

*IBM CrossWorlds*  
WebSphere® Business Integration for  
Retail Distribution



# UCCnetMessageReceive Collaboration

*Version 4.1.1*

**Note!**

Before using this information and the product it supports, be sure to read the general information under “Notices and Trademarks” on page 13.

**First Edition (October 2002)**

This edition applies to Version 4, Release 1, Modification 1, of *IBM® CrossWorlds®* (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. All rights reserved.

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

---

## Contents

### **UCCnetMessageReceive collaboration . 1**

Required documents . . . . .	2
Collaboration setup . . . . .	2
Port information . . . . .	2
Setting up the collaboration . . . . .	4
Synchronization process . . . . .	4
Overall process logic. . . . .	4
Inherited process logic . . . . .	8
Configuration properties . . . . .	8
Standard properties . . . . .	9

Collaboration-specific properties . . . . .	9
Viewing collaboration messages . . . . .	12
See also. . . . .	12

### **Notices and Trademarks . . . . . 13**

Notices . . . . .	13
Programming interface information . . . . .	14
Trademarks and service marks . . . . .	15



---

## UCCnetMessageReceive collaboration

The UCCnetMessageReceive collaboration receives messages from UCCnet, acting as an interface between the Trading Partner Interchange (TPI) server and any other components bound to this collaboration for continued processing. The collaboration iterates through a UCCnet message, separating each instance of an Item Publication, and sending each through processing. If the UCCnet message does not contain Item Publications, the message is searched for item and message errors sent from UCCnet. Any error found is logged and processing stops for that message.

Item Publication commands are used to publish, or send, information to demand-side trading partners utilizing UCCnet. This collaboration currently supports only UCCnet Item Publication commands required for item synchronization, which include the following:

### **PUB\_RELEASE\_NEW\_ITEM**

Used to notify all demand-side trading partners that a new item has been submitted (published) to UCCnet by the supply-side trading partner.

### **PUB\_RELEASE\_INITIAL\_ITEM\_LOAD**

Used by a supply-side trading partner to initiate synchronization of data for items that were traded prior to the supply-side trading partner utilizing UCCnet or to initiate resynchronization of item data. From the viewpoint of the demand-side trading partner, this command is handled as a batch of PUB\_RELEASE\_NEW\_ITEM commands.

### **PUB\_RELEASE\_CORRECTION**

Used to notify all demand-side trading partners that an existing item is being updated to reflect changes in data previously contained within UCCnet. This command can be performed only on items that have been previously published.

### **PUB\_RELEASE\_DATA\_CHANGE**

Another command used to notify all demand-side trading partners that an existing item is being updated to reflect changes in data previously contained within UCCnet. However, when a supply-side trading partner issues a PUB\_RELEASE\_DATA\_CHANGE command to UCCnet, the command creates a new item with the new data changes with the same identification number, as opposed to simply updating the existing item with new data (such as the PUB\_RELEASE\_CORRECTION command performs). The demand-side trading partner receives the PUB\_RELEASE\_DATA\_CHANGE showing the new item that is to replace the old one.

### **PUB\_RELEASE\_WITHDRAW**

Used to notify one or more demand-side trading partners that an item is temporarily unavailable from the supply-side trading partner (a seasonal item, for instance). No additional item data synchronization occurs with the selected demand-side trading partners for this particular item.

### **PUB\_RELEASE\_DELIST**

Used to notify all demand-side trading partners that an item is no longer being manufactured by the supply-side trading partner and must be removed from the demand-side trading partners' systems. No additional item data synchronization occurs for this particular item.

The UCCnetMessageReceive collaboration handles the complete set of item description fields used by UCCnet.

---

## Required documents

The UCCnetMessageReceive collaboration is based on CollaborationFoundation and uses its features, ports, and configuration properties. UCCnetMessageReceive also has features, ports, and configuration properties unique to it.

To create and configure a UCCnetMessageReceive collaboration object, use the following documents:

- This document for the UCCnetMessageReceive collaboration-specific information.
- Standard Collaboration Processes for information about business processes inherited from CollaborationFoundation.
- Standard Collaboration Properties for information about configuration properties inherited from CollaborationFoundation.
- Collaboration Development Guide for information about CollaborationFoundation, and for general information about creating and configuring collaboration objects.

---

## Collaboration setup

This section includes the following information:

- “Port information”
- “Setting up the collaboration” on page 4

### Port information

The following figure illustrates UCCnetMessageReceive collaboration’s ports, as they are displayed in IBM® CrossWorlds® System Manager (CSM):

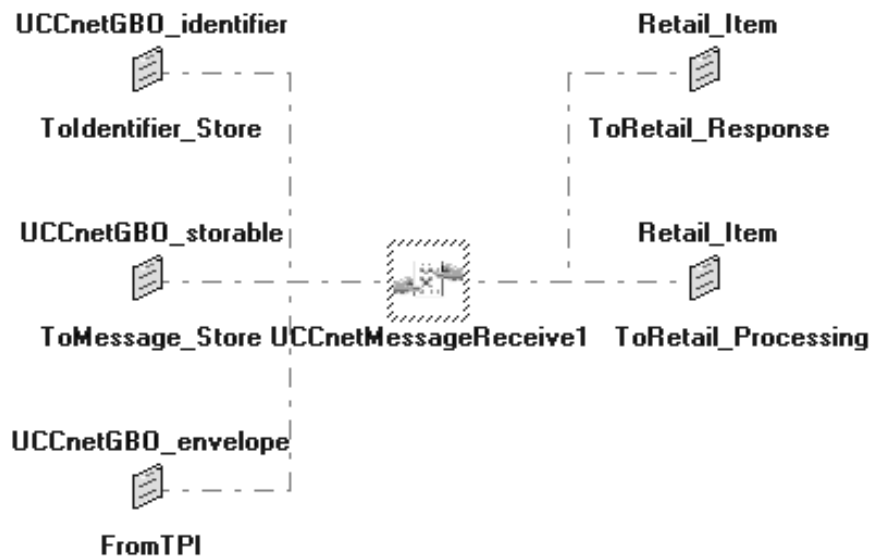


Figure 1. UCCnetMessageReceive collaboration's ports

**Note:** In the tables throughout this section, spaces have been inserted in some values in the Business object, Bound to, and Function columns to enable the entries to fit in the table cells. The actual values do not include spaces.

Table 1. Port name: FromTPI

Business object	Bound to	Function	Verbs used
UCCnetGBO _envelope (result of UCCnet _envelope business object passing from the TPI connector through the UCCnet_envelope _to_UCCnetGBO _envelope map)	TPI connector	Receives the triggering business object. At the end of a synchronous call, this port also returns the triggering business object to the source application when the collaboration ends successfully.	Create

Table 2. Port name: ToRetail\_Processing

Business object	Bound to	Function	Verbs used
Retail_Item	A collaboration or connector that provides continued processing of the Retail_Item business object.	Sends the Retail_Item to the collaboration or connector for further processing.	Create

Table 3. Port name: ToMessage\_Store

Business object	Bound to	Function	Verbs used
UCCnetGBO_storable	A persistent storage mechanism, such as an implementation instance of the IBM CrossWorlds Collaboration for Data Storage.	Persists the UCCnetGBO_storable business object to a local message store.	Create, Retrieve, Update, Delete

Table 4. Port name: ToRetail\_Response

Business object	Bound to	Function	Verbs used
Retail_Item	UCCnetMessage Send collaboration	On the pend or rejection of an invalid item, sends the Retail_Item to the UCCnetMessage Send collaboration for immediate transmittal to UCCnet.	Create

Table 5. Port name: Toldentifier\_Store

Business object	Bound to	Function	Verbs used
UCCnetGBO_identifier	A persistent storage mechanism, such as an implementation instance of the IBM CrossWorlds Collaboration for Data Storage.	Persists the UCCnetGBO_identifier business object to an identifier store.	Create, Retrieve, Update, Delete

## Setting up the collaboration

To set up UCCnetMessageReceive as a stand-alone collaboration object, complete the following steps:

1. Create the UCCnetMessageReceive collaboration object.
2. Bind the ports as described in “Port information” on page 2.
3. Set the “Configuration properties” on page 8 for UCCnetMessageReceive.

---

## Synchronization process

This section illustrates the following business processes for this collaboration:

- “Overall process logic”
- “Inherited process logic” on page 8

## Overall process logic

The following flow shows the process logic for this collaboration’s Create verb:

1. The communication process begins when the TPI server receives the UCCnet worklist. The TPI server decrypts the message, writes it to a directory, and notifies the TPI connector with the data necessary to retrieve the document.



2. The TPI connector starts the flow through the UCCnetMessageReceive collaboration by instantiating the IBM CrossWorlds XML Data Handler with the received UCCnet message, which contains the UCCnet worklist.
3. The IBM CrossWorlds XML Data Handler creates a UCCnet\_envelope business object containing the entire UCCnet message, including each individual data instance and the commands related to it.
4. The TPI connector checks for subscriptions to the UCCnet\_envelope business object by collaborations. The UCCnetMessageReceive collaboration subscribes to this business object. Therefore, the UCCnetMessageReceive collaboration is passed the UCCnet\_envelope business object through its FromTPI port. However, first the UCCnet\_envelope business object is converted to a UCCnetGBO\_envelope business object by passing through the UCCnet\_envelope\_to\_UCCnetGBO\_envelope map.
5. The UCCnetMessageReceive collaboration iterates through the UCCnetGBO\_envelope business object, separating each instance of an Item Publication command. It allows UDEX category codes as received by UCCnet to be changed to internal category codes through the collaboration configuration property CATEGORYMAP\_FILE (see the section “Mapping category codes” on page 6 for more information on this feature; “CATEGORYMAP\_FILE” on page 9 for more information on the CATEGORYMAP\_FILE property).
6. For an Item Publication with a PUB\_RELEASE\_NEW\_ITEM, PUB\_RELEASE\_DATA\_CHANGE, PUB\_RELEASE\_WITHDRAW, or PUB\_RELEASE\_DELIST command, the UCCnetMessageReceive collaboration accommodates persisting the UCCnetGBO\_envelope business object to a local identifier store by sending the object to its ToIdentifier\_Store port (see the section “Persisting items to a local identifier store” on page 6 for more information on this feature). Before sending the business object to the ToIdentifier\_Store port, the collaboration can ensure that a duplicate item doesn’t already exist through configuration of the FILTER\_DUPLICATE configuration property (see the section “Filtering to eliminate processing of duplicate items” on page 8 for more information on this feature; “FILTER\_DUPLICATE” on page 10 for more information on the FILTER\_DUPLICATE property).
7. The UCCnetMessageReceive collaboration accommodates persisting the UCCnetGBO\_envelope business object to a local message store for use in a return message by sending it to its ToMessage\_Store port (see the section “Persisting items to a local message store” on page 7 for more information on this feature).
8. The UCCnetMessageReceive collaboration examines the UCCnetGBO\_envelope business object to detect whether certain attributes required by UCCnet are missing, or if the item is from a vendor or in a category accepted by the demand-side trading partner. See the section “Filtering publication requests” on page 7 for more information about filtering business objects. Based on this analysis, the collaboration handles processing as follows:
  - If the UCCnetMessageReceive collaboration detects that certain attributes required by UCCnet are missing, or that the item is from a vendor or in a category not accepted by the demand-side trading partner, it creates a Retail\_Item business object by passing the UCCnetGBO\_envelope business object containing the individual Item Publication command and its data through the map specified in the TORETAIL\_RESPONSE\_MAP property (see “TORETAIL\_RESPONSE\_MAP” on page 11 for more information on this property). It then handles the resulting business object according to the value specified in the FILTER\_FAIL\_RESPONSE configuration property (see

“FILTER\_FAIL\_RESPONSE” on page 10 for more information about this property). Any status specified in the FILTER\_FAIL\_RESPONSE property is attached regardless of what the original message type was in the UCCnetGBO\_envelope business object. The collaboration then sends this Retail\_Item business object through its ToRetail\_Response port to the UCCNetMessageSend collaboration for transmittal to UCCnet.

- If the UCCNetMessageReceive collaboration detects that all required information is complete and that the item is from a vendor and in a category accepted by the demand-side trading partner, it creates a Retail\_Item business object by passing the UCCnetGBO\_envelope business object containing the individual Item Publication command and its data through the map specified in the TORETAIL\_PROCESSING\_MAP property (see “TORETAIL\_PROCESSING\_MAP” on page 11 for more information on this property). It then sets the status of the resulting item to Pending, and sets the command type according to the original message type received from UCCnet. It sends this Retail\_Item business object through its ToRetail\_Processing port to a collaboration or connector that provides continued processing of the business object.

The UCCNetMessageReceive collaboration waits for a return from the processing collaboration before sending the next item.

### Mapping category codes

The UCCNetMessageReceive collaboration allows UDEX category codes as received by UCCnet to be changed to internal category codes through the collaboration configuration property CATEGORYMAP\_FILE. This property specifies the path and name of a file that contains a list of categories to be mapped and the values to which they are mapped.

The collaboration checks the item.itemInformation.categoryList.categoryCode field of the UCCnetGBO\_envelope business object. If the item contains a category value that matches a value listed in the file specified by the CATEGORYMAP\_FILE property, the collaboration changes that category value to the new value for it indicated in the file. If the category value is not listed in this file, the collaboration does not change the value.

Entries in this file have the structure category,newCategory, where the value represented by category is changed to the new value represented by newCategory (for instance, cat0001,FLCat001).

### Persisting items to local identifier and message stores

The UCCNetMessageReceive collaboration accommodates the ability to persist the UCCnetGBO\_envelope business object to local identifier and message stores.

**Persisting items to a local identifier store:** For an Item Publication with a PUB\_RELEASE\_NEW\_ITEM, PUB\_RELEASE\_DATA\_CHANGE, PUB\_RELEASE\_WITHDRAW, or PUB\_RELEASE\_DELIST command, the UCCNetMessageReceive collaboration accommodates persisting the business object to a local identifier store by sending it to its ToIdentifier\_Store port. The ToIdentifier\_Store port can then be bound to a persistence mechanism, such as an instance of the IBM CrossWorlds Collaboration for Data Storage.

Before sending the UCCnetGBO\_envelope business object to the port, the collaboration first converts the UCCnetGBO\_envelope business object to a UCCnetGBO\_identifier business object by passing it through the map specified in the TOIDENTIFIER\_STORE\_MAP property (see “TOIDENTIFIER\_STORE\_MAP” on page 11 for more information on this property). The collaboration can also

check if a duplicate item already exists in the identifier store. See the section “Filtering to eliminate processing of duplicate items” on page 8 for more information on this feature.

**Persisting items to a local message store:** The UCCnetMessageReceive collaboration accommodates persisting the business object to a local message store for use in a return message by sending it to its ToMessage\_Store port. The ToMessage\_Store port can then be bound to a persistence mechanism, such as an instance of the IBM CrossWorlds Collaboration for Data Storage.

Before persisting the UCCnetGBO\_envelope business object to the message store, the collaboration first converts the UCCnetGBO\_envelope business object to a UCCnetGBO\_storable business object by passing it through the map specified in the TOMESSAGE\_STORE\_MAP property (see “TOMESSAGE\_STORE\_MAP” on page 11 for more information on this property). The UCCnetGBO\_storable business object is a combination of one instance of an Item Publication, together with the message header and routing information for the worklist message in which the item was contained.

### **Filtering publication requests**

The UCCnetMessageReceive collaboration can filter the publication requests that arrive from UCCnet to ensure that certain attributes required by UCCnet are present, or that the items are from a supply-side trading partner or in a category accepted by the demand-side trading partner. Item Publications with PUB\_RELEASE\_NEW\_ITEM, PUB\_RELEASE\_DATA\_CHANGE, PUB\_RELEASE\_WITHDRAW, or PUB\_RELEASE\_DELIST commands can also be checked against an existing identifier store to eliminate processing of duplicate items.

**Filtering based on the presence of required attributes:** The user can specify through an external file that certain UCCnetGBO\_envelope attributes must be present and not contain NULL or BLANK values (as determined by the IBM CrossWorlds business object methods isNull() and isBlank(); isBlank() returns true when an attribute contains a zero-length string). The path and name of this file are specified in the REQUIRED\_ATTRIBUTE\_FILE configuration property.

The collaboration checks the attributes contained in the UCCnetGBO\_envelope business object against the list of attributes named in the file specified in the REQUIRED\_ATTRIBUTE\_FILE property. If any of the attributes are missing or are present but missing data, the collaboration logs the business object and handles it according to the value specified in the FILTER\_FAIL\_RESPONSE configuration property. If no file is specified in the REQUIRED\_ATTRIBUTE\_FILE property, or if no fields are indicated within it, all items are accepted for further processing by the collaboration.

### **Filtering based on items belonging to approved supply-side trading partners:**

The user can specify through an external file a list of trading partners or vendors from which items are accepted. The path and name of this file are specified in the VENDOR\_FILE configuration property. The collaboration checks the notification.entityIdentification.globalLocationNumber.gln field of the UCCnetGBO\_envelope business object. If the item is from a vendor listed in the file specified in the VENDOR\_FILE property, the collaboration processes it normally. If the item is from a vendor not listed in this file, the collaboration logs it and handles it according to the value specified in the FILTER\_FAIL\_RESPONSE configuration property. If no file is specified in the property, all items are accepted for further processing by the collaboration.

Entries in the file specified in the `VENDOR_FILE` configuration property have the structure *vendorGLN,other\_data,other\_data,...*, where the *vendorGLN* is the GLN (assigned by UCCnet) to an organization permitted through the collaboration's vendor filter and the *other\_data* entries are other data entries the user might want to include in the file. The vendor GLN must be the first entry in the line. Any number of associated attributes can be appended to the line as long as they are separated by commas (.). Do not place a comma before the vendor GLN. The following is an example of a valid entry:

```
0001111222333,TestVendor,(111)111-1111,contact@TestVendor.com
```

**Filtering based on items belonging to accepted categories:** The user can specify through an external file a list of categories from which items are accepted. The path and name of this file are specified in the `CATEGORY_FILE` configuration property. The collaboration checks the `item.itemInformation.categoryList.categoryCode` field of the `UCCnetGBO_envelope` business object. If the item is from a category listed in the file specified in the `CATEGORY_FILE` property, the collaboration processes it normally. If the item is not from a category listed in this file, it handles it according to the value specified in the `FILTER_FAIL_RESPONSE` configuration property. If no file is specified in the property, all items are accepted for further processing by the collaboration.

Each line of the file must contain only one category value. Each entry can include embedded special characters, such as periods (.) and commas (,), as long as the characters are valid within the category. Do not place characters in the line other than those specified in the category. The following is an example of a valid entry:

```
UDEX.0001.0001.001
```

**Filtering to eliminate processing of duplicate items:** For an Item Publication with a `PUB_RELEASE_NEW_ITEM`, `PUB_RELEASE_DATA_CHANGE`, `PUB_RELEASE_WITHDRAW`, or `PUB_RELEASE_DELIST` command, the `UCCnetMessageReceive` collaboration persists the `UCCnetGBO_envelope` business object through its `ToIdentifier_Store` port to a local identifier store as a `UCCnetGBO_identifier` business object. Before persisting the item, the collaboration checks the value of the `FILTER_DUPLICATE` configuration property. If the value is `true`, the collaboration checks if an item with an identical `gtin`, `version`, and `topic` exists in the identifier store. If an identical item does exist, the second entry with identical information is logged as a duplicate and further processing of the identical item ends. If an identical item does not exist, the item is added to the local identifier store and processed normally.

## Inherited process logic

This collaboration inherits the following business process from the `CollaborationFoundation` template:

- Exception handling

For information on this process, see *Standard Collaboration Processes*.

---

## Configuration properties

This section describes the following properties for this collaboration:

- “Standard properties” on page 9
- “Collaboration-specific properties” on page 9

## Standard properties

This collaboration inherits the following standard configuration properties from the CollaborationFoundation template:

- ADDITIONAL\_RETRIEVE
- 1\_FILTER\_ATTRIBUTE
- INFORMATIONAL\_EXCEPTIONS
- 1\_INCLUDE\_VALUES
- 1\_FAIL\_ON\_INVALID\_VALUE
- CONVERT\_UPDATE
- USE\_RETRIEVE
- 1\_EXCLUDE\_VALUES
- SEND\_EMAIL
- CONVERT\_CREATE

For more information on these configuration properties, see Standard Collaboration Properties.

**Note:** Do not enable this collaboration's inherited configuration properties. Changing default values for these inherited properties can cause processing to malfunction.

## Collaboration-specific properties

This collaboration has the following collaboration-specific configuration properties:

- "CATEGORY\_FILE"
- "CATEGORYMAP\_FILE"
- "DEBUG" on page 10
- "FILTER\_DUPLICATE" on page 10
- "FILTER\_FAIL\_RESPONSE" on page 10
- "TOIDENTIFIER\_STORE\_MAP" on page 11
- "TOMESSAGE\_STORE\_MAP" on page 11
- "REQUIRED\_ATTRIBUTE\_FILE" on page 11
- "TORETAIL\_PROCESSING\_MAP" on page 11
- "TORETAIL\_RESPONSE\_MAP" on page 11
- "VENDOR\_FILE" on page 12
- "UTILITY\_CLASS" on page 12

### CATEGORY\_FILE

This property specifies the path and name of a file that contains a list of the categories from which items are accepted. See the section "Filtering based on items belonging to accepted categories" on page 8 for information on how to create this file.

The CATEGORY\_FILE property has no default value.

### CATEGORYMAP\_FILE

This property specifies the path and name of a file that contains a list of categories to be mapped and the values to which they are mapped. Mapping might be necessary to change UDEX category codes as received by UCCnet to the demand-side trading partner's internal category codes, for instance. See the section "Mapping category codes" on page 6 for information on how to create this file.

The CATEGORYMAP\_FILE property has no default value.

## DEBUG

This property allows debug messages to be sent to the IBM CrossWorlds InterChange Server (ICS) trace log. Messages include variable values, object values, and other run-time status.

**Note:** Use this property only for development testing purposes; do not use it in production code.

Table 6. DEBUG configuration property

Possible values	Usage
False (default)	Prevents debug messages from being sent to the trace log of the ICS.
True	Allows debug messages to be sent to the trace log of the ICS. Do not use this setting for production operation.

## FILTER\_DUPLICATE

This property controls whether the collaboration checks the identifier store for duplicates when PUB\_RELEASE\_NEW\_ITEM, PUB\_RELEASE\_DATA\_CHANGE, PUB\_RELEASE\_WITHDRAW, or PUB\_RELEASE\_DELIST Item Publications are received.

Table 7. FILTER\_DUPLICATE configuration property

Possible values	Usage
true (default)	The collaboration checks the identifier store for duplicates.
false	The collaboration does not check the identifier store for duplicates. As a result, duplicated items will exist in the identifier log. Multiple entries in the identifier store must be deleted manually.

## FILTER\_FAIL\_RESPONSE

This property determines how an item is processed if the collaboration discovers that it contains invalid data or that required data is missing from it.

Table 8. FILTER\_FAIL\_RESPONSE configuration property

Possible values	Usage
Pend (default)	The status of Retail_Item with missing or invalid information is set to Pending, and the business object is sent to the ToRetail_Response port so that a response can be built and sent to UCCnet.
Reject	The status of Retail_Item with missing or invalid information is set to Rejected, and the business object is sent to the ToRetail_Response port so that a response can be built and sent to UCCnet.
None	The Retail_Item with missing or invalid information continues processing. No response is built or sent to UCCnet.



## TOIDENTIFIER\_STORE\_MAP

This property contains the name of the map used to convert the UCCnetGBO\_envelope business object to a UCCnetGBO\_identifier business object. The default map name can be changed to any value desired. A space has been inserted in the value in the Possible values column to enable the entry to fit in the table cell. The actual value does not include a space.

Table 9. TOIDENTIFIER\_STORE\_MAP configuration property

Possible values	Usage
UCCnetGBO_envelope_to_UCCnetGBO_identifier	The name of the map for the ToIdentifier_Store port.

## TOMESSAGE\_STORE\_MAP

This property contains the name of the map used to convert the UCCnetGBO\_envelope business object to a UCCnetGBO\_storable business object. The default map name can be changed to any value desired. A space has been inserted in the value in the Possible values column to enable the entry to fit in the table cell. The actual value does not include a space.

Table 10. TOMESSAGE\_STORE\_MAP configuration property

Possible values	Usage
UCCnetGBO_envelope_to_UCCnetGBO_storable	The name of the map for the ToMessage_Store port.

## REQUIRED\_ATTRIBUTE\_FILE

This property specifies the path and name of a file that contains a list of UCCnetGBO\_envelope attributes, which if present in the UCCnetGBO\_envelope business object, must contain data. See the section “Filtering based on the presence of required attributes” on page 7 for information on how to create this file.

The REQUIRED\_ATTRIBUTE\_FILE property has no default value.

## TORETAIL\_PROCESSING\_MAP

This property contains the name of the map used to convert the UCCnetGBO\_envelope business object to a Retail\_Item business object when the object is being sent to the ToRetail\_Processing port. The default map name can be changed to any value desired.

Table 11. TORETAIL\_PROCESSING\_MAP configuration property

Possible values	Usage
UCCnetGBO_envelope_to_Retail_Item	The name of the map for the ToRetail_Processing port.

## TORETAIL\_RESPONSE\_MAP

This property contains the name of the map used to convert the UCCnetGBO\_envelope business object to a Retail\_Item business object when the object is being sent to the ToRetail\_Response port. The default map name can be changed to any value desired.

Table 12. TORETAIL\_RESPONSE\_MAP configuration property

Possible values	Usage
UCCnetGBO_envelope_to_Retail_Item	The name of the map for the ToRetail_Response port.

## VENDOR\_FILE

This property specifies the path and name of a file that contains a list of the vendors from which items are accepted. See the section “Filtering based on items belonging to approved supply-side trading partners” on page 7 for information on how to create this file.

The VENDOR\_FILE property has no default value.

## UTILITY\_CLASS

This property specifies the class name required by the collaboration to perform the filtering logic. The collaboration uses this property to check for the presence of the class before attempting to execute the required filtering logic. If this class is not available, the collaboration returns the Retail\_Item business object with its status set to Error.

Table 13. UTILITY\_CLASS configuration property

Possible values	Usage
com.ibm.wbi.retail.utils.RetailUtility	The default external Java™ class used by the collaboration to perform the filtering logic.

---

## Viewing collaboration messages

To view an explanation of this collaboration’s messages, invoke Message Browser and open the collaboration’s message file.

To invoke Message Browser and open the collaboration message file, complete the following actions:

1. In the Start menu, click **Programs > CrossWorlds > Server and Tools > Message Browser**.
2. On the **File** menu, click **Open**.
3. Use the **Look In** field to change the current folder to:

*IBM\_CrossWorlds\_root\_dir\collaborations\messages\  
UCCnetMessageReceive.txt*

---

## See also

- DataStore Collaboration
- UCCnetMessageSend Collaboration
- Retail\_Item Business Object
- UCCnet\_envelope Business Object
- UCCnetGBO\_envelope Business Object
- UCCnetGBO\_identifier Business Object
- UCCnetGBO\_storable Business Object



---

## Notices and Trademarks

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 CrossWorlds Lab Director  
IBM RTP Laboratory  
3039 Cornwallis Road  
P.O. BOX 12195  
Raleigh, NC 27709-2195  
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  
CrossWorlds  
the CrossWorlds logo  
DB2  
DB2 Universal Database  
MQIntegrator  
MQSeries  
Tivoli  
WebSphere

Lotus, Domino, Lotus Notes, and Notes Mail are trademarks of the Lotus Development Corporation in the United States, other countries, or both.

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.

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

IBM CrossWorlds Servers V4.1.1  
IBM CrossWorlds Full Toolset V4.1.1  
IBM CrossWorlds Connectors V4.1.1  
IBM CrossWorlds Collaborations V4.1.1

