

IBM CrossWorlds
WebSphere® Business Integration for
Retail Distribution



DataStore 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

DataStore collaboration 1

Required documents.	1
Collaboration setup	2
Port information	2
Setting up the collaboration	3
Synchronization process	5
Overall process logic.	5
Inherited process logic	8
Configuration properties	8
Standard properties	9

Collaboration-specific properties	9
Viewing collaboration messages	10
See also.	11

Notices and Trademarks 13

Notices	13
Programming interface information	14
Trademarks and service marks	15

DataStore collaboration

The DataStore collaboration serializes a Generic Business Object (GBO) into XML and stores it in a database table by using a user-specified key. The attributes in the triggering business object that are used by the DataStore collaboration to form the unique database table key are specified in a DataStore configuration property. The output of the DataStore collaboration is bound to a JDBC connector, which actually performs the storage operations. In addition to supporting a user-specified database key, the DataStore collaboration can also be configured to generate a unique key and return it to the calling collaboration.

The collaboration also enables the GBO to be retrieved, updated, or deleted. To retrieve or delete the GBO from the database, the calling collaboration must pass in a GBO containing values in its attributes, which are used to form the user-specified unique key.

For example purposes only, two sample generic business objects and one sample application-specific business object (ASBO) are shipped with the DataStore collaboration. The SampleObject business object is the sample triggering business object to be stored. The DataStoreSampleObject is a sample wrapper object used to contain the SampleObject GBO, which is passed to the IBM® CrossWorlds® XML Data Handler by the collaboration. The collaboration copies the serialized XML data handler output string into the sample SerialObject ASBO, which is actually stored in the database. As shipped, the DataStore collaboration template is configured to accept a SampleObject GBO, employ a DataStoreSampleObject business object, and produce a SerialObject ASBO.

For production purposes, however, users must edit these business objects, as follows:

- Copy the DataStore template and change the From port definition to specify the actual GBO to be supported.
- Create a wrapper object *DataStoreObjectType*, where *ObjectType* is the type of the GBO the From port is configured to accept.
- Change the To and DestinationAppRetrieve port definitions to specify support for an actual ASBO with the correct application-specific information.

Required documents

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

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

- This document for the DataStore 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 DataStore collaboration’s ports, as they are displayed in IBM CrossWorlds System Manager (CSM):

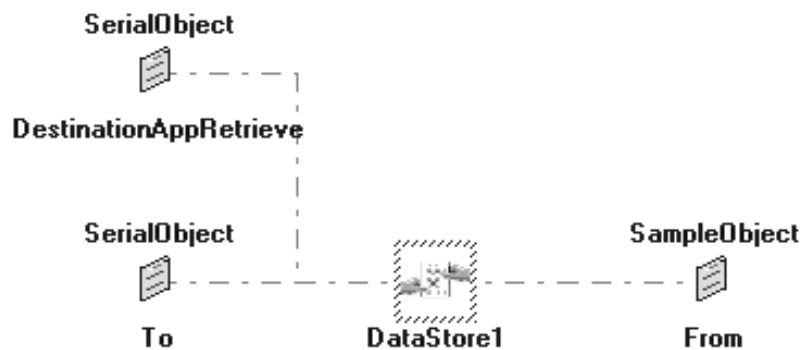


Figure 1. DataStore collaboration’s ports

Note: Throughout the tables, the sample business object names **SampleObject** and **SerialObject** are used to indicate the business objects involved in collaboration operation. In a production environment, these sample business objects would be replaced with user-specified ones.

Table 1. Port name: From

Business object	Bound to	Function	Verbs used
SampleObject	Collaboration that needs to persist a business object in a database.	Receives a SampleObject business object for a database operation. For Retrieve or Delete operations, only key attributes in the SampleObject business object need to be set.	Create, Retrieve, Delete, Update

Table 2. Port name: To

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

Table 3. Port name: DestinationAppRetrieve

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

Setting up the collaboration

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

1. Create a table in a database. See “Creating a database table” on page 4 for more information.
2. Create the wrapper object to be passed into the IBM CrossWorlds XML Data Handler. See the document DataStoreSampleObject for a description of the required components of this type of object.
3. Create the ASBO that will contain the serialized data string to be sent to the JDBC connector. See the document SerialObject for a description of the required components of this type of object.
4. Configure the JDBC connector. See “Configuring the JDBC connector” on page 4 for more information.
5. Configure the XML data handler. See “Configuring the XML data handler” on page 5 for more information.
6. Copy the DataStore collaboration template and modify the new template to specify the supported business objects for each port.

7. Create a collaboration object from the copied and modified collaboration template.
8. Bind the ports as described in “Port information” on page 2.
9. Set the “Configuration properties” on page 8 for DataStore.

Creating a database table

The DataStore collaboration copies business objects into a database table. This table must be created by the user in any database instance supported by the IBM CrossWorlds JDBC Connector. The database instance does not necessarily have to be on the same machine as the IBM CrossWorlds InterChange Server (ICS).

The database table must consist of two columns with the suggested names of ObjectKey and ObjectData. Specify values for these columns as shown in the table below. ObjectKey must be specified as a unique key for the table. If the database allows specification of the length of ObjectData, choose a value larger than the maximum anticipated length of the XML representation of the GBO being serialized. This value will vary according to the GBO being stored.

Table 4. Database table configuration

Column name	DB2® data type	Microsoft® SQL Server data type	Oracle data type
ObjectKey	VARCHAR (length =255) if the value of the DataStore GENERATE_KEY configuration property is false. Integer if the value of GENERATE_KEY is true.	VARCHAR (length =255) if the value of the DataStore GENERATE_KEY configuration property is false. Integer if the value of GENERATE_KEY is true.	VARCHAR (length =255) if the value of the DataStore GENERATE_KEY configuration property is false. Integer if the value of GENERATE_KEY is true.
ObjectData	CLOB	TEXT	CLOB

Configuring the JDBC connector

The JDBC connector must be set up to access the database table. See the Guide to the IBM CrossWorlds Connector for JDBC for configuration information specific to the database instance being used. The JDBC connector does not need to poll an events table looking for events. As a result, set the JDBC connector standard configuration property PollFrequency to the value no. The JDBC connector must be configured to support the SerialObject (or its replacement) with agent support.

If the DataStore configuration property GENERATE_KEY is true, the JDBC connector must be configured to generate a unique key. Any of the three methods specified in the Guide to the IBM CrossWorlds Connector for JDBC can be used. The App Spec Info in the ObjectKey attribute of the SerialObject ASBO (or its replacement) must be modified accordingly. Some JDBC connector configuration values specific to usage with the DataStore collaboration are listed in the following table:

Table 5. JDBC connector properties configuration

Property type	Property	Value
Standard	PollFrequency	no
Connector-specific	ArchiveProcessed	false
Connector-specific	EventTableName	null

Configuring the XML data handler

Configure the DataStore collaboration XML data handler by adding the following to the attributes of the MO_ServerDataHandler business object:

```
Name = text_xml_datastore
Type = MO_DataHandler_XMLDataStoreConfig
Card = 1
```

The MO_DataHandler_XMLDataStoreConfig business object is an example of a meta-object that is used to configure the XML data handler. It is identical to the MO_DataHandler_DefaultXMLConfig business object, except the BOPrefix attribute is blank. For the SampleObject GBO example, the BOPrefix must be blank so a meta-object is required.

See the Data Handler Guide for more information.

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 sections describe operation when the DataStore collaboration configuration properties GENERATE_KEY, CONVERT_CREATE, and CONVERT_UPDATE are set to false. See “GENERATE_KEY” on page 9 for information on the GENERATE_KEY property, Standard Collaboration Properties for information on the CONVERT_CREATE and CONVERT_UPDATE properties, and “Process logic changes resulting from GENERATE_KEY, CONVERT_CREATE, and CONVERT_UPDATE being set to true” on page 8 for information on collaboration operation when these properties are set to true.

Note: Throughout this section, the sample business object names SampleObject, DataStoreSampleObject, and SerialObject are used to indicate the business objects involved in collaboration operation. In a production environment, these sample business objects would be replaced with user-specified ones.

Create

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

1. The DataStore collaboration is triggered by the receipt of a SampleObject business object from a source collaboration.
2. The SampleObject business object is copied into a high-level object called DataStoreSampleObject. 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 SampleObject business object passed into the collaboration.
3. The DataStore collaboration builds an application-specific business object called SerialObject, 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 SampleObject business object and the value of the DataStore collaboration OBJECT_KEY

property (by default, `uniqueId`). See “OBJECT_KEY” on page 10 for more information on this property. This attribute holds the unique key used to store the object in the database.

4. The completed `SerialObject` is sent to the To port of the DataStore collaboration, which is bound to a JDBC connector. The JDBC connector uses the application-specific information in the `SerialObject` 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. Also, although it is possible for different instances of the DataStore collaboration to store different types of objects in the same table of the database (if the keys for all objects are unique), it is recommended that different types of objects be stored in different tables in the database.

Retrieve and Delete

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

1. The DataStore collaboration is triggered by the receipt of a `SampleObject` business object from a source collaboration. The triggering business object passed into the DataStore collaboration must contain values for all the fields that make up the `ObjectKey` attribute of the stored `SerialObject` application-specific business object, as defined by the OBJECT_KEY DataStore collaboration property. See “OBJECT_KEY” on page 10 for more information on this property.
2. The DataStore collaboration builds a new `SerialObject` application-specific business object, which includes two attributes:
 - `ObjectData`, which is NULL.
 - `ObjectKey`, which is created from the key information in the `SampleObject` business object and the value of the DataStore collaboration OBJECT_KEY property (by default, `uniqueId`). This attribute holds the unique key used to store the object in the database.
3. The newly created `SerialObject` is passed to the JDBC connector via the `DestinationAppRetrieve` port.
4. Using the value in the `ObjectKey` attribute of the new `SerialObject` 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 `SerialObject`, and passes the new `SerialObject` back to the DataStore collaboration.
5. The DataStore collaboration passes the `SerialObject` to the IBM CrossWorlds XML Data Handler. The data handler uses the information in the `ObjectData` attribute to build a `DataStoreSampleObject` business object. The data handler passes this object back to the DataStore collaboration.
6. The DataStore collaboration converts the `DataStoreSampleObject` business object into a `SampleObject` 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 DataStore collaboration is triggered by the receipt of a `SampleObject` business object from a source collaboration. The triggering business object passed into the DataStore collaboration must contain values for all the fields that make up the `ObjectKey` attribute of the stored `SerialObject`

application-specific business object, as defined by the OBJECT_KEY DataStore collaboration property. See “OBJECT_KEY” on page 10 for more information on this property.

2. The DataStore collaboration builds a new SerialObject application-specific business object, which includes two attributes:
 - ObjectData, which is NULL.
 - ObjectKey, which is created from the key information in the SampleObject business object and the value of the DataStore collaboration OBJECT_KEY property (by default, uniqueId). This attribute holds the unique key used to store the object in the database.
3. The newly created SerialObject 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 DataStore collaboration is triggered by the receipt of a SampleObject business object from a source collaboration. The triggering business object passed into the DataStore collaboration must contain values for all the fields that make up the ObjectKey attribute of the stored SerialObject application-specific business object, as defined by the OBJECT_KEY DataStore collaboration property. See “OBJECT_KEY” on page 10 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 6 for details on this process), modify any attributes which need to be updated, and then again invoke the DataStore collaboration with the modified business object and the Update verb.

2. The DataStore collaboration copies the triggering SampleObject business object into a high-level object called DataStoreSampleObject. 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 SampleObject business object passed into the collaboration.
3. The DataStore collaboration builds an application-specific business object called SerialObject, 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 SampleObject business object and the value of the DataStore collaboration OBJECT_KEY property (by default, uniqueId). This attribute holds the unique key used to store the object in the database.
4. The completed SerialObject is sent to the To port of the DataStore collaboration, which is bound to a JDBC connector. The JDBC connector overwrites the existing SerialObject 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.

Process logic changes resulting from **GENERATE_KEY**, **CONVERT_CREATE**, and **CONVERT_UPDATE** being set to true

The following changes in process logic occur when the indicated collaboration configuration properties are set to true:

- When the inherited collaboration configuration property **CONVERT_CREATE** is set to true, the Create verb is changed to Update if an object with the same ObjectKey as the triggering business object already exists in the database. The object currently in the database is overwritten with the new triggering business object. The DataStore collaboration prints out a trace message indicating that this verb change has occurred. With the exception of this change, all other parts of the flow remain the same.
- When the inherited collaboration configuration property **CONVERT_UPDATE** is set to true, the Update verb is changed to Create if no object with the ObjectKey exists in the database. A new record is added to the database for the new triggering business object. The DataStore collaboration prints out a trace message indicating that this verb change has occurred. With the exception of this change, all other parts of the flow remain the same.
- When the collaboration-specific configuration property **GENERATE_KEY** is set to true, the DataStore collaboration allows the JDBC connector to generate a unique key, use that key to store the serial data, and then return the key to the collaboration. The DataStore collaboration then puts the connector-generated key into the key field of the triggering business object, as specified in the **OBJECT_KEY** property. The **OBJECT_KEY** property must specify a single attribute when **GENERATE_KEY** is true. The JDBC connector provides three different ways to generate a unique key, using the UID parameter in the application-specific information for a business object attribute. Any of these three methods for generating UIDs is acceptable, so the choice might depend on the database being used. See the Guide to the IBM CrossWorlds Connector for JDBC for more information. The **CONVERT_CREATE** property must not be set to true when **GENERATE_KEY** is true because no key is specified with the Create verb. When the **CONVERT_UPDATE** property is set to true and **GENERATE_KEY** is true, a new row is created in the database. Use caution when enabling **CONVERT_UPDATE** when **GENERATE_KEY** is true, because the conversion can occur only when the triggering business object (with the verb Update) contains a previously generated key and the record with that key is not in the database. For many implementations, this might be considered an error condition.

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

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 DataStore 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 10
- "TEST" on page 10

Note: The property TEST is reserved.

GENERATE_KEY

This property is used to correctly handle a scenario in which the JDBC connector generates a unique key. The application-specific business object passed to the JDBC connector must be correctly set up so that the JDBC connector knows how to generate the key. See the Guide to the IBM CrossWorlds Connector for JDBC for more information.

Table 6. GENERATE_KEY configuration property

Possible values	Usage
false (default)	The key used to create a record in the database is created from the attributes of the triggering business object specified in the configuration property OBJECT_KEY.
true	A unique key is generated by the JDBC connector, updated in the triggering business object, and passed back to the calling collaboration.

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 7. *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

The OBJECT_KEY collaboration property consists of a comma-separated list of attributes that exist in the triggering business object, which will be concatenated to form a unique key. Each attribute in the list uses IBM CrossWorlds notation to specify a specific string in the hierarchy of the triggering business object. For instance, for a generic business object that contains child_business_object_1, and in which child_buisness_object_1 contains child_business_object_2 (where child_business_object_2 contains the target string), set the OBJECT_KEY property to: child_business_object_1.child_business_object_2.target. Note that the name of the actual generic business object does not have to be included.

If the GENERATE_KEY property is true, OBJECT_KEY must be a single String attribute of the triggering business object. When the unique key generated by the JDBC connector is returned to the DataStore collaboration, the attribute specified by the OBJECT_KEY is updated with the unique key.

Table 8. *OBJECT_KEY* configuration property

Possible values	Usage
comma-separated list of attributes	Any comma-separated list of attributes.

TEST

This property is reserved.

Table 9. *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\DataStore.txt

See also

- DataStoreSampleObject Business Object
- MO_DataHandler_XMLDataStoreConfig Business Object
- SampleObject Business Object
- SerialObject 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

