sent.
	SET_UNIQUE_IDS	Controls whether unique IDs (messageIdentifier and uniqueCreateorIdentification) are set in the output XML messages. Possible values for this property are: <ul style="list-style-type: none">• ALL (the default value — set all three unique IDs)• NONE• BLANK (set unique ID only if it is blank in the input)	No, unless the default value needs to be changed.

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

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

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

Table 59. UCCnet_processWorklist collaboration object ports

Collaboration object	Port	Type	Bind to
UCCnet_processWorklistObject	UNKNOWN_RESPONSE	collaboration object	UCCnet_processWorklist_UNKNOWN_RESPONSEObject:From
	CATALOGUE_ITEM_CONFIRMATION	DTD support connector XSD support collaboration object	DTD support PortConnector XSD support UCCnet_processWorklist_CATALOGUE_ITEM_CONFIRMATIONObject:From
	DestinationAppRetrieve	connector	PortConnector
	NEW_ITEM_PUBLICATION_REQUEST	connector	AS2 channel connector
	AUTHORIZATION_RESPONSES	collaboration object	UCCnet_processWorklist_AUTHORIZATION_RESPONSESObject:From
	DEAD_LETTER_PUB_RECEIPT	collaboration object	UCCnet_processWorklist_DEAD_LETTER_PUB_RECEIPTObject:From
	From	connector	AS2 channel connector
	ITEM_ADD_CHANGE	connector	AS2 channel connector
	INITIAL_ITEM_LOAD_REQUEST	connector	AS2 channel connector
	CATEGORY_ADD_CHANGE	collaboration object	UCCnet_processWorklist_CATEGORY_ADD_CHANGEObject:From
	UNKNOWN_MESSAGES	collaboration object	UCCnet_processWorklist_UNKNOWN_MESSAGESObject:From
	SIMPLE_RESPONSE	collaboration object	UCCnet_processWorklist_SIMPLE_RESPONSEObject:From
	RCIR_RESPONSE	DTD support connector XSD support, CIN operation collaboration object XSD support, CIP operation connector	DTD support PortConnector XSD support, CIN operation CIN_CIP_Dispatcher_forPWL_Obj:From XSD support, CIP operation Second instance of the AS2 channel connector
	CIN_RESPONSE	DTD support connector XSD support, CIN operation collaboration object XSD support, CIP operation connector	DTD support PortConnector XSD support, CIN operation UCCnet_processWorklist_CIN_RESPONSEObject:From XSD support, CIP operation PortConnector
	CI_RESPONSE	DTD support connector XSD support, CIN operation connector XSD support, CIP operation collaboration object	DTD support PortConnector XSD support, CIN operation PortConnector XSD support, CIP operation UCCnet_processWorklist_CI_RESPONSEObject:From

Table 59. UCCnet_processWorklist collaboration object ports (continued)

Collaboration object	Port	Type	Bind to
	CIP_RESPONSE	DTD support connector XSD support, CIN operation connector XSD support, CIP operation collaboration object	DTD support PortConnector XSD support, CIN operation PortConnector XSD support, CIP operation UCCnet_processWorklist_CIP_RESPON \ SEObject:From
	PUBLICATION_CMD_RESPONSE	DTD support connector XSD support, CIN operation connector XSD support, CIP operation collaboration object	DTD support PortConnector XSD support, CIN operation PortConnector XSD support, CIP operation CIN_CIP_Dispatcher_forPWL_Obj:From
	FAILURE	Collaboration object	UCCnet_processWorklist_FAILURE_R \ ESPONSEObject:From
	RCIR_QUERY_RESPONSE	DTD support Connector XSD support Collaboration object	DTD support Port connector XSD support UCCnet_processWorklist_RCIR_QUERY_RESPONSEObject:From

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

Table 60. UCCnet_processWorklist collaboration object configuration properties

Collaboration object	Property name	Property description	Required
UCCnet_processWorklistObject	DB_USER	The user of the database associated with the WebSphere business integration system.	Yes
	DB_PASSWORD	The password for the DB_USER.	Yes
	JDBC_DRIVER	Pointer to the driver code used to connect to the database associated with the WebSphere business integration system.	Only if the ICS does not register the driver in the JVM. The Microsoft SQL driver, for example, is already registered in the JVM by the ICS and does not need to be specified.
	JDBC_URL	The JDBC URL corresponding to the JDBC_DRIVER specified.	Yes

Table 60. UCCnet_processWorklist collaboration object configuration properties (continued)

Collaboration object	Property name	Property description	Required
	SEND_EMAIL_TO	Defines the email address to which problems detected during execution of the collaboration object are sent. This entry is for ICS 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 and a valid entry in this field to send email. If this field is left blank, no email is sent.
	AUDITLOG_INSTANCE_NAME	Identifies the tag in the UCCnet section within the InterchangeServer.cfg file under which other tags needed for audit logging are contained. The default value is UCCnet. This value must be correct for the CSIAuditLogger class to find the database information to make audit log entries.	Yes
	SUPPLIER_NAME	The name of the partner that is synchronizing item information with UCCnet. This value is logged in audit entries. A sample entry is SAP.	No
	DTD_URL	Sets the value for the DocType line in outgoing XML. The default value is: DOCTYPE envelope SYSTEM \ "http://www.uccnet.net/ \ xmlschema/2.2/Envelope.dtd" For schema support, the value is left blank.	No, unless the default value is incorrect for the environment to which the XML messages are being sent.
	AUTO_RESPOND	Controls whether an automatic response is sent to UCCnet. The default value is: YES.	No, unless the default value needs to be changed.

Creating and configuring Notify_by_eMail collaboration objects and making their port connections

When using a DTD implementation, you must create and configure seven collaboration objects based on the Notify_by_eMail collaboration template. When using a schema implementation, you must create and configure either ten or twelve, depending on whether you are using CIN or CIP operation. To do this, complete the following steps:

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

Table 61. Notify_by_eMail collaboration objects and ports

Collaboration object	Port	Type	Bind to
UCCnet_processWorklist_AUTHORIZATION_RESPONSESObject	From	collaboration object	UCCnet_processWorklistObject: AUTHORIZATION_RESPONSES

Table 61. *Notify_by_eMail* collaboration objects and ports (continued)

Collaboration object	Port	Type	Bind to
	FromItemBasic	connector	Port connector
UCCnet_processWorklist_CATEGORY_ADD_CHANGEObject	From	collaboration object	UCCnet_processWorklistObject:CATEGORY_ADD_CHANGE
	FromItemBasic	connector	Port connector
UCCnet_processWorklist_DEAD_LETTER_PUB_RECEIPTObject	From	collaboration object	UCCnet_processWorklistObject:DEAD_LETTER_PUB_RECEIPT
	FromItemBasic	connector	Port connector
UCCnet_processWorklist_SIMPLE_RESPONSEObject	From	collaboration object	UCCnet_processWorklistObject:SIMPLE_RESPONSE
	FromItemBasic	connector	Port connector
UCCnet_processWorklist_UNKNOWN_MESSAGESObject	From	collaboration object	UCCnet_processWorklistObject:UNKNOWN_MESSAGES
	FromItemBasic	connector	Port connector
UCCnet_processWorklist_UNKNOWN_RESPONSEObject	From	collaboration object	UCCnet_processWorklistObject:UNKNOWN_RESPONSE
	FromItemBasic	connector	Port connector
UCCnet_processWorklist_FAILURE_RESPONSEObject	From	collaboration object	UCCnet_processWorklistObject:FAILURE
	FromItemBasic	connector	Port connector

Table 62. *Notify_by_eMail* collaboration objects and ports (schema specific)

Collaboration object	Port	Type	Bind to
UCCnet_processWorklist_CATALOGUE_ITEM_CONFIRMATIONObject	From	collaboration object	UCCnet_processWorklistObject:CATALOGUE_ITEM_CONFIRMATION
	FromItemBasic	connector	Port connector
UCCnet_processWorklist_RCIR_QUERY_RESPONSEObject	From	collaboration object	UCCnet_processWorklistObject:RCIR_QUERY_RESPONSE
	FromItemBasic	connector	Port connector
UCCnet_processWorklist_CI_RESPONSEObject (For CIP operation only)	From	collaboration object	UCCnet_processWorklistObject:CI_RESPONSE
	FromItemBasic	connector	Port connector
UCCnet_processWorklist_CIP_RESPONSEObject (For CIP operation only)	From	collaboration object	UCCnet_processWorklistObject:CIP_RESPONSE
	FromItemBasic	connector	Port connector
ItemCommandRouter_DE_LIST_WITHDRAWObject (For CIP operation only)	From	connector	Port connector
	FromItemBasic	collaboration object	ItemCommandRouter_Obj:ToCIN_CI
UCCnet_processWorklist_CIN_RESPONSEObject (For CIN operation only)	From	collaboration object	UCCnet_processWorklistObject:CIN_RESPONSE
	FromItemBasic	connector	Port connector

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

Note: Several sample email files exist in the following directory:

- Windows:**
<ICS_installation_path>\Samples\UCCnet\collaborations\eMail
- UNIX:** You must FTP (ASCII mode) the eMail files to a directory on your UNIX system.

All of the collaboration objects based on the Notify_by_eMail collaboration template must have their properties updated to point to the correct location of these sample files or to your own custom email files.

Table 63. Notify_by_eMail collaboration object configuration properties. Use the values in this table for all of the following collaboration objects:

ItemCommandRouter_DE_LIST_WITHDRAWObject
 UCCnet_processWorklist_AUTHORIZATION_RESPONSEObject
 UCCnet_processWorklist_CATALOGUE_ITEM_CONFIRMATIONObject
 UCCnet_processWorklist_CATEGORY_ADD_CHANGEObject
 UCCnet_processWorklist_DEAD_LETTER_PUB_RECEIPTObject
 UCCnet_processWorklist_FAILURE_RESPONSEObject
 UCCnet_processWorklist_SIMPLE_RESPONSEObject
 UCCnet_processWorklist_UNKNOWN_MESSAGESObject
 UCCnet_processWorklist_UNKNOWN_RESPONSEObject
 UCCnet_processWorklist_CATALOGUE_ITEM_CONFIRMATIONObject (used with schema support only)
 UCCnet_processWorklist_CI_RESPONSEObject (used with schema support only)
 UCCnet_processWorklist_CIP_RESPONSEObject (used with schema support only)
 UCCnet_processWorklist_CIN_RESPONSEObject (used with schema support only)
 UCCnet_processWorklist_RCIR_QUERY_RESPONSEObject (used with schema support only)

Property name	Property description	Required
EMAIL_NOTIFICATION_RCPTS	Email address of the recipients.	Yes
EMAIL_SUBJECT	<p>Subject line of the email. This value can contain variables in the form <code>\${variable_name}</code> into which the collaboration object substitutes data from the business object dynamically. Type text using the following escape codes:</p> <ul style="list-style-type: none"> <code>\${getRoot}</code> — substitutes the entire triggering business object. <code>\${getDate}</code> — substitutes the current date and time. <code>\${getName}</code> — substitutes the name of the triggering business object. <code>\${getVerb}</code> — substitutes the verb of the triggering business object. <code>\${attribute}</code> — substitutes the value of the named attribute from the triggering business object. If the value for the <i>variable_name</i> does not match one of the specific values above, the collaboration object interprets it as the name of a business object attribute. <p>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.</p> <p>See the Solution development guide for information about how the solution handles sending email and for a sample value for this property.</p>	Yes

Table 63. *Notify_by_eMail* collaboration object configuration properties (continued). Use the values in this table for all of the following collaboration objects:

ItemCommandRouter_DE_LIST_WITHDRAWObject
 UCCnet_processWorklist_AUTHORIZATION_RESPONSEObject
 UCCnet_processWorklist_CATALOGUE_ITEM_CONFIRMATIONObject
 UCCnet_processWorklist_CATEGORY_ADD_CHANGEObject
 UCCnet_processWorklist_DEAD_LETTER_PUB_RECEIPTObject
 UCCnet_processWorklist_FAILURE_RESPONSEObject
 UCCnet_processWorklist_SIMPLE_RESPONSEObject
 UCCnet_processWorklist_UNKNOWN_MESSAGESObject
 UCCnet_processWorklist_UNKNOWN_RESPONSEObject
 UCCnet_processWorklist_CATALOGUE_ITEM_CONFIRMATIONObject (used with schema support only)
 UCCnet_processWorklist_CI_RESPONSEObject (used with schema support only)
 UCCnet_processWorklist_CIP_RESPONSEObject (used with schema support only)
 UCCnet_processWorklist_CIN_RESPONSEObject (used with schema support only)
 UCCnet_processWorklist_RCIR_QUERY_RESPONSEObject (used with schema support only)

Property name	Property description	Required
EMAIL_MESSAGE	<p>Body of the email. 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:</p> <p>@c:\IBM\WebSphereICS\UCCnet\collaborations\eMail\ \UCCnet_processWorklist_AUTHORIZATION_RESPONSES.mail</p>	Yes

Configuring the relationships before deployment

Set up the relationships, as follows:

1. In the System Manager, navigate to the Static relationships in the Integration Component Library.
2. Open the Relationship Designer tool by selecting a relationship and choosing **Edit Definition** from the right-click menu or, alternatively, by double-clicking any one of the static relationships directly.
3. Verify the URL and DBMS type values for each relationship in the **Advanced Settings** view by selecting each relationship and choosing **Advanced Settings** from the right-click menu. These values must agree with the global DBMS settings for the rest of the server configuration.
4. Save any revised settings. Note that each relationship must be saved individually (do not use the **Save All** menu option).
5. Select each relationship to be saved and choose **Save** from right-click menu.

Note: If you are using DB2, the relationships contain SQL stored procedures. DB2 requires that a C compiler be installed on the same server on which the ICS resides so that the stored procedures can be compiled during the relationship deployment. This is not required for Microsoft SQL Server or Oracle databases.

Note: After the relationships have been deployed as a result of the steps in the section “Deploying the solution” on page 44, you must then populate the

relationships with default values and make modifications needed in the default schemas. See the section “Populating the relationships after deployment” for more information.

Deploying the solution

Deploy the solution, as follows:

1. After all of the components of the solution have been configured, compile the maps and collaboration templates in the System Manager.
2. Create a User Project that contains all of the solution components.
3. Use the Deploy wizard to deploy the solution to the WebSphere InterChange Server (running in design mode). The wizard allows you to choose what parts of the solution to deploy. It is recommended that you deploy the solution in stages, as follows:
 - a. Deploy the business objects.
 - b. Deploy the maps. Ensure that you have checked the **Compile** option.
 - c. Deploy the relationships. Ensure that you have checked the **Create schema** option.
 - d. Deploy the connectors.
 - e. Deploy the collaboration templates. Ensure that you have checked the **Compile** option.
 - f. Restart the WebSphere InterChange Server.
 - g. Start the System Monitor and start the relationships.
 - h. Deploy the collaboration objects.

Populating the relationships after deployment

After the relationships have been deployed as a result of the steps in the section “Deploying the solution,” you must then populate the relationships with default values and make modifications needed in the default schemas by completing the following steps. Be sure to follow the instructions appropriate for your platform where indicated.

- **Windows:**

- DB2 —

1. Start a DB2 session by opening a DB2 command window by clicking **Start > Programs > IBMDB2 > Command Line Tools > Command Window**.
2. Move to the following directory:
`<ICS_installation_path>\collaborations\dependencies\ \ UCCnet\db2`
3. At the prompt, enter the following command: **db2 connect to <repository_name> USER <userid> USING <password>** where **<repository_name>** is the name of your WebSphere InterChange Server repository database, and **<userid>** and **<password>** are the ID and password that were used to create it.
4. After you are connected:
 - For solutions implemented with DTD support, enter the command **db2 < InitializeRelationshipTables.sql** to run the script.
 - For solutions implemented with schema support, enter the command **db2 < InitializeRelationshipTablesForXSD.sql** to run the script.

- Oracle —

1. Move to the following directory:

<ICS_installation_path>\collaborations\dependencies\ \
UCCnet\oracle

2. Enter the following command:

- For solutions implemented with DTD support:

sqlplus dbuser/passwd@SID @InitializeRelationshipTables.sql

- For solutions implemented with schema support:

sqlplus dbuser/passwd@SID @InitializeRelationshipTablesForXSD.sql

You might have to enable autoextension of the CWRepository datafile.
Select the **AUTOEXTEND** option on the **Storage** tab for the datafile in the
DBA Studio.

– Microsoft SQL Server —

1. Move to the following directory:

<ICS_installation_path>\collaborations\dependencies\ \
UCCnet\mss

2. Select the CWRepository database in the drop-down in the menu bar.

3. Execute the following script by using the Query Analyzer tool:

- For solutions implemented with DTD support,

InitializeRelationshipTables.sql

- For solutions implemented with schema support,

InitializeRelationshipTablesForXSD.sql

• UNIX:

– DB2 —

1. Transfer the

<ICS_installation_path>\collaborations\dependencies\UCCnet\db2
\InitializeRelationshipTables file or

<ICS_installation_path>\collaborations\dependencies\UCCnet\db2
\InitializeRelationshipTablesForXSD file to the UNIX system using FTP
in ASCII mode. Place it in the

<ICS_installation_path>/collaborations/dependencies/UCCnet/db2
directory.

2. Move to the following directory on the UNIX system:

<ICS_installation_path>/collaborations/dependencies/ \
UCCnet/db2

3. Start a DB2 session by entering the command **db2**.

4. At the prompt, enter the command **connect to repository_name**.

5. After you are connected:

- For solutions implemented with DTD support, enter the command

<InitializeRelationshipTables.sql to run the script.

- For solutions implemented with schema support, enter the command

<InitializeRelationshipTablesForXSD.sql to run the script.

– Oracle —

1. Transfer the

<ICS_installation_path>\collaborations\dependencies\UCCnet
\oracle\InitializeRelationshipTables file or

<ICS_installation_path>\collaborations\dependencies\UCCnet\db2
\InitializeRelationshipTablesForXSD file to the UNIX system using FTP
in ASCII mode. Place it in the

<ICS_installation_path>/collaborations/dependencies /UCCnet/oracle
directory.

2. Move to the following directory on the UNIX system:

```
<ICS_installation_path>/collaborations/dependencies/ \
UCCnet/oracle
```

3. Enter the following command:

- For solutions implemented with DTD support:

```
sqlplus dbuser/passwd@SID @InitializeRelationshipTables.sql
```

- For solutions implemented with schema support:

```
sqlplus dbuser/passwd@SID @InitializeRelationshipTablesForXSD.sql
```

You might have to enable autoextension of the CWRepository datafile.
Select the **AUTOEXTEND** option on the **Storage** tab for the datafile in the
DBA Studio.

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
AS/400e
CrossWorlds
DB2
DB2 Universal Database
iSeries
Lotus
Lotus Notes
MQIntegrator
MQSeries
OS/400
Tivoli
WebSphere

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

MMX, Pentium, and ProShare are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

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

UCC and UCCnet are trademarks of Uniform Code Council, Inc., UCCnet, Inc. or both, in the United States, other countries, or both.

UCCnet Messaging is a product and/or trademark of UCCnet and is used with permission.

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

IBM WebSphere InterChange Server Version 4.2.2

IBM WebSphere Business Integration Toolset Version 4.2.2

IBM WebSphere Business Integration Adapters Version 2.4



