

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 31

First Edition (June 2001)

This edition applies to Version 2.1 of the IBM® WebSphere® Business Integrator (program number 5724-A78) DataInterchange for Windows NT®, and to all subsequent release 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

About this book.	v	Appendix A. Messages and Codes.	15
Who this book is for	v	Error processing	15
What you need to know	v	Return Codes	15
Related books.	v	Exception Codes	16
Glossaries	vi	Appendix B. Programmer's reference.	23
How to send your comments	vi	Control record format	23
Documentation conventions	vii	MQAO message header contents	23
Chapter 1. Introduction	1	Output files	25
DataInterchange gateway	1	Usage Scenarios.	26
DataInterchange	1	Receiving EDI data and translating it into	
Flexible setup and administration	2	XML	28
Superior translation capability	2	Solution Manager exception service	28
Versatile communications	2	Event log query.	29
Extensive reporting and auditing	2	Transaction Store query	30
Standards support	2	Notices	31
Additional features	3	Trademarks	33
Chapter 2. Installation	5	Bibliography	35
Requirements	5	IBM WebSphere Business Integrator library	35
Installing WebSphere Business Integrator		Related documentation	36
DataInterchange V2.1	5	WebSphere Partner Agreement Manager	
Chapter 3. Using the DataInterchange Client		library	37
with WebSphere Studio	7	DataInterchange library	37
DataInterchange solution artifacts	7	Other Libraries	37
Export/Import files (.EIF)	8	Index	39
Dictionary files (.DIC)	8		
PERFORM command files (.DICMD)	8		
Building a solution	9		

About this book

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

- “Chapter 1. Introduction” on page 1 provides an overview of this product, supported standards, functions and features.
- “Chapter 2. Installation” on page 5 describes installation requirements and provides instructions on how to perform installation.
- “Chapter 3. Using the DataInterchange Client with WebSphere Studio” on page 7 describes solution artifacts and how to build a solution.
- “Appendix A. Messages and Codes” on page 15 describes error handling and provides a list of DataInterchange return codes.
- “Appendix B. Programmer’s reference” on page 23 describes the control record format, header fields, and provides scenarios for using WebSphere Studio.

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/>

- *IBM 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.

- *IBM 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.
- *IBM 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.
- *IBM 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 guide is for the EDI administrator, the system programmer, and the database administrator.
- *IBM DataInterchange Programmer's Reference*, (SB34–2001)
This book provides information for the DataInterchange application programmer. It describes general-use programming and provides reference information for developing application programs that use DataInterchange.

Other related books include:

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

Glossaries

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

- *IBM DataInterchange Administrator's Guide*.
- *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: idrcef@hursley.ibm.com](mailto:idrcef@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:

- **bold** – 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.

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 the *WebSphere Business Integrator Concepts and Planning* book.

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.

Introduction

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.
- Windows 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.

Introduction

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 disk space is 25 MB.

Manual configuration of MQSeries Adapter Kernel 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[®], MQSeries Adapter Kernel, MQSeries, and a Java Runtime Environment 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 the 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 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

Installation

- 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 MQSeries, 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 installs 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:

Table 1. Post-Installation Directory Structure

\bin	\runtime\dtd
\bnd	\runtime\edi
\config	\runtime\fak
\data	\runtime\prt
\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 WebSphere Business Integrator DataInterchange V2.1 is now complete.

Chapter 3. Using the DataInterchange 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

DataInterchange can create the following solution artifacts:

- Export/Import files (.EIF)
- Dictionary files (.DIC)
- PERFORM command files (.DICMD)

Using the DataInterchange Client

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 the 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 create new ones for your 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 the 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.

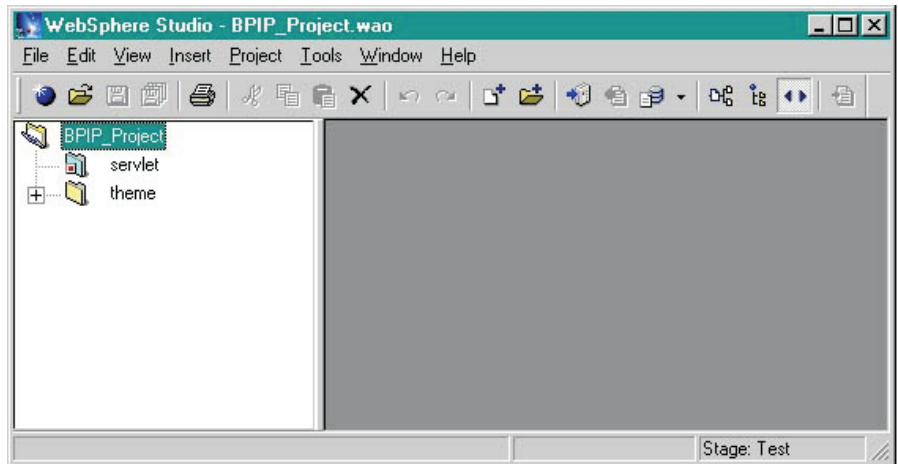
- %SessionID
- %TransactionID
- %BodyType
- %SourceLogicalID
- %RespondToLogicalID
- %BodyCategory
- %DeliveryCategory

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.

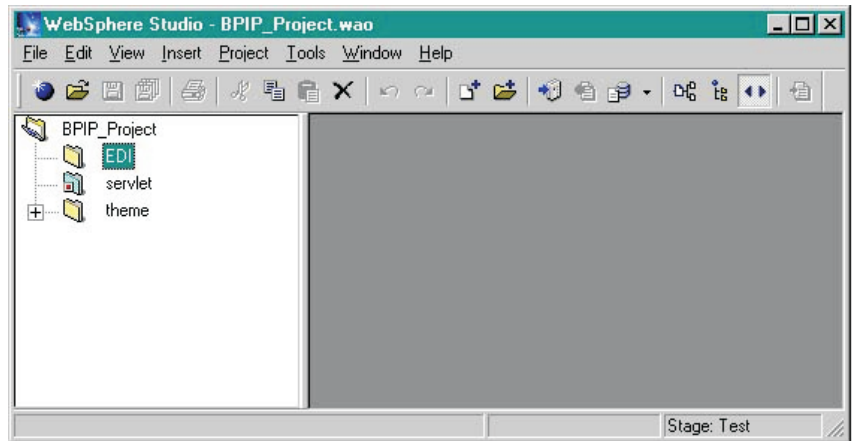


3. In the Project folder, create another folder to hold the DataInterchange files you will build for this solution using the Client and DTDCConvert.
 - a. Select the Project folder and right-click to access the menu.
 - b. Select Insert→Folder and enter the name of the Solution folder to be created.
4. Import the DTDs to be used for this Solution.

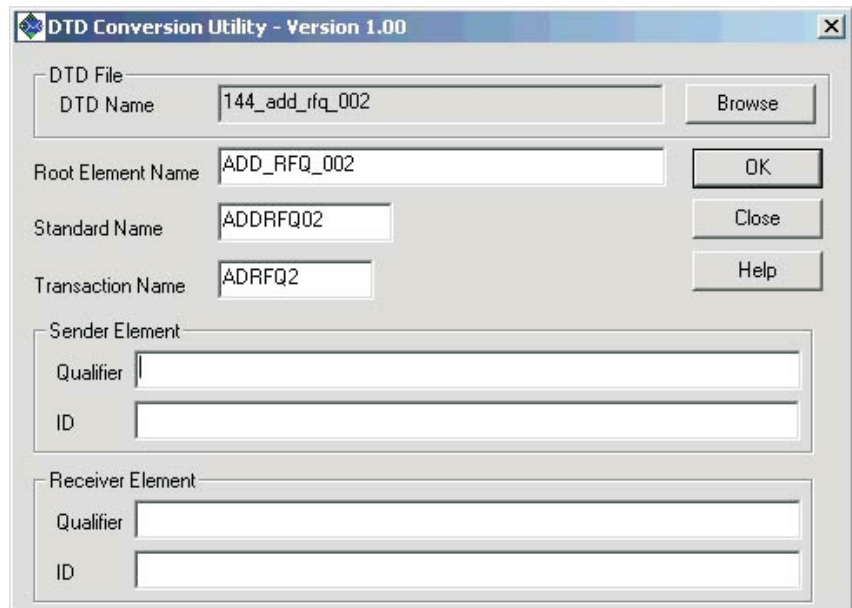
Using the DataInterchange Client

- 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.
- 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.



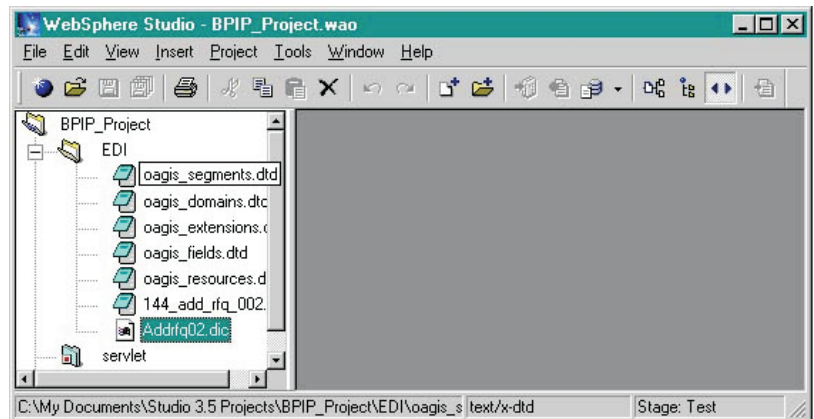
5. Convert the DTDs into DataInterchange EIF and DIC files.
 - a. Start DTDCConvert. A window displays where you can select the DTD file from the Checked-Out folder.



- b. Use the Browse button to locate and select the DTD.
- c. Enter a Root Element Name, a Standard Name (8 characters), and a 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.
- d. When you are finished, click OK.

The DTD is read, the standard data is written to the EIF file, and the data dictionary information is written to the DIC file. 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.

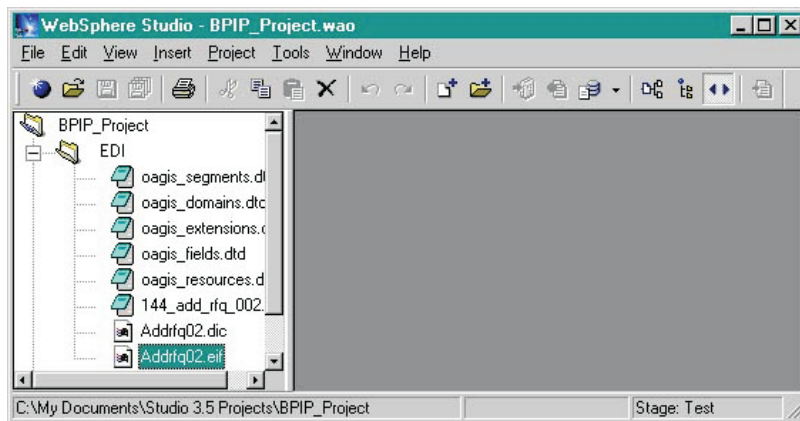
- e. Exit from DTDCConvert.
6. Using WebSphere Studio, import the DIC and EIF files.
- a. Import the DIC file into WebSphere Studio.
 - 1) Right-click on the Solution folder and select Insert→File. A pop-up window displays a list of the solution files including the DIC file created with DTDCConvert in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



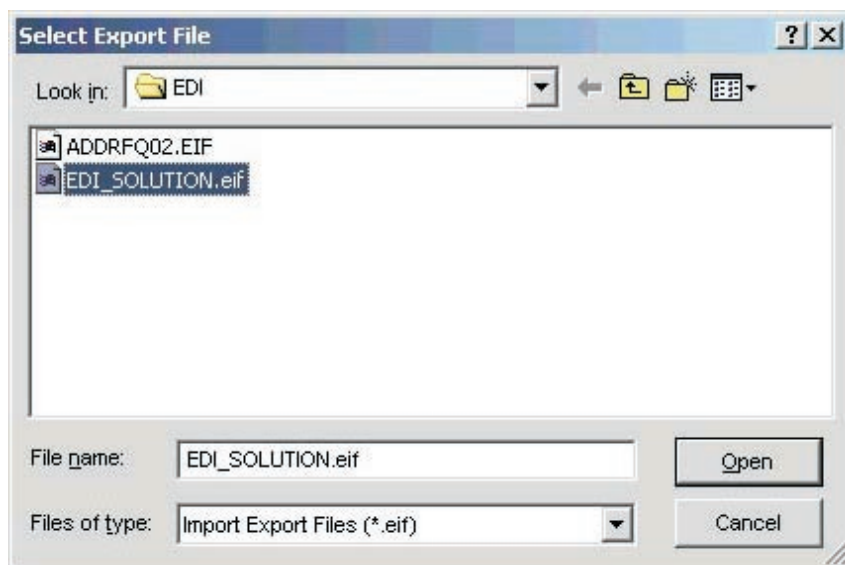
- 2) Select the DIC file to import and click OK.
- b. Import the EIF file generated by DTDCConvert (in Step 5d above) into the Client.
 - 1) From the Client menu, select File→Open Import File. A pop-up file selection window displays the list of checked-out EIFs available in

Using the DataInterchange Client

WebSphere Studio for importing.

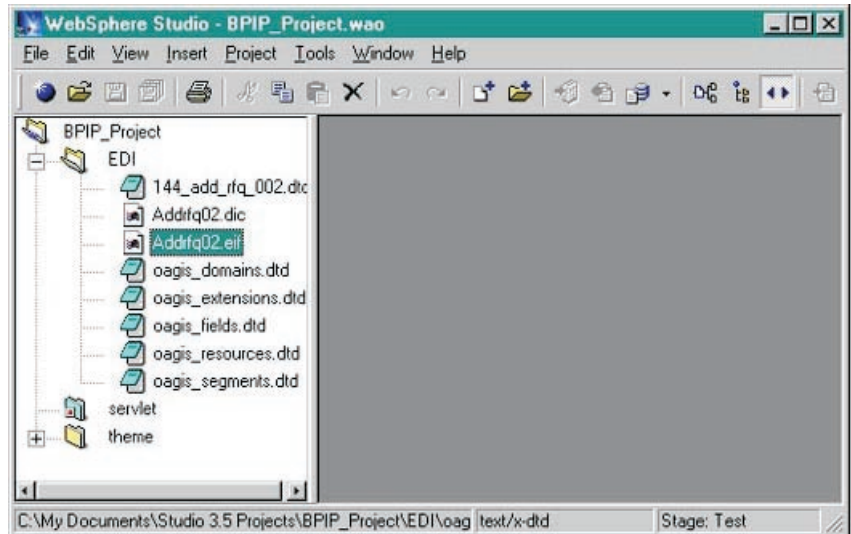


- 2) Double-click the EIF to import. The Client displays a warning message that the EIF is empty.
- 3) Click OK and continue. The Execute Status window displays. The DTD is now defined to DataInterchange and can be used for creating maps.
7. In the Client, create the definitions needed to deploy the 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.

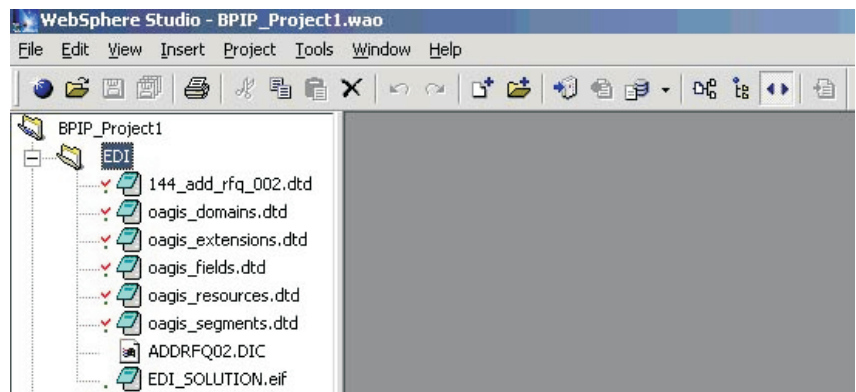


Using the DataInterchange Client

- 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 the 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.

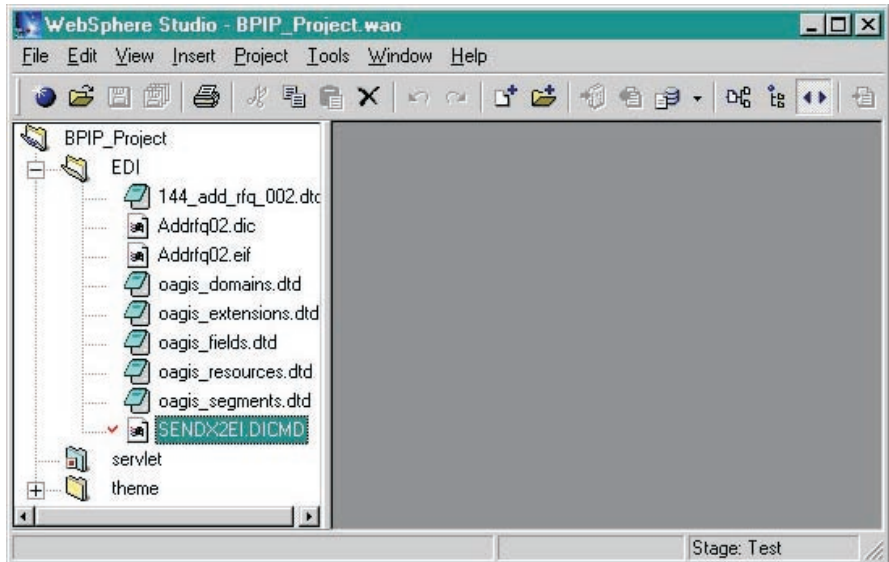


10. Check in the EIF. Checking in a file saves the contents and removes it from the checked-out directory.

Using the DataInterchange Client

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.

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.

Appendix A. Messages and Codes

This appendix describes error processing and descriptions for return codes and exception codes.

Error processing

When a message is processed, return codes are generated by both DataInterchange and by Business Integrator. 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.

Return Codes

The return codes generated by the Business Integrator are as follows:

0 - 1

Explanation: Processing was successful.

5

Explanation: 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 *IBM DataInterchange Programmer's Reference*.

6 or higher

Explanation: 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. Programmer's reference" on page 23 or refer to the *IBM DataInterchange Programmer's Reference*.

Exception Codes

The exception codes consist of three separate codes:

- The API return code (described in the following message list), usually indicates a programming error.
- The Severity Code.
- The Extended Code (documented in the *IBM DataInterchange Programmer's Reference*).

0

Explanation: Good return.

1

Explanation: Not used.

2

Explanation: The request has been queued to a translation server.

3

Explanation: Not used.

4

Explanation: There was a problem with the service director. Check the event log. Then call `GetRetCode()` and `GetExtRetCode()` to find more information.

5

Explanation: There was a problem executing the request. Check the event log. Then call `GetRetCode()` and `GetExtRetCode()` to find more information.

6

Explanation: `CDIMsgQueue::Open` - Invalid router queue type. Valid types are file and pipe.

7

Explanation: Check `errno` for more information.

8

Explanation: `CDIMsgQueue::Open` - Queue name is NULL.

9

Explanation: CDIMsgQueue::Open - Queue already open.

10

Explanation: CDIMsgQueue::Open - Open on queue failed. Check errno for more information.

11

Explanation: CDIMsgQueue::Write - The file descriptor is invalid.

12

Explanation: CDIMsgQueue::Write - Attempted write to router queue failed. Check errno for more information.

13

Explanation: CDIEnvironment::Parse - argc/argv parameters are invalid.

14

Explanation: CDIEnvironment::Parse - Command line parameters to the ediservr program are invalid.

15

Explanation: CSyncTranslator::Initialize - Translator already initialized. Initialize only once.

16

Explanation: CSyncTranslator::Initialize - An attempt was made to initialize the translator without specifying a DB2 plan. Set the plan before initializing.

17

Explanation: 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

Explanation: CSyncTranslator::ProcessRequest - A call was made to process a request, but the translator has not been initialized yet.

19

Explanation: CSyncTranslator::Initialize - A call was made to process a request, but there is no request to process.

20

Explanation: CSyncTranslator::Initialize - The unit of work specified for this request is invalid.

21

Explanation: CSyncTranslator::SetFileName - The mandatory parameter logical filename is NULL.

22

Explanation: CSyncTranslator::SetFileName - The mandatory parameter logical filename is an empty string.

23

Explanation: CSyncTranslator::SetFileName - The mandatory parameter logical filename is >8 characters long.

24

Explanation: CSyncTranslator::SetFileName - Error calling the service director to set the fully qualified name of one of the files used in a request.

25

Explanation: CSyncTranslator::GetFileName - The mandatory parameter logical filename is NULL.

26

Explanation: CSyncTranslator::GetFileName - The mandatory parameter logical filename is an empty string.

27

Explanation: CSyncTranslator::GetFileName - The mandatory parameter logical filename is >8 characters long.

28

Explanation: CSyncTranslator::GetFileName - The service director was not able to get the fully qualified name of one of the files used in a request.

29

Explanation: CSyncTranslator::GetFileName - Error calling the service director to get the fully qualified name of one of the files used in a request.

30

Explanation: CSyncTranslator::Terminate - A call was made to terminate the translator, but the translator has not been initialized yet.

31

Explanation: CAsyncTranslator::Initialize - An attempt was made to initialize an asynchronous translator in a WIN32 environment. This is not supported. Use CRemote Translator instead.

32

Explanation: CAsyncTranslator::Initialize - An attempt was made to initialize an asynchronous translator that has already been initialized.

33

Explanation: CAsyncTranslator::Initialize - The asynchronous translation server is already running.

34

Explanation: CAsyncTranslator::Initialize - Error creating the command pipes to communicate with the asynchronous translation server. Check errno for more information.

35

Explanation: CAsyncTranslator::Initialize - Error attempting to fork() to create the asynchronous translation server. Check errno for more information.

36

Explanation: 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.

37

Explanation: 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

Explanation: 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

Explanation: 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

Explanation: `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

Explanation: `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

Explanation: `CAsyncTranslator::UpdateCurReqCnt` - An error occurred attempting to determine if there is any data in the response queue. `Select` returned an invalid code.

43

Explanation: `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

Explanation: `CAsyncTranslator::SetFileName` - A call was made to send a SET command to the translation server, but this is the server.

45

Explanation: `CAsyncTranslator::StartTranslator` - A call was made to send start commands to the translation server, but this is the translation server.

46

Explanation: `CAsyncTranslator::StartTranslator` - The router type specified in the `CDIRRequest` is invalid. Valid types are file, pipe, socket, and email.

47

Explanation: `CAsyncTranslator::StartTranslator` - The unit of work specified in the `CDIRRequest` is invalid. Valid types are envelope, transaction, and no commit.

48

Explanation: `CAsyncTranslator::Terminate` - A call was made to send a terminate message to the translation server, but this is the translation server.

49

Explanation: CAsyncTranslator::Terminate - A call was made to terminate the translator, but the translator has not been initialized yet.

50

Explanation: CRemoteTranslator::Initialize - An attempt was made to initialize a remote translator that has already been initialized.

51

Explanation: 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

Explanation: CRemoteTranslator::Initialize - Failed to create the socket to communicate with the translation server. Check errno for more information.

53

Explanation: 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.

Messages and Codes

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.

Table 2. Control record fields

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

Programmer's Reference

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 Business Integrator component. Valid values are:

- DI* – The message is for the *DataInterchange 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 *MessageType* is *GatewayMessage*, then this field contains business data in ADF or XML format.

If *MessageType* is *GatewayCommand* and *BodyCategory* 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 SendWait MQSeries Adapter Kernel method.

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

ProcessingCategory

If MessageType is *GatewayCommand* and BodyCategory is *DI_Command*, then the value in this field is the logical identifier of the command string contained in the message.

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

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.

Programmer's Reference

Usage Scenarios

Receiving an application data format message from the Business Flow Manager application

The Business Flow Manager 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 = Business Flow Manager 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 Business Integrator applications can use Business Flow Manager 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 Business Flow Manager 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)
XMLDIDS(%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 MQSeries Adapter Kernel send/wait reply is used to send a command to DataInterchange from the Business Flow Manager, receive response data from DataInterchange, and then return it to Business Flow Manager. The value in BodyCategory determines the type of response data. Some PERFORM commands do not return data or indicate if errors have

Programmer's Reference

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

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. See "DataInterchange library" on page 37.

Event log query

DataInterchange maintains an event log that can be queried with a PERFORM command.

To query for a single transaction, issue the following command to DataInterchange::

```
PERFORM PRINT EVENT LOG WHERE ACFIELD(%TransactionID)
```

To query for a batch series, issue the following command:

```
PERFORM PRINT EVENT LOG WHERE BATCH(%SessionID)
```

Sample Report:

```
EVENTS LOGGING REPORT
REQUESTOR . . . . . : USER21
DATE . . . . . : 97/12/14
TIME . . . . . : 08:37:36
Selection Criteria for Log Report
Batch ID . . . . . : 072912
Date . . . . . :
Time . . . . . :
User ID . . . . . :
Format ID . . . . . :
Associated Entry ID . . . : 19970721164325210000
EVENTS LOGGING REPORT DATE: 97/12/14 TIME: 08:37:36 Page 1
Date . . . . . : 93/07/21
Time . . . . . : 16:43:26
Entry ID . . . . . : E00000000000000000000000000000000000000000000000019970721
                164326213000000
Associated Entry ID . . . : 19970721164325210000
User ID . . . . . : USER21
Job ID . . . . . : EDITR
Application ID . . . . . : EDIFFS
Format ID . . . . . : $$STD-'++:..?'
Data . . . . . : UNB+UNOA:1+GBLLD .LLD324+GBLLD
                .LLD053:ZZ+930602:1646+40000092++PLACIN++1++1'

UNH+40000200+LIMUWM:1:2:LI'RFF+UMR+B9999LIMUWMUMR1'RFF+TR+B9999LIMUW
TR1'RFF+BQR+B9999LIMUWBQR1'
AD+BK+9999'NAD+BU+LL'NAD+CS+ 10 AEC'GIS+02:04'NAD+IV+ LINE
:LL'GIS+01:26'GIS+01:15'GIS+Y:16'
RFF+LIR+0001'RFF+CS1+COSLINE 1'RFF+CS1+COSLINE
'RFF+LIN+001'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 1 TE'
L02+1::LINE ONE OF THE INVITEDITER LINE 1 TEXT+3::LINE TWO OF THE
```

Programmer's Reference

```
NVITEDITER LINE 1 TEXT+::LINE THREE
HE INVITEDITER LINE 1 TEXT'NAD+IV+ LINE
:LL'GIS+02:26'GIS+01:15'GIS+Y:16'RFF+LIR+0002'
RFF+CS1+COSLINE 2'RFF+CS1+COSLINE
'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
NVITEDITER LINE 2 TEXT+::LINE THREE
HE 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
CFIELD(%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 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

Notices

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.

Licenses 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.

Bibliography

This bibliography lists the books in the IBM WebSphere Business Integrator and associated libraries.

IBM WebSphere Business Integrator library

The Business Integrator library consists of the following books:

- *WebSphere Business Integrator Concepts and Planning, GC34-5960*
This book introduces the Business Integrator system, providing a high-level system overview, defining the system capabilities, and describing its value to e-businesses. This book also provides the information that you need to plan the installation of Business Integrator.
- *WebSphere Business Integrator Installation Guide, GC34-5961*
This book is a guide to installing and configuring Business Integrator, It contains information about:
 - Selecting your required topology
 - Installing and configuring the base products and software components of Business Integrator on each machine in the topology
 - Installing and configuring firewalls and proxies
- *WebSphere Studio Business Integrator Extensions Installation Guide, SC34-5962*
This book is a guide to installing and configuring Solution Studio, It also contains information about setting up clients and servers, and creating projects.
- *WebSphere Business Integrator Run Time*
This book is a comprehensive guide to the Business Integrator runtime system, providing the following information:
 - Detailed conceptual information about the runtime components of Business Integrator.
 - Deployment of solutions to the runtime system
 - System administration, such as starting and stopping software components and base products, defining users, and using the Exception Console.
 - General problem determination information, including how to trace and debug, and information on obtaining help from technical support

- *WebSphere Business Integrator Messages*
This book lists the error messages that are produced by Business Integrator and provides references to the documentation for the messages of base products.
- *WebSphere Studio Business Integrator Extensions Developer's Guide*
This book describes how to create a Business Integrator solution, beginning with the solution design phase, to the solution implementation phase, and finally the solution deployment phase using a sample business problem. This book also provides procedures for assembling a Business Integrator solution in the run-time environment and a description of how to use the Solution Studio for solution design and implementation.
- *WebSphere Business Integrator DataInterchange for Windows NT User's Guide, SC34-5963*
This book is a guide to installing and using DataInterchange, in the Business Integrator environment.

You can find the latest versions of the books at the following Web site:

<http://www-4.ibm.com/software/webservers/btobintegrator/>

This site contains links to the Web sites of the underlying products of IBM WebSphere Business Integrator.

Related documentation

The *utilities* subdirectory on the Documentation CD contains documentation about utilities that can prove useful in building and running solutions. This documentation is not available on the IBM WebSphere Business Integrator Web site.

WebSphere Business Integrator also provides a number of external application programming interfaces (API). HTML documentation that is generated using the Javadoc tool is provided for these APIs. For a list of the APIs, refer to the *WebSphere Business Integrator Run Time* book.

WebSphere Partner Agreement Manager library

The Partner Agreement Manager Version 2 Release 1 library consists of:

- *Partner Agreement Manager Installation Guide*, GC34-5964
- *Partner Agreement Manager Administrator's Guide*
- *Partner Agreement Manager User's Guide*
- *Partner Agreement Manager Adapter Developer's Guide*
- *Partner Agreement Manager Script Developer's Guide*
- *Partner Agreement Manager API Guide*
- *Partner Agreement Manager Adapters for MQSeries User's Guide*
- *Partner Agreement View User's Guide*, GC34-5965
- *WebSphere Partner Agreement Manager Business Process Integration Adapter Guide*.

DataInterchange library

The DataInterchange Version 3 Release 1 library consists of:

- *DataInterchange Client User's Guide*, SB34-2010
- *DataInterchange Administrator's Guide*, SB34-2002
- *DataInterchange Installation Guide*, GB09-8070
- *DataInterchange Messages and Codes*, SB34-2000
- *DataInterchange Programmer's Reference*, SB34-2001

Other Libraries

You can find important information in the libraries of the following products:

- DB2[®] UDB
 - *IBM DB2 Universal Database Quick Beginnings Version 6.1* , S10J-8149
- MQSeries[®]
 - *MQSeries for Windows NT Quick Beginnings*, GC34-5389
 - *MQSeries System Administration*, SC33-1873
 - *MQSeries Using Java*, SC34-5456
 - *MQSeries MQSC Command Reference*, SC33-1369
 - *MQSeries Queue Manager Clusters*, SC34-5349
 - *MQSeries Integrator Introduction and Planning*, GC24-5599
 - *MQSeries Workflow Getting Started with Buildtime*, SH12-6286
 - *MQSeries Workflow Getting Started with Runtime*, SH12-6287
 - *MQSeries Adapter Kernel for Multiplatforms: Quick Beginnings*, GC34-5855
 - *MQSeries Adapter Kernel for Multiplatforms: Problem Determination Guide*, GC34-5897

- *MQSeries Adapter Builder for Windows NT: Using the Control Center, GC34-5882*
- **SecureWay[®]**
 - *SecureWay Policy Director Up and Running, SCT6-3KNA*
 - *SecureWay Policy Director Base Administration Guide*
 - *SecureWay Firewall User's Guide, CG31-8658*
- **VisualAge[®]**
 - *VisualAge Java, Enterprise Edition Getting Started*
 - *VisualAge C++ Professional for Windows NT Getting Started*
- **WebSphere[™] Application Server**
 - *Introduction to WebSphere Application Server, SC09-4430*

Index

A

artifacts 7
 associate to editor 7
 dictionary files 8
 export/import files 8
 PERFORM command files 8
 XML DTD 8

B

bizInstall.cmd 5
BodyType field 24
building a solution 9

C

channels, define 7
Client\Setup.exe 6
codes
 diagnostic 28
 exception 16
 return 15
communication
 summary 2
CONFIG32, register 6
configure.bat 6
Control record format 23

D

database, register for ODBC
 CONFIG32 6
 EDICT31E 6
 EDIEC31E 6
DataInterchange Client
 installing 5
 using 7
DataInterchange Gateway 1
DataInterchange Server, installing 6
DB2 Client Configuration Assistant 6
define
 channels 7
 data definitions 7
 maps 7
 standards 7
 trading partners 7
DeliveryCategory field 24
DestinationLogicalID field 24
diagnostic code 28
dictionary files 8
directory, verify installation 6
DTDCConvert 7

E

EDI data, translating to XML 28
EDICT31E, register 6
EDIEC31E, register 6
EpicExceptions 28
ePICHeaderField 8
Event log query 29
exception codes 16
exception service, Solution Manager 28
export/import files 8
ExternalID field 24

F

field names, control record
 DataInterchange 23
 WebSphere 23
files
 ADF 23
 bizInstall.cmd 5
 configure.bat 6
 DataInterchange Dictionary 7
 export/import 7
 PERFORM command, create 7
 setup.exe 6

G

gateway 1

I

installation
 DataInterchange Client 5
 DataInterchange Server 6
 disk space 5
 MDAC 2.5 6
 program 5
InternalID field 24
introduction 1, 5

J

Java Run Time Environment 5

M

mapping 2
maps, define 7
messages 15
MQAO message header 23
 BodyCategory 23
 BodyType 24
 DeliveryCategory 24

MQAO message header (*continued*)

- DestinationLogicalID 24
- ExternalID 24
- InternalID 24
- Message 24
- Message Type 24
- ProcessingCategory 25
- RespondToLogicID 25
- SessionID 25
- SourceLogicID 25
- TransactionID 25

MQSeries Adapter Kernel 5

O

output files 25

P

PERFORM command files 8

ProcessingCategory field 25

Q

query, event log 29

query, transaction log 30

R

register database for ODBC

- CONFIG32 6

- EDICT31E 6

- EDIEC31E 6

report sample 29

Reporting

- summary 2

requirements, installation 5

- DB2 5

- DTDs 5

- Java Run Time Environment 5

- MQSeries 5

- MQSeries Adapter Kernel 5

RespondToLogicID field 25

return codes 15

S

scenarios 26

SessionID field 25

setup

- summary 2

setup.exe 6

solution, building 9

Solution Manager exception service 28

SourceLogicID field 25

standards

- defining 7

- summary 2

T

tools registration 7

trading partners, define 7

transaction, define 7

transaction store query 30

TransactionID field 25

translation 2

types, files

- DIC 8

- DICMD 8

- EIC 8

U

utilities 36

Utilities

- DTDCConvert 7

X

XML data translate 28

XML DTD 8



Part Number: BIZAAI01
Program Number: 5724-A78



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC34-5963-00



(1P) P/N: BIZAAI01

