



IBM Software Group

IBM WebSphere® Data Interchange V3.3

Architecture of the WebSphere Data Interchange utility



@business on demand.

© 2007 IBM Corporation

This presentation will describe the WebSphere Data Interchange Utility architecture.

Agenda

- Describe the WebSphere Data Interchange utility architecture
 - ▶ Utility
 - ▶ Send translation process
 - ▶ Receive translation process
 - ▶ Data transformation process
 - ▶ CICS environment
 - ▶ AIX and Windows environment



The presentation will distinguish the differences between the Send, Receive translation process and the Data Transformation process. The Utility interface in the CICS, AIX and Windows environments will also be reviewed.

Utility

- Provides command-level access to WebSphere Data Interchange Server services
 - ▶ Outbound EDI documents
 - ▶ Inbound EDI documents
 - ▶ General data transformation or translation
 - ▶ Managing data
 - ▶ Reporting and extracting data
 - ▶ Export/Import administrative data
 - ▶ Managing mailboxes



The WebSphere Data Interchange Utility is the central point of access to WebSphere Data Interchange functions and the most commonly used interface. The Utility provides command-level access to WebSphere Data Interchange Server services. These services can be divided into the following categories:

Translating data from any EDI, XML, or data format to any other EDI, XML, or data format using a data transformation map.

Translating to EDI standard format using a send map, enveloping, and sending.

Receiving, deenveloping, and translating to data format using a receive map.

Updating, removing, and controlling record status.

Formatting and printing reports, extracting data, and printing application data.

Exporting and importing administrative data.

And closing CICS mailboxes.

Utility

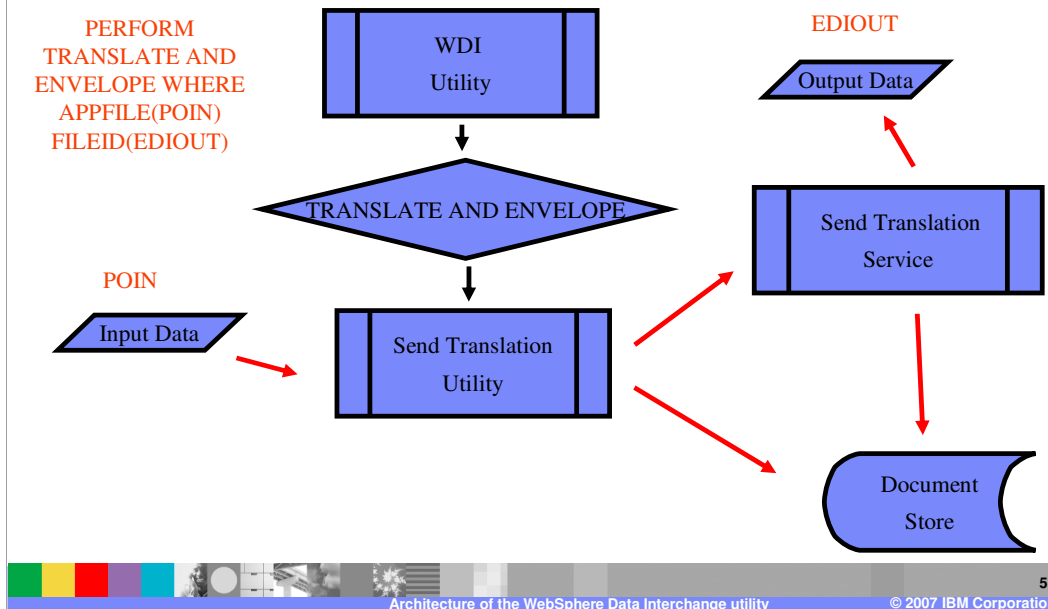
- WebSphere Data Interchange Utility command language:
 - ▶ PERFORM statements
 - defines the action the server takes
 - ▶ WHERE clauses
 - supplies selection criteria and other information that WebSphere Data Interchange requires to process your requests
 - ▶ SELECTING clauses
 - provide selection criteria



The WebSphere Data Interchange Utility command language consists of PERFORM statements, WHERE clauses, and SELECTING clauses.

A PERFORM statement defines the action WebSphere Data Interchange takes. For example: PERFORM followed by TRANSLATE AND ENVELOPE. A WHERE clause supplies selection criteria and other information that WebSphere Data Interchange requires to process your requests. A WHERE clause consists of the word WHERE, followed by one or more keywords and their associated values. The keywords are documents in the WebSphere Data Interchange V3.3 Utility Commands and File Formats Reference. Some statements use a SELECTING clause. Like WHERE clauses, SELECTING clauses provide additional selection criteria.

Send translation message flow



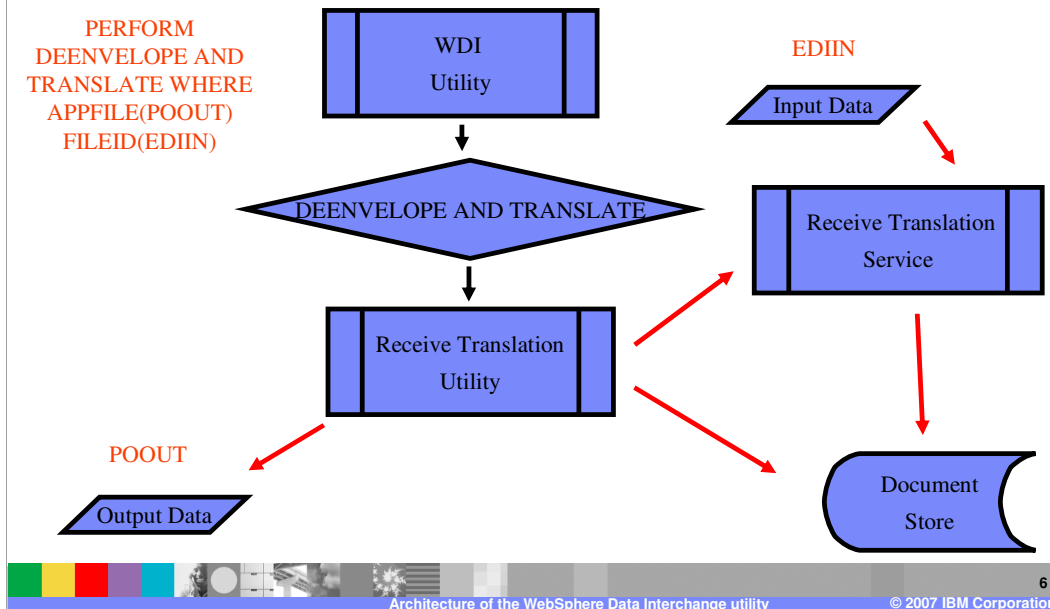
The Send translation process is an outbound process. With outbound processing the source or input data is always application format and the target or output data is always EDI format. Send translation processing requires send maps and includes the following functions:

Translating application data into an EDI standard format and placing it in the Document Store.

Enveloping standard transactions or messages so they are ready to be sent.
Sending enveloped data to trading partners.

Commands are supplied to do each of these three steps independently or in combination.

Receive translation message flow



The Receive translation process is an inbound process. With inbound processing the source or input data is always EDI format and the target or output data is always application data format. Receive translation processing requires receive maps and includes the following functions:

Receiving data from trading partners.

Deenveloping interchanges and placing the EDI standard transactions or messages into the Document Store.

Translating EDI standard transactions or messages into application formats.

Commands are supplied to do each of these three steps independently or in combination.

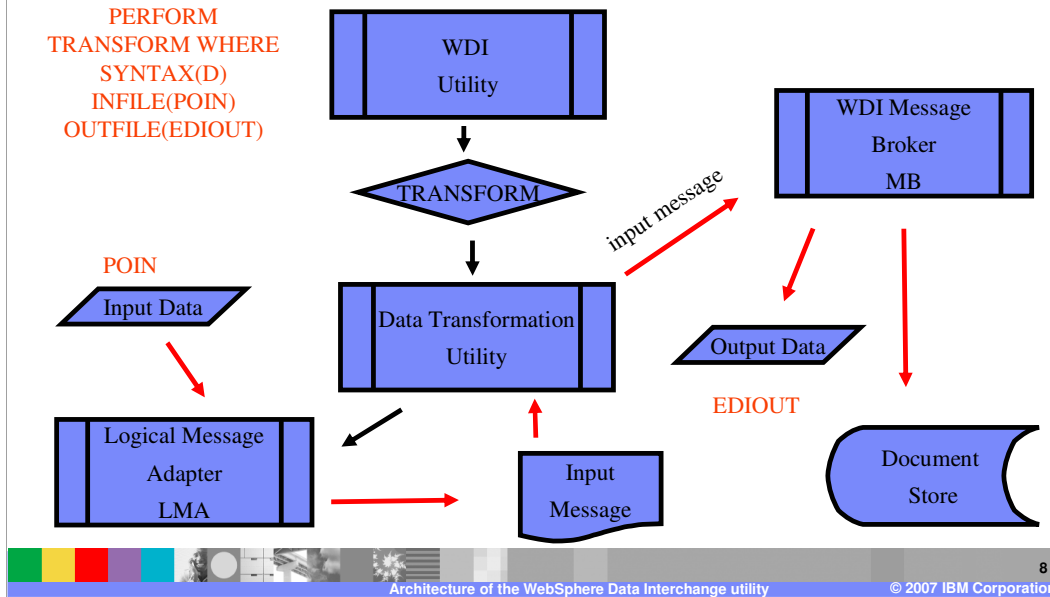
Data transformation processing

- Any-to-any translation
- Supports EDI, XML, application data
- SYNTAX keyword identifies the source message type
- Data transformation map selection identify the target message type



Data transformation is a WebSphere Data Interchange feature that you can use to translate or transform data from any supported source document type to any supported target document type. Supported document types include data formats, EDI standards, and XML data. Data transformation processing requires data transformation (DT) maps. Keywords are supplied to do each of the steps for outbound and inbound processing independently or in combination. The SYNTAX keyword identifies the source or input message type.

Data transformation message flow



The Data Transformation Utility calls the Logical Message Adapter to read the input data and parse out 1 logical input message based on the SYNTAX keyword. Valid values for the SYNTAX keyword are: E for EDI, X for XML, and D for application data. The Utility does setup for the WebSphere Data Interchange Message Broker including PERFORM keyword options and document store options. Each logical message is passed to the MB for processing.

The Message Broker initializes the message flow for processing. It creates and initializes the source Logical Message and the source document properties in the form of an Abstract Message. During translation a target Abstract Message is created for the output. The Message Broker causes the target Abstract Message to be serialized and written to the output file. The Message Broker also controls the document store updates.

Managing data

- Rebuilding interchanges
- Updating status of transactions in document store
- Removing transaction from document store
- Updating and removing statistics



Data management includes the following functions:

Rebuilding interchanges from transactions in the Document Store.

Updating status of transactions in the Document Store.

Removing transactions from the Document Store.

Updating management reporting statistics.

And Removing management reporting statistics.

The WebSphere Data Interchange Utility provides commands to do each of these functions. For example, the PERFORM RECONSTRUCT command can be used to rebuild an interchange.

Reporting and extracting data

- Generate formatted reports containing data from the document store
- Format application data into conventional business documents
- Extract data from the document store for further processing outside WebSphere Data Interchange
- Extract data from the management reporting statistics tables
 - ▶ to be processed outside WebSphere Data Interchange



You can use the Utility PRINT commands with selection criteria to extract information from the Document Store and write formatted reports to ddname RPTFILE. To obtain the same type of information without formatting, use the **ENVELOPE** and **TRANSACTION DATA EXTRACT** commands. You can print the information using your system facilities.

WebSphere Data Interchange provides two mechanisms for producing reports:

Management Reporting data extracts.

Document Store data extracts.

Both mechanisms collect, update, and extract WebSphere Data Interchange trading partner data and transaction information. Both are invoked using the WebSphere Data Interchange Utility PERFORM commands to extract the data, and both require user-written programs to sort, format, and print data.

The Management Reporting and Document Store Data Extracts differ in the type of information they provide. Management Reporting Data Extracts pull information from the statistics tables, while Document Store Data Extracts pull information from the Document Store.

Export and import

- Metadata definitions
 - EDI, XML, application data format
- ▶ Maps
- ▶ Trading partners



The **EXPORT** and **IMPORT** Utility commands exports or imports administrative data. Objects that are used during translation or transformation such as metadata definitions, maps, and trading partner setup can be exported and imported or exchanged between WebSphere Data Interchange systems. You tell WebSphere Data Interchange specifically which setup information you want to exchange using a control file.

Managing mailboxes

- CICS only
- PERFORM CLOSE MAILBOX



WebSphere Data Interchange does not wait for send requests to complete when using GXS Expedite/CICS and Information Exchange. When any type of access to Information Exchange is accomplished (send, receive, continuous receive, network status update), the mailbox opens and remains open until you close it. Closing your mailbox is only necessary when the same mailbox will be used in the z/OS environment. To close the mailbox, you must use this command to end the session. This command is only available in the CICS environment and is not required for general processing.

CICS environment

- Utility functions operate the same
- Continuous receive
- CICS interfaces provided
 - ▶ Can use different CICS storage mechanisms
 - ▶ Invocation options
 - CICS LINK
 - CICS START
 - CICS ATI
 - EDIW



Most functions of WebSphere Data Interchange operate similarly regardless of the environment in which they are executed.

The continuous receive Facility is a WebSphere Data Interchange service that works in conjunction with GXS Expedite/CICS and Information Exchange. It also can be used with WMQ trigger queues. Using the continuous receive Facility, you can receive and develope EDI standard data, translate the EDI standard data to application format, automatically initiate transaction-level response applications that process application data placed in TS queues, and automatically receive and process network acknowledgments. Because Information Exchange passes the data to the host system immediately after the data enters the mailbox, WebSphere Data Interchange gains control shortly after data is sent from the trading partner's system.

In CICS, when you develop an application program to run the WebSphere Data Interchange Utility, you can write the application in any programming language supported by CICS. The Utility provides standard CICS interfaces and can be instructed to use different CICS storage mechanisms.

The following methods can be used to invoke the WebSphere Data Interchange Utility:

IBM Confidential

AIX and Windows environment

- Command Line
 - ▶ Example invocation: `ediservr < sample.cmd > results.txt`
 - ▶ Utility PERFORM commands located in a file
- WebSphere Data Interchange Adapter
 - ▶ Triggering from WebSphere MQ
 - ▶ Executes Utility command `PERFORM RECEIVE AND PROCESS`
 - ▶ WebSphere Data Interchange Advanced Adapter
 - Same function as WebSphere Data Interchange Adapter
 - Multiple requests in parallel



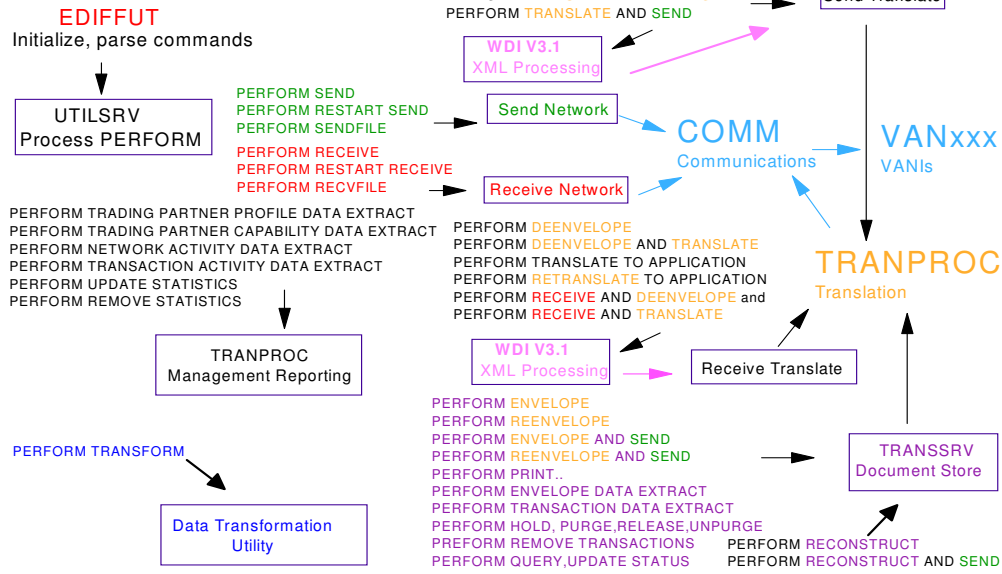
Command line invocation executes the `ediserver` program. WebSphere Data Interchange AIX and Windows Server always reads commands from STDIN and writes the results to STDOUT: these are treated as STREAMs. When invoked from the command line, the command line processor automatically opens STDIN and STDOUT, piping them wherever the user requests. You typically prepare a file of PERFORM commands for input and redirect the input from that file. You would probably redirect the STDOUT to a file.

The WebSphere Data Interchange adapter program is installed as part of WebSphere Data Interchange for Multiplatform Version 3.2. The configuration scripts provided set up the necessary queues and definition objects. The adapter uses WebSphere MQ Triggering to know when messages need processing. When a message is put to an application queue, a trigger message is created. The WebSphere MQ trigger monitor receives the message and executes the adapter. The adapter then passes the information needed to process the application message to the WebSphere Data Interchange server/translator. Application messages are committed, rolled back, or moved to a failure queue depending on the return codes from the WebSphere Data Interchange Server.

The WebSphere Data Interchange advanced adapter is designed to provide an WebSphere MQ interface to WebSphere Data Interchange which is capable of handling multiple requests in parallel. The original WebSphere Data Interchange Adapter program was designed to handle requests in a serial fashion. This meant that when a trigger message was generated for a queue a single translator instance was assigned to that queue to process all

WebSphere Data Interchange utility

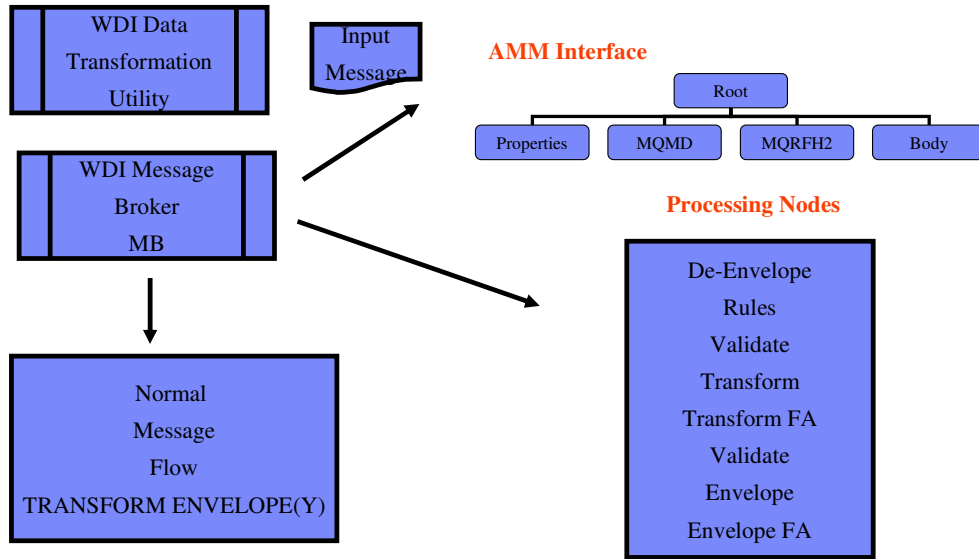
Start WDI Utility (EDIFFUT)



This diagram shows the WebSphere Data Interchange utility commands and the translation services executed. The major components are the WDI Utility (UTILSRV), Send Translate, Receive Translate (TRANPROC), Data Transformation Utility, Communications (COMM), and the document store (TRANSRRV).

WebSphere Data Interchange utility

Data Transformation Processing : Message Flow



With the data transformation process, the data transformation utility is used. A Normal Message Flow is the message flow for a translate and envelope process as opposed to delayed enveloping processing. It defines the processing NODES for the message. All processing nodes create and update information for the Document Store and optional record processing. Each node also has a source abstract message and propagate a target abstract message for the next processing node.

WebSphere Data Interchange utility

■ References

- ▶ WebSphere Data Interchange V3.3 Utility Commands and File Formats Reference
- ▶ WebSphere Data Interchange V3.3 Programmer's Reference Guide



More information on the WebSphere Data Interchange Utility Commands and keywords may be found in the WebSphere Data Interchange Version 3.3 Utility Commands and File Formats Reference. Information for using CICS interfaces and AIX and Windows adapters can be found in the WebSphere Data Interchange Version 3.3 Programmer's Reference Guide.

Trademarks, copyrights, and disclaimers

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

IBM	CICS	IMS	WebSphere MQ	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
ef (logo)/business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

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

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

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2006. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

