IBM CrossWorlds
WebSphere® Business Integration for
Retail Distribution

MessageStore 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 9.

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.

Contents

MessageStore collaboration

The MessageStore collaboration serializes a UCCnetGBO_storable business object into XML and stores it in a database table through use of a user-specified key. It also enables the stored business object to be retrieved, updated, or deleted. The output of the MessageStore collaboration is bound to a JDBC connector, which actually performs the storage operations.

Required documents

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

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

- This document for the MessageStore 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.
- Data Handler Guide for information on how to configure data handlers using mime types, together with the application-specific information required in the business object that the IBM® CrossWorlds® XML Data Handler uses to serialize the object.
- Guide to the IBM CrossWorlds Connector for JDBC for information on how to use the JDBC connector, including how to set up the application-specific business object to be stored.

Collaboration setup

This section includes the following information:

- "Port information"
- "Setting up the collaboration" on page 3

Port information

The following figure illustrates the MessageStore collaboration's ports, as they are displayed in IBM CrossWorlds System Manager (CSM):

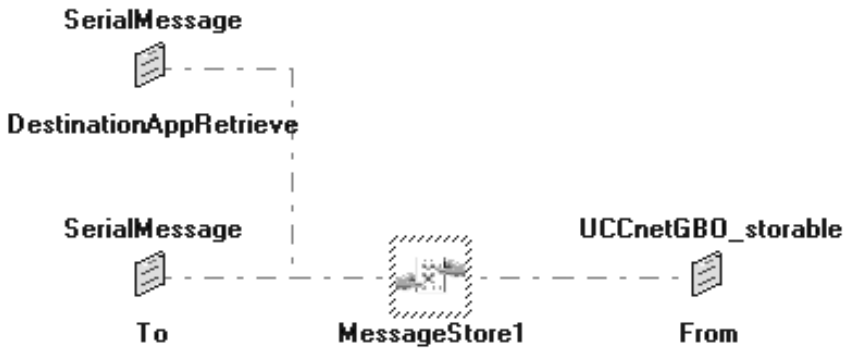

Figure 1. MessageStore collaboration's ports

Note: Some values in the Bound to column might include spaces to allow the entries to fit in the table cells. The actual values do not include spaces.

Table 1. Port name: From

Business object	Bound to	Function	Verbs used
UCCnetGBO_storable	UCCnetMessage Receive or UCCnetMessage Send collaborations, depending on the point in processing of the Solution.	Receives a UCCnetGBO_storable business object for a database operation. For Retrieve or Delete operations, only key attributes in the UCCnetGBO_storable business object need to be set.	Create, Update, Retrieve, Delete

Table 2. Port name: To

Business object	Bound to	Function	Verbs used
SerialMessage	JDBC connector	Creates, updates, or deletes a SerialMessage in the database. For the Delete operation, only the ObjectKey attribute in the SerialMessage business object needs to be set.	Create, Update, Delete

Table 3. Port name: DestinationAppRetrieve

Business object	Bound to	Function	Verbs used
SerialMessage	JDBC connector	Retrieves a SerialMessage from the database. Only the ObjectKey attribute in the SerialMessage business object needs to be set.	Retrieve

Setting up the collaboration

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

1. Create the MessageStore collaboration object.
2. Bind the ports as described in "Port information" on page 1.
3. Set the "Configuration properties" on page 6 for MessageStore.

Synchronization process

This section illustrates the following business processes for this collaboration:

- "Overall process logic"
- "Inherited process logic" on page 5

Overall process logic

Create

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

1. The MessageStore collaboration is triggered by the receipt of a UCCnetGBO_storable business object from a source such as the UCCnetMessageReceive collaboration.
2. The UCCnetGBO_storable business object is copied into a high-level object called DataStoreUCCnetGBO_storable. This object is passed into the IBM CrossWorlds XML Data Handler. By default, it contains the application-specific information needed by the data handler. The data handler outputs a string, which contains an XML representation of the UCCnetGBO_storable business object passed into the collaboration.
3. The MessageStore collaboration builds an application-specific business object called SerialMessage, which includes two attributes:
   - ObjectData, which is set equal to the XML output from the IBM CrossWorlds XML Data Handler.
   - ObjectKey, which is created from the key information in the UCCnetGBO_storable business object and the value of the MessageStore collaboration OBJECT_KEY property (see "OBJECT_KEY" on page 7 for more information on this property). This attribute holds the unique key used to store the object in the database.
4. The completed SerialMessage is sent to the To port of the MessageStore collaboration, which is bound to a JDBC connector. The JDBC connector uses the application-specific information in the SerialMessage to store it in the database. If the object already exists in the database, a ServiceCall exception is thrown and the Create operation fails.

Note: Although multiple objects in the database with identical keys can be created, the behavior of record retrievals, updates, and deletes is unpredictable, so avoid the use of non-unique keys.

Retrieve and Delete

The following flow shows the process logic for this collaboration's Retrieve verb:

1. The MessageStore collaboration is triggered by the receipt of a UCCnetGBO_storable business object from a source such as the UCCnetMessageReceive collaboration. The triggering business object passed into the MessageStore collaboration must contain values for all the fields that make up the ObjectKey attribute of the stored SerialMessage application-specific business object, as defined by the OBJECT_KEY MessageStore collaboration property (see "OBJECT_KEY" on page 7 for more information on this property).
2. The MessageStore collaboration builds a new SerialMessage application-specific business object, which includes two attributes:
   - ObjectData, which is NULL.
   - ObjectKey, which is created from the key information in the UCCnetGBO_storable business object and the value of the MessageStore collaboration OBJECT_KEY property. This attribute holds the unique key used to store the object in the database.
3. The newly created SerialMessage is passed to the JDBC connector via the DestinationAppRetrieve port.
4. Using the value in the ObjectKey attribute of the new SerialMessage to locate the stored object in the database, the JDBC connector retrieves the information in the ObjectData column from the stored object, copies this information into the ObjectData attribute of the new SerialMessage, and passes the new SerialMessage back to the MessageStore collaboration.
5. The MessageStore collaboration passes the SerialMessage to the IBM CrossWorlds XML Data Handler. The data handler uses the information in the ObjectData attribute to build a DataStoreUCCnetGBO_storable business object. The data handler passes this object back to the MessageStore collaboration.
6. The MessageStore collaboration converts the DataStoreUCCnetGBO_storable business object into a UCCnetGBO_storable business object, which is returned to the calling collaboration as the triggering business object.

The following flow shows the process logic for this collaboration's Delete verb:

1. The MessageStore collaboration is triggered by the receipt of a UCCnetGBO_storable business object from a source such as the UCCnetMessageReceive collaboration. The triggering business object passed into the MessageStore collaboration must contain values for all the fields that make up the ObjectKey attribute of the stored SerialMessage application-specific business object, as defined by the OBJECT_KEY MessageStore collaboration property (see "OBJECT_KEY" on page 7 for more information on this property).
2. The MessageStore collaboration builds a new SerialMessage application-specific business object, which includes two attributes:
   - ObjectData, which is NULL.
   - ObjectKey, which is created from the key information in the UCCnetGBO_storable business object and the value of the MessageStore collaboration OBJECT_KEY property. This attribute holds the unique key used to store the object in the database.

3. The newly created SerialMessage is passed to the JDBC connector via the To port.
4. The JDBC connector deletes the row in the database identified by the ObjectKey attribute. If the object does not exist in the database, a ServiceCall exception is thrown and the Delete operation fails.

Update

The following flow shows the process logic for this collaboration's Update verb:

1. The MessageStore collaboration is triggered by the receipt of a UCCnetGBO_storable business object from a source such as the UCCnetMessageReceive collaboration. The triggering business object passed into the MessageStore collaboration must contain values for all the fields that make up the ObjectKey attribute of the stored SerialMessage application-specific business object, as defined by the OBJECT_KEY MessageStore collaboration property (see "OBJECT_KEY" on page 7 for more information on this property).

   Note: The entire business object in the database will be replaced by this operation. Therefore, the calling collaboration might want to first retrieve the business object from the database (refer to the section "Retrieve and Delete" on page 4 for details on this process), modify any attributes which need to be updated, and then again invoke the MessageStore collaboration with the modified business object and the Update verb.
2. The MessageStore collaboration copies the triggering UCCnetGBO_storable business object into a high-level object called DataStoreUCCnetGBO_storable. This object is passed into the IBM CrossWorlds XML Data Handler. By default, it contains the application-specific information needed by the data handler. The data handler outputs a string, which contains an XML representation of the UCCnetGBO_storable business object passed into the collaboration.
3. The MessageStore collaboration builds an application-specific business object called SerialMessage, which includes two attributes:
   - ObjectData, which is set equal to the XML output from the IBM CrossWorlds XML Data Handler.
   - ObjectKey, which is created from the key information in the UCCnetGBO_storable business object and the value of the MessageStore collaboration OBJECT_KEY property. This attribute holds the unique key used to store the object in the database.
4. The completed SerialMessage is sent to the To port of the MessageStore collaboration, which is bound to a JDBC connector. The JDBC connector overwrites the existing SerialMessage in the database with the updated one. If the object does not already exist in the database, a ServiceCall exception is thrown and the Update operation fails.

Inherited process logic

This collaboration inherits the following business processes from the CollaborationFoundation template:

- Filtering data
- Additional Retrieve process
- Email process for error handling

For information on these processes, see Standard Collaboration Processes.

Configuration properties

This section describes the following properties for this collaboration:

- "Standard properties"
- "Collaboration-specific properties"

Standard properties

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

- 1_EXCLUDE_VALUES
- 1_FAIL_ON_INVALID_VALUE
- 1_FILTER_ATTRIBUTE
- 1_INCLUDE_VALUES
- ADDITIONAL_RETRIEVE
- CONVERT_CREATE
- CONVERT_UPDATE
- INFORMATIONAL_EXCEPTIONS
- SEND_EMAIL
- USE_RETRIEVE — not supported by the MessageStore collaboration.

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

Collaboration-specific properties

This collaboration has the following collaboration-specific configuration properties:

- "GENERATE_KEY"
- "MIME_TYPE"
- "OBJECT_KEY" on page 7
- "TEST" on page 7

Note: The property TEST is reserved.

GENERATE_KEY

Accept the default value of `false` for this collaboration, which means that the key used to create a record in the database is created from the attributes of the triggering business object specified in the OBJECT_KEY property.

MIME_TYPE

This property is used by the IBM CrossWorlds Data Handler to determine the type of serialization to be used. It tells the data handler to use the MO_DataHandler_XMLDataStoreConfig configuration meta-object for configuration information. Currently, only XML is supported. See the Data Handler Guide for information on configuring the IBM CrossWorlds Data Handler.

Table 4. MIME_TYPE configuration property

Possible values	Usage
text/xml.datastore	Use of this value assumes that the MO_Server_DataHandler business object has an attribute named text_xml_datastore with type MO_DataHandler_XMLDataStoreConfig.

OBJECT_KEY

Accept the default value of `correlationID` for this collaboration, which is the attribute from the triggering business object used to create the unique key that identifies the business object.

TEST

This property is reserved.

Table 5. TEST configuration property

Possible values	Usage
False (default)	Reserved.

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\MessageStore.txt`

See also

- UCCnetMessageReceive Collaboration
- UCCnetMessageSend Collaboration
- DataStoreUCCnetGBO_storable Business Object
- MO_DataHandler_XMLDataStoreConfig Business Object
- SerialMessage Business Object
- UCCnetGBO_storable Business Object
- Data Handler Guide
- Guide to the IBM CrossWorlds Connector for JDBC

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