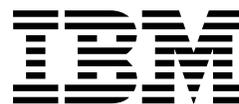


WebSphere® Business Integrator  
DataInterchange for Windows NT®

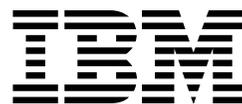


# User's Guide

**Version 2.1**



WebSphere® Business Integrator  
DataInterchange for Windows NT®



# User's Guide

**Version 2.1**

**Note:** Before using this information and the products it supports, read the information in “Notices” on page 35.

**First Edition (June 2001)**

This edition applies to Version 2.1 of IBM WebSphere Business Integrator (product number 5724-A78), DataInterchange for Windows NT<sup>®</sup>, and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2001. All rights reserved.

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

---

# Contents

<b>About this book</b> . . . . .	<b>v</b>	Building a solution . . . . .	12
Who this book is for . . . . .	v	<b>Messages and codes</b> . . . . .	<b>19</b>
What you need to know . . . . .	v	Error processing . . . . .	19
Related books . . . . .	v	<b>Programmer's reference</b> . . . . .	<b>25</b>
Glossaries . . . . .	vi	Control record format . . . . .	25
How to send your comments . . . . .	vii	MQAO message header contents . . . . .	26
Documentation conventions . . . . .	vii	Output files . . . . .	28
<b>Introduction</b> . . . . .	<b>1</b>	Usage scenarios . . . . .	29
DataInterchange gateway . . . . .	1	Receiving an application data format	
DataInterchange . . . . .	2	message from the BFM application. . . . .	29
Flexible setup and administration . . . . .	2	Sending an XML data message from BFM	
Superior translation capability . . . . .	2	for translation to EDI and sending . . . . .	30
Versatile communications . . . . .	3	Sending a command and receiving	
Extensive reporting and auditing . . . . .	3	response data from DataInterchange . . . . .	31
Standards support . . . . .	3	Receiving EDI data and translating it into	
Additional features . . . . .	4	XML . . . . .	32
. . . . .	4	Solution Manager exception service . . . . .	32
<b>Installation</b> . . . . .	<b>5</b>	Event log query . . . . .	33
Requirements . . . . .	5	Transaction Store query . . . . .	34
Installing WebSphere Business Integrator		<b>Notices</b> . . . . .	<b>35</b>
DataInterchange V2.1 . . . . .	6	Trademarks . . . . .	38
<b>Using Client with WebSphere</b>			
<b>Studio</b> . . . . .	<b>9</b>		
DataInterchange solution artifacts . . . . .	10		

**Contents**

---

## About this book

This guide contains information and instructions for installing and using DataInterchange in IBM® WebSphere® Business Integrator.

**Note:** The WebSphere Business Integrator DataInterchange Version 2.1 component is based on the DataInterchange Version 3 Release 1 product.

## Who this book is for

This document is intended for the developer designing a solution and the system administrator responsible for installing and managing DataInterchange in IBM WebSphere Business Integrator.

## What you need to know

Developers and those who run solutions need to be familiar with WebSphere Studio and have an expert knowledge of DataInterchange. An Application developer writing applications to drive DataInterchange through the adapter must know Java.

## Related books

The DataInterchange Version 3 Release 1 library consists of the books listed below. These books are relevant to WebSphere Business Integrator DataInterchange Version 2.1. Most of the functionality is the same, except for the specific OS/390® functionality as specified in this book.

All of the DataInterchange books are available at <http://www.ibm.com/services/interchange/>

- *DataInterchange Client User's Guide* (SB34-2010)

This guide describes how to setup and use the DataInterchange Client interface to create, and maintain DataInterchange profiles, maps, data formats, and EDI standards.

## About this book

- *DataInterchange Administrator's Guide* (SB34-2002)  
This guide describes the administrator activities for the product, including defining profiles, defining networks, and authorizing users. The audience for the book is the application programmer or data processing analyst.
- *DataInterchange Installation Guide* (GB09-8070)  
This book, in conjunction with the Program Directory, provides the necessary steps and information for installing the product. The book is intended for the system programmer and database administrator responsible for installing and implementing DataInterchange/MVS for OS/390 and DataInterchange/CICS® for OS/390.
- *DataInterchange Messages and Codes* (SB34-2000)  
This guide provides all logged and displayed messages, as well as information to aid the user and support personnel in identifying problems in DataInterchange/MVS for OS/390 and DataInterchange/CICS for OS/390. This reference guide is for the EDI administrator, the system programmer, and the database administrator.
- *DataInterchange Programmer's Reference* (SB34-2001)  
This book describes the administrator activities for the product, including defining profiles, defining networks, and authorizing users. The audience for the book is the application programmer or the data processing analyst.

Other related books include:

- *DataInterchange v3.1 Technical Implementation Guide*
- *IBM WebSphere Studio 3.5 Guide*

## Glossaries

For glossary definition terms and abbreviations used in DataInterchange and WebSphere Business Integrator, refer to the glossaries in the following documents.

- IBM Global Services, *DataInterchange Administrator's Guide* (SB34-2002)
- WebSphere Business Integrator, *Concepts and Planning*

## How to send your comments

IBM welcomes your comments. You can send your comments by any one of the following methods:

1. Electronically to the network ID listed below. Be sure to include your entire network address if you wish a reply.

Internet: `mailto: idrcf@hursley.ibm.com`

2. By FAX, use the following numbers:

UK: 01962-842327

Other countries: +44-1962-842327

3. By mail to the following address:

User Technologies  
Mail Point 095  
IBM United Kingdom Laboratories  
Hursley Park  
Hampshire  
S021 2JN  
United Kingdom

## Documentation conventions

Throughout this book, the following type conventions distinguish different elements of text:

<b>bold</b>	User input at the command prompt or in an entry field
plain text	Window, screen, and dialog box titles, file names, directories, and keyboard key names
monospaced	Output to the display

**About this book**

---

## Chapter 1. Introduction

IBM WebSphere Business Integrator is in a category of business software referred to as an e-business process management (e-BPM) system. An e-BPM system enables companies to create, execute, and manage the processes and supporting components that span diverse applications, enterprises, and people, providing a single, extensive, flexible solution.

An e-BPM system can reduce the time, complexity, and cost of implementing enterprise business process solutions. After solutions have been implemented, companies are able to rapidly change their applications and processes to meet new business needs.

WebSphere Business Integrator brings together the required elements necessary for an e-BPM system. It enables companies and systems integrators to efficiently architect and implement end-to-end, process-based solutions.

WebSphere Business Integrator comes with the elements needed to create, execute, and manage business processes. It incorporates the capabilities of the IBM middleware products used in a business integration scenario. It works with these underlying IBM products to choreograph the business process lifecycle. This approach delivers content to individuals based on their roles and on the state of the overall business process.

In certain configurations of WebSphere Business Integrator, DataInterchange acts as WebSphere Business Integrator's EDI gateway.

For more information about WebSphere Business Integrator, see WebSphere Business Integrator *Concepts and Planning*.

---

### DataInterchange gateway

The DataInterchange gateway, one of the components of the WebSphere Business Process Integration Platform, is a customized version of the DataInterchange licensed software package Version 3 Release 1 for OS/390.

---

# DataInterchange

The DataInterchange licensed product is a single application that reformats data for electronic transmission, and it includes the following features.

## Flexible setup and administration

- Online customization of standards, maps, and trading partner relationships
- Mapping designed to support:
  - Literals/constants
  - Accumulators, arithmetic and logical operations
  - Qualified loop and element mapping
  - Hierarchical loop mapping
  - Envelope field mapping
  - User exits at the field level
  - User-defined translation and validation tables
  - Boolean logic
- Maps can be used by one or more trading partners
- Export/import available to move user data between test and production systems

## Superior translation capability

- Syntax checking
- Test and production support
- Ability to translate and envelope separately
- Flexible command language interface
- Interactive, batch, event-driven, and real-time processing
- Automatic generation of functional acknowledgments

## Versatile communications

- Support for networks and direct connections to trading partners
- Ability to resend individual transactions or entire envelopes
- Support for MQSeries® Queues as a means of exchanging data between trading partners

## Extensive reporting and auditing

- Reporting of trading partner relationships, including the transaction set being used, and the last communication with a trading partner, and others
- Reporting of envelope and transaction status for both online and batch processing
- Exception reporting
- Setting acceptable error levels for the trading partner/map combination
- Reporting of SAP status for online and batch processing
- Optional audit log with archive recovery capability

## Standards support

- Multiple standards, including EDIFACT, X12, ODETTE, TDCC (EDIA), UCS, UNTDI/TRADACOMS, and VICS
- Multiple versions and releases of standards
- Electronic standards distribution to speed delivery of new standards
- Ability to migrate a map from one version/release of a standard to another, or from one transaction to another
- Online creation and customization of standards
- Full standards compliance checking (user option)

### Additional features

- Designed for high throughput and performance
- Support for concurrent users and applications
- Support for shared Trading Partner's profile (Minimal Trading Partners)
- Ability to process in multiple environments
- NT implementation
- Support for encryption and authentication
- Application program interface (API) to integrate directly with your application
- Java program interface to integrate with MQAO-based applications
- Use of the Security Access Facility (SAF) to establish system security down to the record level
- Graphical User Interface (GUI) to simplify the management of profiles, standards, data formats, and maps
- Support for MQSeries Queues and MQSeries Service MQSS

---

## Chapter 2. Installation

WebSphere Business Integrator DataInterchange for Windows NT includes an install program that guides you through installation. Close any applications you have running and make sure you have enough space on the hard drive on which you are installing this program. The minimum recommended space is 25 MB.

Manual configuration of MQAK is necessary.

After installation and configuration, verification tests are executed. Installation Verification communicates with the DataInterchange gateway by means of a Java API, which binds to the server to check that the gateway is installed and working. A return code displays to show Installation Verification that the gateway is working.

---

## Requirements

The following items are required to install and use WebSphere Business Integrator DataInterchange Version 2.1:

- DB2®, MQAK, MQSeries, and JRE are installed and some configuration is done prior to DataInterchange Adapter installation.
- The DataInterchange Adapter Java package is installed (unzipped) into the target folder resolved by the installation wrapper.
- The WebSphere Business Integrator installation wrapper is set up with the CLASSPATH of the target directory.
- DataInterchange Client is installed.
- Text or XML editor of choice is installed.
- User ID for network is available.
- DTDs are available at the installation site.

---

# Installing WebSphere Business Integrator DataInterchange V2.1

The following instructions guide you through the installation of WebSphere Business Integrator for DataInterchange V2.1.

1. From the product CD, run the command file `bizInstall.cmd`. This installs the base components of WebSphere Business Integrator DataInterchange. The last panel notifies you that the following products have to be installed manually:
  - DataInterchange Client
  - DataInterchange Server

**Important:** The DataInterchange Server must be installed first.

2. From the product CD, run `Server\setup.exe` to install the server.
3. From a DOS prompt, run the command file `configure.bat`. This configures MQ Series, LDAP, and DB2.
4. Use the DB2 Client Configuration Assistant to register the following databases for ODBC:

CONFIG32

EDIEC31E

EDICT31E

5. The server install is now complete. From the product CD, run `Client\Setup.exe` to install the DataInterchange client. Setup install MDAC 2.5 and then requests a reboot of the machine. The DataInterchange CD must remain in the machine. Removing the CD before the reboot completes causes the installation to fail. Reboot the machine, the setup program resumes and completes the installation of the DataInterchange Client.
6. After the Client install is complete, review the directory structure to verify it matches the following table:

<code>\bin</code>	<code>\runtime\dtd</code>
<code>\bnd</code>	<code>\runtime\edi</code>
<code>\config</code>	<code>\runtime\fak</code>
<code>\data</code>	<code>\runtime\prt</code>

\ddl	\runtime\qry
\help	\runtime\rpt
\lib	\runtime\trk
\runtime	\runtime\wrk
\runtime\aex	\runtime\xex
\runtime\app	\runtime\xml
\runtime\dicmd	\samples

The installation of the WebSphere Business Integrator for DataInterchange V2.1 is now complete.

# Installation

---

## Chapter 3. Using Client with WebSphere Studio

As part of an overall solution using the DataInterchange programs, you must define data definitions, standards, maps, channels, and trading partners to control the flow and structure of business documents exchanged between two trading partners.

The DataInterchange Client (the Client) and the DataInterchange DTD Conversion Utility (DTDConvert) are used to create these definitions.

You can use the text editor of your choice to create DataInterchange PERFORM command files, and you can associate all DataInterchange artifacts to text editors using the Tool Registration feature of WebSphere Studio. The association has no effect outside of WebSphere Studio and will not interfere with other tools.

The Client is used to define a DataInterchange transaction in WebSphere Studio. The data generated from the Client editor is stored in a project directory and in a central repository.

DTDConvert is used to create a DataInterchange standard based on XML. You must provide the DTD for DTDConvert to use. You can create and edit a customized DTD, or use an industry standard. DTDConvert produces an XML DataInterchange Dictionary file (.DIC) and an export/import file (.EIF) for importing into the Client. The DIC, which is not editable, is compiled from the DTD.

You can then import the standard into the Client for incorporation into an overall DataInterchange Solution in an EIF.

Solution Manager is used to deploy the completed solution, copying the artifacts as a single ZIP file to the correct location before sending the request for deployment.

## DataInterchange solution artifacts

### Export/Import files (.EIF)

Use the Client to create all the definitions required for channel connections, trading partners, maps, and standards. Once the definitions are completed, they can be exported into the solution Export/Import file (.EIF). The EIF can later be imported into the Client for further editing and then re-exported. Whenever the Client exports definitions (new or changed) into an existing EIF, they are appended to the file, and the header record in the file is time-stamped. No information is deleted or overwritten. Every time an EIF is opened, DataInterchange reads the entire file and uses the latest definitions to control processing.

Use DTDCConvert to convert XML DTDs into artifacts understood by the Client and by the DataInterchange run-time server. DTDCConvert reads a DTD and outputs a DIC file and a temporary EIF. The EIF is imported into the Client so the new XML-based standard defined in the EIF can be used in development and is exported back into WebSphere Studio as part of the solution. The solution EIF is deployed to run time by publishing it, which feeds the EIF definitions into the DataInterchange run time database.

### Dictionary files (.DIC)

When you use DTDCConvert to convert XML DTDs into artifacts understood by the Client and the DataInterchange run-time server, DTDCConvert creates a dictionary file (.DIC) and a temporary EIF. The DIC file is used during DataInterchange run-time processing to correlate the XML element and attribute names with EDI standard segments and data elements. The DIC file is deployed to run time with the associated DTD by copying them to an installation directory.

### PERFORM command files (.DICMD)

The DataInterchange Utility provides command-level access to DataInterchange services through the DataInterchange Adapter. Command files (.DICMD) are plain text files containing DataInterchange PERFORM commands. Solutions will typically consist of several DataInterchange command files, each containing numerous commands directing document processing, data management, report generation, and so forth. You can use command files already available or

## Using Client with WebSphere Studio

create new ones for this project. Use a text editor to create and maintain these files. The DataInterchange Adapter can execute the DICMD files by using the command file name defined at the top of the file. DICMD files are deployed to run time by copying them to an installation directory.

To allow setting PERFORM command parameters at run time, the command message and command files are scanned for substitution parameters. Parameters that accept substituted values are identified by ePICHeaderField names prefixed with a percent sign (%); these are listed below. Whenever the parameter is encountered in the DICMD file, the value of the corresponding ePICHeaderField is substituted before the commands are submitted to the DataInterchange server for execution.

%SessionID
%TransactionID
%BodyType
%SourceLogicalID
%RespondToLogicalID
%BodyCategory
%DeliveryCategory

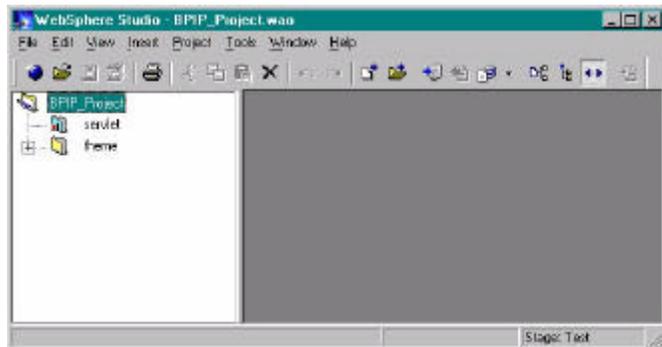
### Building a solution

---

To build a Solution, start by creating a DataInterchange Project folder in WebSphere Studio. All Solution artifacts are maintained in this folder.

1. To build a solution for deployment, start both WebSphere Studio and the Client.
2. In WebSphere Studio, create a Project folder.

Select File>New Project and enter the name of the Project folder to be created.

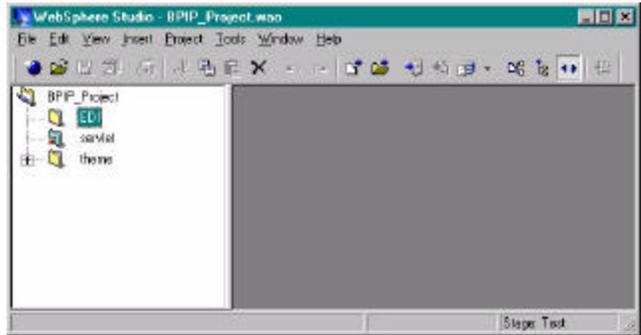


3. In the Project folder, create another folder to hold the DataInterchange files you will build for this solution using the Client and DTDCConvert.
  - a. Select the Project folder and right-click to access the menu.
  - b. Select Insert>Folder and enter the name of the Solution folder to be created.
4. Import the DTDs to be used for this Solution.
  - a. Right-click on the Solution folder and select Insert>File. A pop-up window displays a list of available DTDs. Use industry-defined standards or standards created especially for this project.

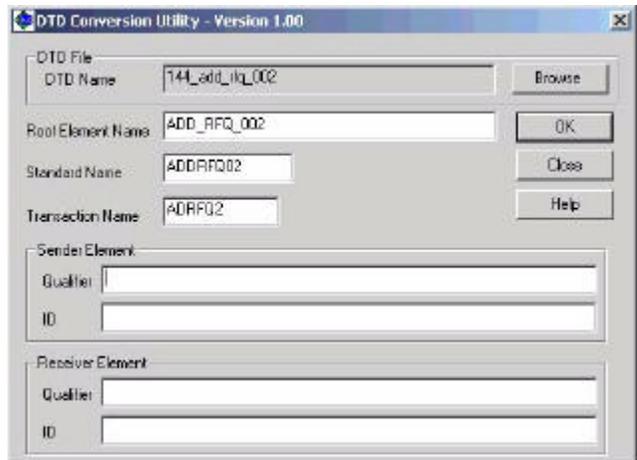
## Using Client with WebSphere Studio

- b. Select a DTD, and then click OK.

The DTD displays as checked out in the navigator window below the Project folder. Repeat this step to add additional DTDs to the folder.



5. Convert the DTDs into DataInterchange EIF and DIC files.
  - a. Start DTDCConvert. A window displays where you can select the DTD file from the Checked-Out folder.



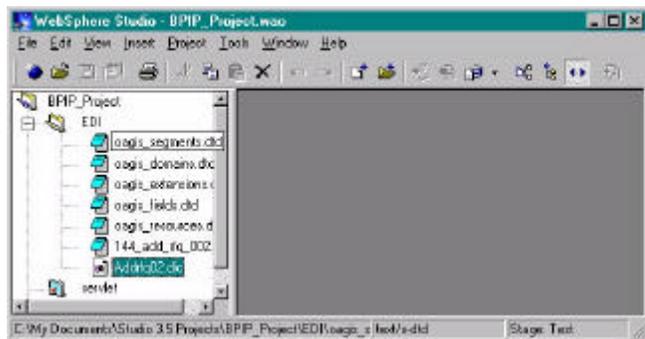
- b. Use the Browse button to locate and select the DTD.
- c. Enter Root Element Name, Standard Name (8 characters), and Transaction Name (6 characters). If you do not enter names in these fields, DTDCConvert takes the information from the DTD. If the field names taken from the DTD, or entered by you, are longer than the maximum field lengths, DTDCConvert truncates the names to fit.

## Using Client with WebSphere Studio

- d. When you are finished, click OK.

The DTD is read, the standard data is written to the EIF, and the data dictionary information is written to the DIC. The Execution Status window displays, showing that both files have been created. This EIF is a temporary file created for importing to the Client, and it is replaced by a Solution EIF later in this process. Do not insert this temporary file into WebSphere Studio or deploy it directly to the DataInterchange run-time server. The DIC file is inserted into WebSphere Studio for later deployment as part of the Solution. You may use the same name for the DIC file that you used for the DTD (in the same Project folder). If you do so, make sure these files are checked out before deployment.

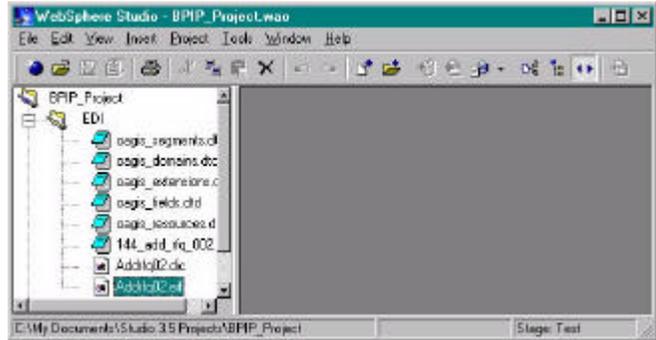
- e. Exit from DTDCConvert.
6. Using WebSphere Studio, import the DIC and EIF files.
    - a. Import the DIC file into WebSphere Studio.
      - 1) Right-click on the Solution folder and select Insert>File. A pop-up window displays a list of the solution files including the DIC file created with DTDCConvert in Step 5d above.



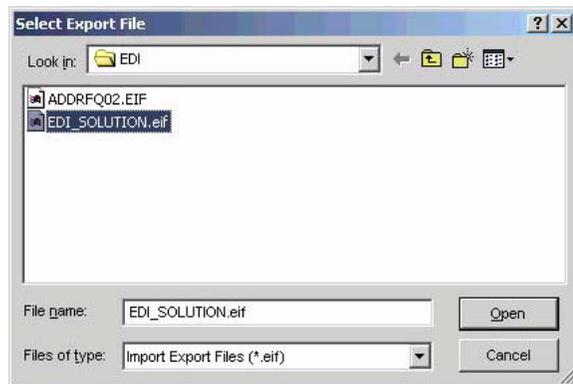
- 2) Select the DIC file to import and click OK.
- b. Import the EIF file generated by DTDCConvert (in Step 5d above) into the Client.

## Using Client with WebSphere Studio

- 1) From the Client menu, select File>Open Import File. A pop-up file selection window displays the list of checked-out EIFs available in WebSphere Studio for importing.

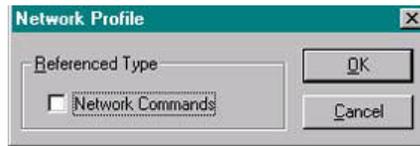


- 2) Double-click the EIF to import. The Client displays a warning message that the EIF is empty.
  - 3) Click OK and continue. The Execute Status window displays. The DTD is now defined to DataInterchange and can be used for creating maps.
7. In the Client, create the definitions needed to deploy the solution. When finished, save the definitions.
  8. Export the definitions into an EIF for deployment.
    - a. Select Actions>Export to File from the File menu. A pop-up window displays the list of available EIFs.

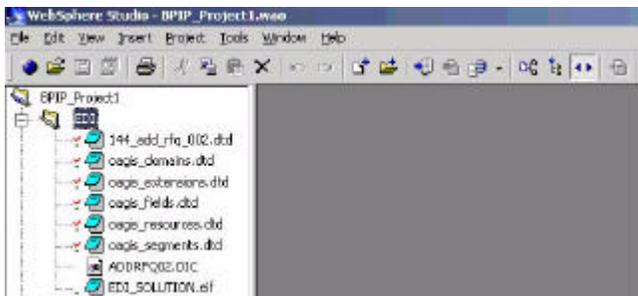


## Using Client with WebSphere Studio

- b. Select the target EIF. Another pop-up window displays where you can select the definitions to be exported. The data is exported into the checked-out EIF, creating the Solution EIF.



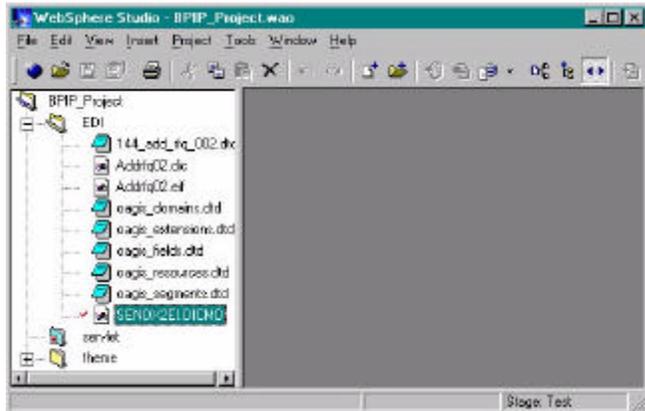
- c. Exit from DataInterchange Client.
9. Since this is the first time this EIF will be used as part of a Solution, you must insert it into the Solution folder.
  - a. Using WebSphere Studio, right-click on the Solution folder and select Insert>File. A pop-up window displays the list of available EIFs.
  - b. Select the EIF and click OK. The EIF displays in the navigator window below the Solution folder.



10. Check in the EIF. Checking in a file saves the contents and removes it from the checked-out directory.
  - a. Select the EIF, and then select Project>Check In. All additional Client work such as new or changed DTDs, maps, standards, trading partner profiles, and application data formats will be placed in this EIF.

## Using Client with WebSphere Studio

11. Insert the DataInterchange PERFORM command files into the Solution folder. Select Insert>File>Create New. A pop-up window displays where you can enter a name for the file with the file extension .DICMD. Ignore the warning about the extension type and continue. WebSphere Studio creates the file and marks it as checked out.



12. Associate an editor with this file.
  - a. Double-click on the file. The Edit with... window displays.
  - b. Select Other to select an editor to associate with the file. The Open File with.. window displays.
  - c. Click OK. The new file opens and you can type the commands required for this solution.
  - d. Save the file and exit the editor.
13. Deploy a completed Solution, or an update to an existing Solution, to the run-time environment using the Publishing Wizard. During the publishing process, EIFs are imported, DIC and DTD files are moved to the DataInterchange dictionary directory, and DICMD files are moved to the DataInterchange command files directory.

# Using Client with WebSphere Studio

---

## Appendix A. Messages and codes

---

### Error processing

When a message is processed, return codes are generated by both DataInterchange and by the WSBI. Return codes are written to an event log that can be queried using a PRINT EVENT LOG command. You can review a single transaction by specifying the Transaction ID or review a batch of transactions by specifying the Batch ID. You can also query the Transaction Store for specific transactions by using a PRINT TRANSACTION DETAILS command and specifying the Transaction ID.

Although all successful processing codes are written to the event log, only exception codes for serious errors are written to the event log. Sufficient information is provided to allow you to find further information in the DataInterchange publications.

The return codes generated by the WSBI are as follows:

Code	Definition
0-1	Processing was successful.
5	A terminal error occurred and the Solution Manager was shut down. Check the error log to see which error occurred. For more information on error codes, see Appendix B or refer to the DataInterchange Programmer's Reference.
6 or higher	A terminal error occurred and the Solution Manager was shut down. Check the error log to see which error occurred. For more information on error codes, see Appendix B or refer to the DataInterchange Programmer's Reference.

The exception codes consist of three separate codes:

- The API return code (documented below), usually indicates a programming error
- The Severity Code

## Messages and codes

- The Extended Code (documented in the *DataInterchange Programmers Reference*)

Code	Definition
0	Good return.
1	Not used.
2	The request has been queued to a translation server.
3	Not used.
4	There was a problem with the service director. Check the event log. Then call GetRetCode() and GetExtRetCode() to find more information.
5	There was a problem executing the request. Check the event log. Then call GetRetCode() and GetExtRetCode() to find more information.
6	CDIMsgQueue::Open - Invalid router queue type. Valid types are file and pipe.
7	Check errno for more information.
8	CDIMsgQueue::Open - Queue name is NULL.
9	CDIMsgQueue::Open - Queue already open.
10	CDIMsgQueue::Open - Open on queue failed. Check errno for more information.
11	CDIMsgQueue::Write - The file descriptor is invalid.
12	CDIMsgQueue::Write - Attempted write to router queue failed. Check errno for more information.
13	CDIEnvironment::Parse - argc/argv parameters are invalid.
14	CDIEnvironment::Parse - Command line parameters to the ediservr program are invalid.

## Messages and codes

Code	Definition
15	CSyncTranslator::Initialize - Translator already initialized. Initialize only once.
16	CSyncTranslator::Initialize - An attempt was made to initialize the translator without specifying a DB2 plan. Set the plan before initializing.
17	CSyncTranslator::Initialize - Failed to initialize on call to service director. Check the return code and extended return code by calling the GetRetCode() and GetExtRetCode() methods.
18	CSyncTranslator::ProcessRequest - A call was made to process a request, but the translator has not been initialized yet.
19	CSyncTranslator::Initialize - A call was made to process a request, but there is no request to process.
20	CSyncTranslator::Initialize - The unit of work specified for this request is invalid.
21	CSyncTranslator::SetFileName - The mandatory parameter logical filename is NULL.
22	CSyncTranslator::SetFileName - The mandatory parameter logical filename is an empty string.
23	CSyncTranslator::SetFileName - The mandatory parameter logical filename is >8 characters long.
24	CSyncTranslator::SetFileName - Error calling the service director to set the fully qualified name of one of the files used in a request.
25	CSyncTranslator::GetFileName - The mandatory parameter logical filename is NULL.
26	CSyncTranslator::GetFileName - The mandatory parameter logical filename is an empty string.

## Messages and codes

Code	Definition
27	CSyncTranslator::GetFileName - The mandatory parameter logical filename is >8 characters long.
28	CSyncTranslator::GetFileName - The service director was not able to get the fully qualified name of one of the files used in a request.
29	CSyncTranslator::GetFileName - Error calling the service director to get the fully qualified name of one of the files used in a request.
30	CSyncTranslator::Terminate - A call was made to terminate the translator, but the translator has not been initialized yet.
31	CAsyncTranslator::Initialize - An attempt was made to initialize an asynchronous translator in a WIN32 environment. This is not supported at this time. Use CRemote Translator instead.
32	CAsyncTranslator::Initialize - An attempt was made to initialize an asynchronous translator that has already been initialized.
33	CAsyncTranslator::Initialize - The asynchronous translation server is already running.
34	CAsyncTranslator::Initialize - Error creating the command pipes to communicate with the asynchronous translation server. Check errno for more information.
35	CAsyncTranslator::Initialize - Error attempting to fork() to create the asynchronous translation server. Check errno for more information.
36	CAsyncTranslator::Initialize - Failed to load the program ediservr. Make sure it is in the PATH for executables. Check errno for more information in the child process only.

## Messages and codes

Code	Definition
37	CAsyncTranslator::ProcessRequest - The translator has not been initialized yet and a call was made to ProcessRequest(). Before processing any requests, the translator must be initialized.
38	CAsyncTranslator::ProcessRequest - A call was made to process a request, but there is no request to process. Check the request type and PERFORM command.
39	CAsyncTranslator::ProcessRequest - The request has NOT been queued. The maximum number of requests have already been queued to this translation server. Select (or start) another translation server, wait for this server to finish its current request, or increase the max_requests parameter to allow more requests to be queued.
40	CAsyncTranslator::Write - An error occurred attempting to send a command to the translation server. Verify that the translation server is running. Check errno for more information.
41	CAsyncTranslator::UpdateCurReqCnt - An error occurred attempting to determine if there is any data in the response queue. Verify that the translation server is running. Check errno for more information.
42	CAsyncTranslator::UpdateCurReqCnt - An error occurred attempting to determine if there is any data in the response queue. Select returned an invalid code.
43	CAsyncTranslator::UpdateCurReqCnt - An error occurred attempting to read a response from the response queue. Verify that the translation server is running. Check errno for more information.
44	CAsyncTranslator::SetFileName - A call was made to send a SET command to the translation server, but this is the server.

## Messages and codes

Code	Definition
45	CAsyncTranslator::StartTranslator - A call was made to send start commands to the translation server, but this is the translation server.
46	CAsyncTranslator::StartTranslator - The router type specified in the CDIRRequest is invalid. Valid types are file, pipe, socket, and email.
47	CAsyncTranslator::StartTranslator - The unit of work specified in the CDIRRequest is invalid. Valid types are envelope, transaction, and no commit.
48	CAsyncTranslator::Terminate - A call was made to send a terminate message to the translation server, but this is the translation server.
49	CAsyncTranslator::Terminate - A call was made to terminate the translator, but the translator has not been initialized yet.
50	CRemoteTranslator::Initialize - An attempt was made to initialize a remote translator that has already been initialized.
51	CRemoteTranslator::Initialize - An attempt was made to resolve the host name and an error occurred. Most likely, the host was not found. Verify the host name is correct and accessible from the translation system. Check errno for more information.
52	CRemoteTranslator::Initialize - Failed to create the socket to communicate with the translation server. Check errno for more information.
53	CRemoteTranslator::Initialize - Failed to connect to the remote translator. Verify that the remote translator daemon is running and that the host name is correct. Check errno for more information.

---

## Appendix B. Programmer's reference

In this appendix, you will find the control record format, header field descriptions, and scenarios for using WebSphere Studio.

---

### Control record format

The control record provides the DataInterchange server with the information needed to process an ADF message. This record has a fixed format and is the first record in the ADF file. PERFORM command values, if specified, override the values set in the control record and the DataInterchange API. The Adapter will write the control record using the following field layout.

DataInterchange field name	Type	Length	WebSphere field name	Description
RECID	Char	4	WSBI	Record ID
INTID	Char	35	ExternalID	EDI Trading Partner Name
ACFIELD	Char	35	TransactionID	Application Control Number (defined in the ADF map)
FORMAT	Char	16	BodyType	ADF Name (format ID)
APPSNDID	Char	25	InternalID	Application Trading Partner
BATCH	Char	8	SessionID	Batch ID
CMDFILE	Char	32	ProcessingCategory	DataInterchange Command File Name
REQID	Char	16	DeliveryCategory	Requestor ID

## MQAO message header contents

The following fields are included in the message header for both sending and receiving messages:

### BodyCategory

The type of data being received. Valid values are:

**adf** Application Data format

**other** Other value such as DI\_Command or a DTD name

### BodyType

If MessageType is **GatewayMessage**, the value in this field is the name of the Message Definition (format ID) as defined in DataInterchange.

If MessageType is **GatewayCommand**, the value in this field is the data type expected to be returned. Valid values are:

**rpt** Report

**app** Application

**edi** EDI data

**qry** Query

**xml** XML data

**none** No data

### DeliveryCategory

The Requestor ID. Used to retrieve, authenticate, and establish channel connectivity to the external VAN. If this field is empty or not defined, then the Internal ID is used.

### DestinationLogicalID

The identifier used to route the message to the correct WSBI component. Valid values are:

**DI**

The message is for the DataInterchange WSBI Adapter.

**none**

The message is for the Message Bus.

### ExternalID

The EDI Trading Partner name or number, such as a DUNS number.

### InternalID

The Application Trading Partner name or number.

### Message

If Message Type is **GatewayMessage**, then this field contains business data in ADF or XML format.

If Message Type is **GatewayCommand** and Body Category is **DI\_Command**, then this field contains a series of DataInterchange PERFORM commands.

### Message Type

The type of message. For a message sent to DataInterchange from the Message Bus, valid values are:

#### **GatewayMessage**

This message contains data for translation processing by the DataInterchange server.

#### **GatewayCommand**

This message contains a DataInterchange PERFORM command string.

For a message received from DataInterchange to the Message Bus (synchronous reply), valid values are:

#### **GatewayMessageReply**

This message contains translation/receive data in reply to a Send Wait MQAK method.

#### **GatewayCommandReply**

This message contains response data in reply to a SendWait. The Message Type is **GatewayCommand**.

### ProcessingCategory

If Message Type is **GatewayCommand** and Body Category is **DI\_Command**, then the value in this field is the logical identifier of the command string contained in the message.

If Message Type is **GatewayMessage**, then the value in this field is the command procedure file that the Adapter will open and pass to DataInterchange for execution.

## Programmer's reference

### RespondToLogicID

If the message is destined for the DataInterchange Gateway, then the value in this field is **DI**.

If the message is in reply from the DataInterchange Gateway, then the value in this field is the originator ID used for routing the reply back from DataInterchange.

### SourceLogicID

If the message is destined for the DataInterchange Gateway, then the value in this field is the originator ID used for routing the reply back from DataInterchange.

If the message is in reply from the DataInterchange Gateway, then the value in this field is **DI**.

### SessionID

The Batch ID. Used for collecting multiple messages.

### TransactionID

The Application Control Field. Used for the DataInterchange Transaction Correlation ID.

---

## Output files

DataInterchange creates several output files during processing. The names and locations of these files depend on the PERFORM commands executed. These files are named using the following convention:

Concatenate the character 'D' with the Batch\_ID (session ID) and with the App Control Number (transaction ID), "D#", where #={Batch\_ID+App Control Number. For example, D111887766.

All files created and used by DataInterchange will have the D# prefix and a 3-character suffix.

---

## Usage scenarios

### Receiving an application data format message from the BFM application

The BFM application can send application data (ADF) to DataInterchange for translating and sending to an EDI trading partner by setting the ePIC header fields and putting the application data into the message as described below.

#### Message and header contents:

*MessageType*=“GatewayMessage”

*SourceLogicalID*=Originator ID (such as “BFM”)

*ExternalID*=EDI Trading Partner Name

*DestinationLogicalID*=“DI” (message routed to DataInterchange gateway)

*TransactionID*=Application Control Number

*BodyCategory*=“ADF”

*BodyType*=ADF Name (Format ID)

*RespondToID*=SourceLogicalID

*InternalID*=Application Trading Partner

*SessionID*=Batch ID

*ProcessingCategory*=DataInterchange Command File Name

*DeliveryCategory*=Requestor ID

*Message*=Application Data

If the user application will query the data using a transaction identifier whose value is contained within the data (such as a purchase order number), then the transformation map should define the Application Control number for that identifier.

In your command files you may include the TRANSLATE AND SEND command which combines the TRANSLATE TO STANDARD, ENVELOPE, and SEND commands. This is the simplest way that WSBI applications can use DataInterchange

## Programmer's reference

to translate and transmit documents to trading partners. The Requestor ID, by default, will have the same name as the Application Trading Partner (Application Sender ID) if DeliveryCategory is absent.

```
PERFORM TRANSLATE AND SEND WHERE APPFILE(APPFILE)
```

```
REQID(%DeliveryCategory) BATCHSET(%SessionID)
```

## Sending an XML data message from BFM for translation to EDI and sending

### Message and header contents:

*MessageType*=“GatewayMessage”

*SourceLogicalID*=Originator ID (such as “BFM”)

*ExternalID*=EDI Trading Partner Name

*DestinationLogicalID*=“DI” (message routed to DataInterchange gateway)

*TransactionID*=Application Control Number

*BodyCategory*=XMLSTDID (If no DOCTYPE in the business document)

*BodyType*=Root Element Name

*RespondToID*=SourceLogicalID

*InternalID*=Application Trading Partner

*SessionID*=Batch ID

*ProcessingCategory*=DataInterchange Command File Name

*DeliveryCategory*=Requestor ID

*Message*=XML Business Document

You need two PERFORM commands in the command file to translate the XML business document into EDI format and send it to a trading partner. The commands are issued within the same DataInterchange command file.

**Note:** Unlike the DataInterchange Utility, semicolons are **not** used to separate PERFORM commands).

Issue the following commands to DataInterchange.

```
PERFORM DEENVELOPE AND TRANSLATE WHERE XML(Y)  
FILEID(EDIFILE) APPFILE(APFILE) RAWDATA(Y)  
BATCHSET(%SessionID) XMLDICT(%dtddirectory)  
XMLDTDS(%dtddirectory)
```

The XML has been translated into the ADF.

```
PERFORM TRANSLATE AND SEND WHERE APPFILE(APFILE)  
  
REQID(%DeliveryCategory) BATCHSET(%SessionID)
```

The Document has been translated into EDI and sent.

## Sending a command and receiving response data from DataInterchange

A synchronous MQAK send/wait reply is used to send a command to DataInterchange from BFM, receive response data from DataInterchange, and then return it to BFM. The value in BodyCategory determines the type of response data. Some PERFORM commands do not return data or indicate if errors have occurred. When the value in BodyCategory is **none**, the contents of the D#.prt file are returned in the message body. If the API return code indicates an error, then the first line of the message body contains the literal string "ERROR" followed by the contents of the D#.prt file.

### Message and Header Contents:

*MessageType*="GatewayCommand"

*SourceLogicalID*=Originator ID (such as "BFM")

*ExternalID*=EDI Trading Partner Name

*DestinationLogicalID*="DI" (message routed to DataInterchange gateway)

*TransactionID*=Application Control Number

*BodyCategory*="DI\_command"

*BodyType*="rpt" | "app" | "edi" | "qry" | 'xml' | none

*RespondToID*=SourceLogicalID

*InternalID*=Application Trading Partner

## Programmer's reference

*SessionID*=Batch\_ID

*ProcessingCategory*=Not Used

*DeliveryCategory*=Requestor ID

*Message*=DataInterchange PERFORM Commands

---

## Receiving EDI data and translating it into XML

You need two PERFORM commands in the command file to receive EDI data and translate it into an XML format. The commands are issued within the same DataInterchange command file.

Issue the following commands to DataInterchange:

```
PERFORM RECEIVE AND TRANSLATE WHERE  
REQID(%DeliveryCategory) BATCHSET(%SessionID)  
APPFIL(APFILE) RAWDATA(N)
```

```
PERFORM TRANSLATE AND ENVELOPE WHERE APPFILE(APFILE)  
XMLDICT(directory) XMLDTDS(directory)  
FILEID(EDIFILE)
```

## Solution Manager exception service

The Adapter uses the com.ibm.epic.common package for exception messaging and to generate EpicExceptions for all serious errors encountered during execution. Only exceptions indicative of general gateway function failure are generated, such as errors indicating that a business document could not be translated because the service is unavailable.

DataInterchange diagnostic codes are provided in the Exception log so that the Gateway administrator can refer to the DataInterchange manuals for diagnostic descriptions and to perform problem determination.



## Programmer's reference

```
WRITER MESSAGE LINE 1 TE'  
L02+1::LINE ONE OF THE INVITEDITER LINE 1 TEXT+3::LINE TWO OF THE  
INVITEDITER LINE 1 TEXT+::LINE T  
HREE THE INVITEDITER LINE 1 TEXT'NAD+IV+ LINE  
2:LL'GIS+02:26'GIS+01:15'GIS+Y:16'RFF+LIR+0002'  
RFF+CS1+COSLINE 2'RFF+CS1+COSLINE  
2'RFF+LIN+002'PCD+001:1.0000001'PCD+002:2.0000002'  
PCD+003:3.0000003'PCD+004:4.0000004'PCD+005:5.0000005'L01+034+UNDER-  
WRITER MESSAGE LINE 2 TE'  
L02+1::LINE ONE OF THE INVITEDITER LINE 2 TEXT+3::LINE TWO OF THE  
INVITEDITER LINE 2 TEXT+::LINE T  
HREE THE INVITEDITER LINE 2 TEXT'UNT+39+40000200'UNZ+1+40000092'  
The number of records printed from the input file was 2  
Printing has terminated. Return code is 0; extended return code is 0.
```

## Transaction Store query

You can obtain a list of transactions that meet the selection criteria you specify in the keywords by issuing a **PERFORM PRINT** command. In the returned list, the transactions are identified by their Transaction Store ID.

Issue the following command to DataInterchange:

```
PERFORM PRINT TRANSACTION DETAILS WHERE  
ACFIELD(%TransactionID)
```

---

## Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other 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.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

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

## Notices

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 the 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 United Kingdom Limited  
Intellectual Property Department  
Hursley Park  
Winchester SO21 2JN  
United Kingdom

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 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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples 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.

### COPYRIGHT LICENSE:

This information contains 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 or imply reliability, serviceability, or function of these programs. 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 IBM's application programming interfaces.

---

## Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States or other countries, or both:

- CICS
- DB2
- IBM
- MQSeries
- MVS
- OS/390
- WebSphere

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

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

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





Part Number: BIZAAI00  
Program Number: 5724-A78