



IBM Software Group

IBM WebSphere® Data Interchange V3.3

Network Interface



@business on demand.

© 2007 IBM Corporation

This presentation will describe the WebSphere Data Interchange network interface.

Agenda

- Communications
- Generalized Networks
- Point to Point Networks
- Parameters Communication Routine
- CICS
- WebSphere MQ
- Reference



The presentation will describe the network interface for communications.

Communications

- Communication API
 - ▶ sequential flat file
 - ▶ CICS TS queue
 - ▶ Queue standard data
 - FXXZccc (SNB, CCB, FCB, CMCB, TPPDB, DATBLK)
 - ▶ Send Files
 - FXXZccc (SNB, CCB, FCB, CMCB, TPPDB)



All communication requests by WebSphere Data Interchange or by applications using WebSphere Data Interchange are made through the Communications API. The functions provided in the Communication API enable an application program to send or receive transaction data, files, and messages to and from trading partners. The transaction or file data that is to be transmitted is not passed directly through an API parameter. Rather, it is provided indirectly as the name of a sequential flat file that contains the data to be sent or must be used to collect the data that is being received. In CICS, the data to be sent or received is restricted to a TS queue. WebSphere Data Interchange includes support for multiple networks and allows you to write programs to add support for any network that is not supported.

Communications

- Generalized network (VAN)
 - ▶ Network program provided
 - ▶ Command interface
 - ▶ Multiple interchanges and destinations in a file
 - ▶ Hold data in a mailbox
- Point to Point network
 - ▶ Generalized communications package
 - ▶ Direct connection to trading partner
 - ▶ Data for a single trading partner



A generalized network consists of a network program provided by the network provider, such as IEBASE from information exchange and DSXMIT2 from the General Electric Information Services Company. There is an interface to the network program where commands are placed in an input file by the requestor and responses are placed in the output file by the network program. WebSphere Data Interchange has no control over this interface since it is defined by the network provider. Generalized networks have the ability to process multiple interchanges in a single file and use information from the interchanges to determine the destination. There is a connection made to the network itself rather than any particular trading partner, holding data in a mailbox until requested by the trading partner.

A point-to-point network consists of a generalized communications package rather than a specific program designed for a particular network. There is a direct connection to the trading partner. Output files contain data for a single trading partner.

Section

Generalized Networks

Generalized Networks

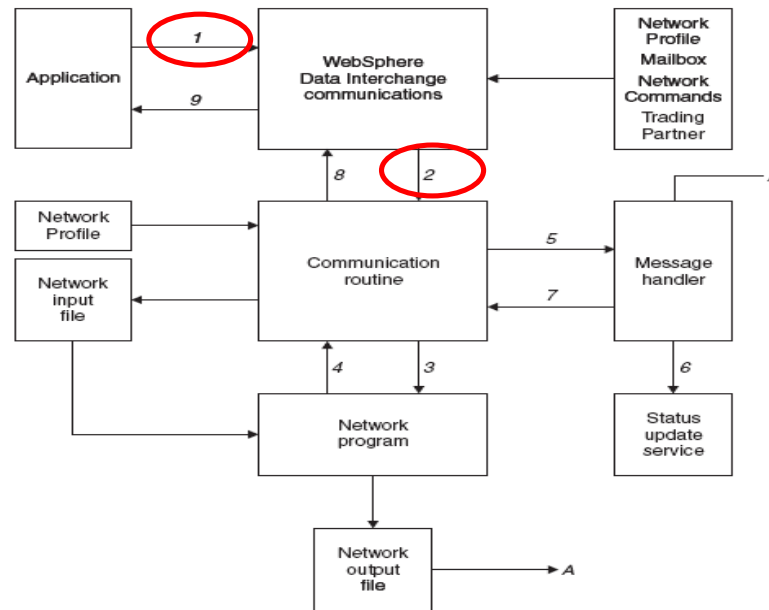
- Reads profiles
- Invokes communication routine
 - ▶ Network Profile
- Passes information
 - ▶ Application
 - ▶ Network program
- Process API request
- Build interface to network program
- Process results from network program



The communications module reads all the necessary profiles and invokes the communications routine defined to handle the network. The name of the communications routine is provided in the Communication routine field of the network profile (NETPROF). The communications routine passes information between the application and the network program and must:

- Process the API request
- Build the appropriate interface to the network program
- Interpret the results from the network program

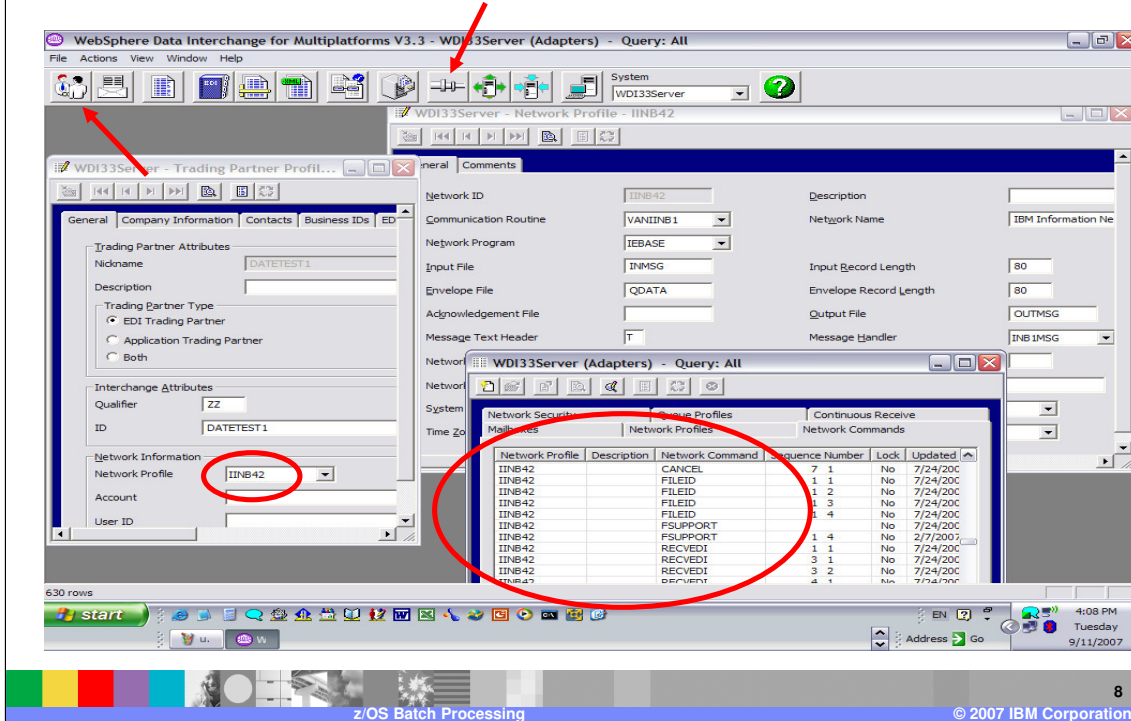
Generalized Networks



1. An application program requests WebSphere Data Interchange communication services through the API. The request retrieves:
 - The mailbox (requestor) profile member (REQPROF) identified by the REQID field of the communications control block.
 - The trading partner profile member (TPPROF) identified by the TPNICKNM field of the communications control block. This retrieval occurs only if the trading partner data block passed to communications contains blanks.
 - The network profile member (NETPROF) identified by the requestor member or by the NETID field in the CCB.
 - The functions supported by the network identified in the FSUPPORT member in the network commands profile (NETOP).

2. Using the program identified in the Communication rtn field of the network profile member, communications invokes this program as a user exit routine through the language interface routine (FXXZccc). The communications routine name is a logical name and must be one of the following:
 - VANINFC
 - VANIINB1
 - VANEXPV4
 - PTTOPT
 - GEISVAN

Generalized Networks



WebSphere Data Interchange Client 3.3 Trading partner profile. The Network and Network Commands profiles are located in the Adapters functional area.

Generalized Networks

WebSphere Data Interchange for Multiplatforms V3.3 - WDI3 Server (Adapters) - Query: All

WDI33Server - Trading Partner Profil...

General | Comments

Trading Partner Attributes

Nickname: DATETEST1

Description:

Trading Partner Type

EDI Trading Partner

Application Trading Partner

Both

Interchange Attributes

Qualifier: ZZ

ID: DATETEST1

Network Information

Network Profile: **IINB42**

Account:

User ID:

Network ID: IINB42

Description:

Communication Routine: VANINB1

Network Name: IBM Information Ne

Network Program: IEBASE

Input File: INMSG

Input Record Length: 80

Envelope File: QDATA

Envelope Record Length: 80

Ad acknowledgement File:

Output File: OUTMSG

Message Text Header: T

Message Handler: INBIMSG

Network Sequence:

Network Parameters

System Type:

Time Zone:

WDI33Server (Adapters) - Query: All

Mailbox Name	Description	Network Profile	Account	User ID
ADF_IN	Mailbox ...	ADF	IBM1	DDSTST1REQ
ADF_OUT	Mailbox ...	ADF	IBM1	DDSTST1REQ
ATEST	Receive	IINB42		
CYCL2WDI	Receive	CYCLONE		
DIEDIIN	IN_XML			
DIEDICOUT	IN_XML			
DIXMLIN	IN_XML			
DIXMLOUT	IN_XML			
EDI_IN	Mailbox ...	EDI	IBM1	DDSTST1REQ
EDI_OUT	Mailbox ...	EDI	IBM1	DDSTST1REQ
TRPT2WDI	Receive	TRNET		

20 rows

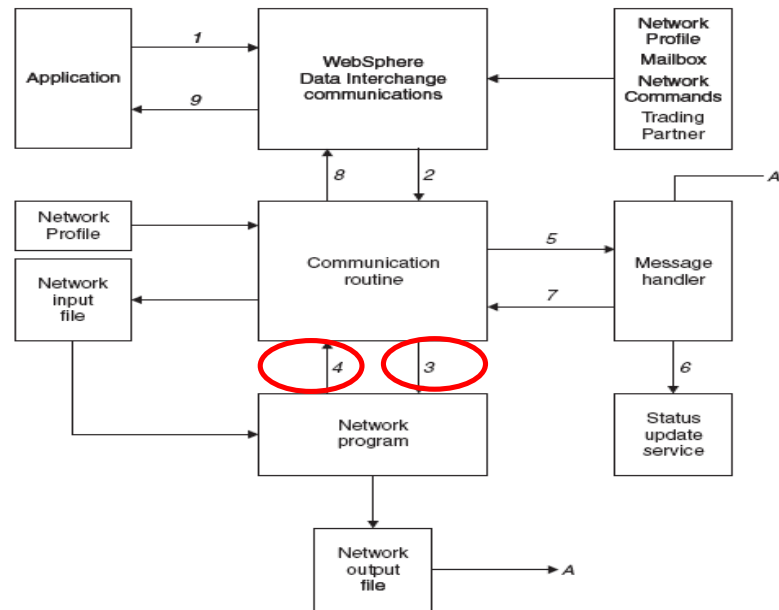
start

z/OS Batch Processing

© 2007 IBM Corporation

WebSphere Data Interchange Client 3.3 Trading partner profile. The Network and Mailboxes or Requestor profiles are located in the Adapters functional area.

Generalized Networks



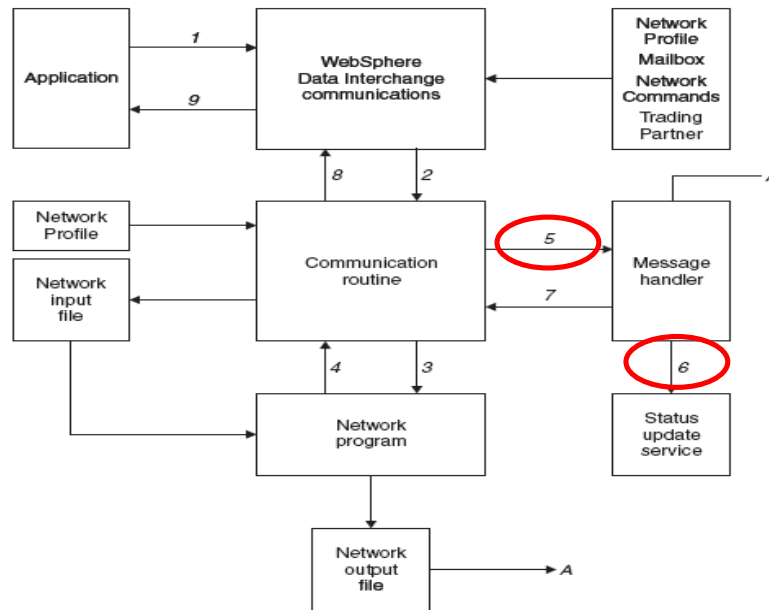
10

z/OS Batch Processing

© 2007 IBM Corporation

3. The communication routines has control. Communications routines are driven by the function requested (send, receive, or cancel) and by the network command specified in the NETOP field of the communications control block. The 8-byte network ID and the 8-byte network command are concatenated and used as a partial key to retrieve all matching entries from the Network Commands profile (NETOP). Network Commands members contain instructions for building commands, which are literal values combined with data from the other control blocks and profile members. The commands are written to a file as specified by the Network input file field from the network profile.
4. This is a synchronous operation where the communication routine waits until the network program is finished. When that program returns control, it is assumed that the operation is completed or that the network program is responsible for making sure the operation completes.

Generalized Networks



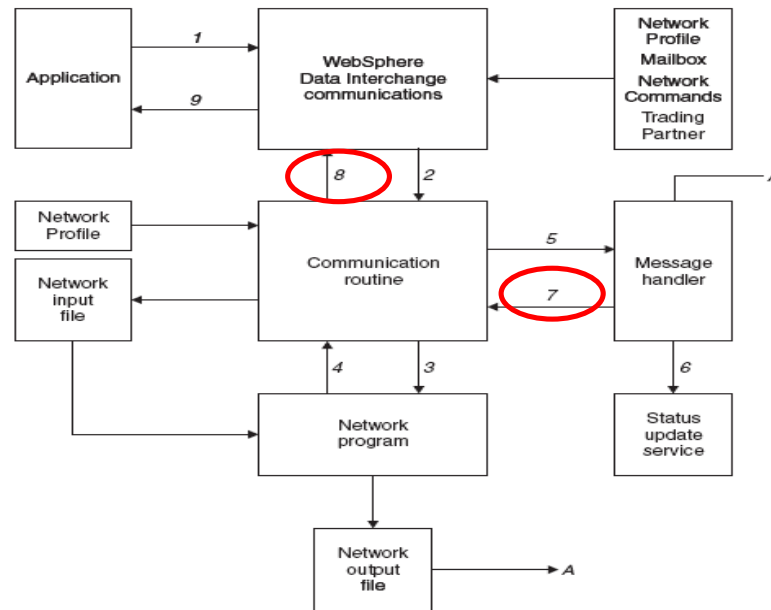
11

z/OS Batch Processing

© 2007 IBM Corporation

5. If sending transaction data is requested, the communication routine updates the status in the Document Store of each interchange in the file based on the return code from the network program.
6. If the network output file contains status information about previously sent data, the message handler should use the status update service to update the status information in the Document Store.

Generalized Networks



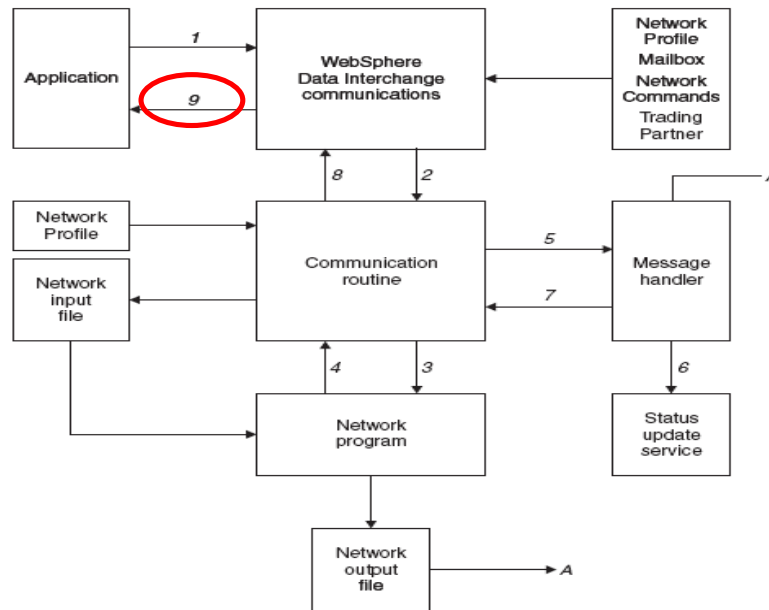
12

z/OS Batch Processing

© 2007 IBM Corporation

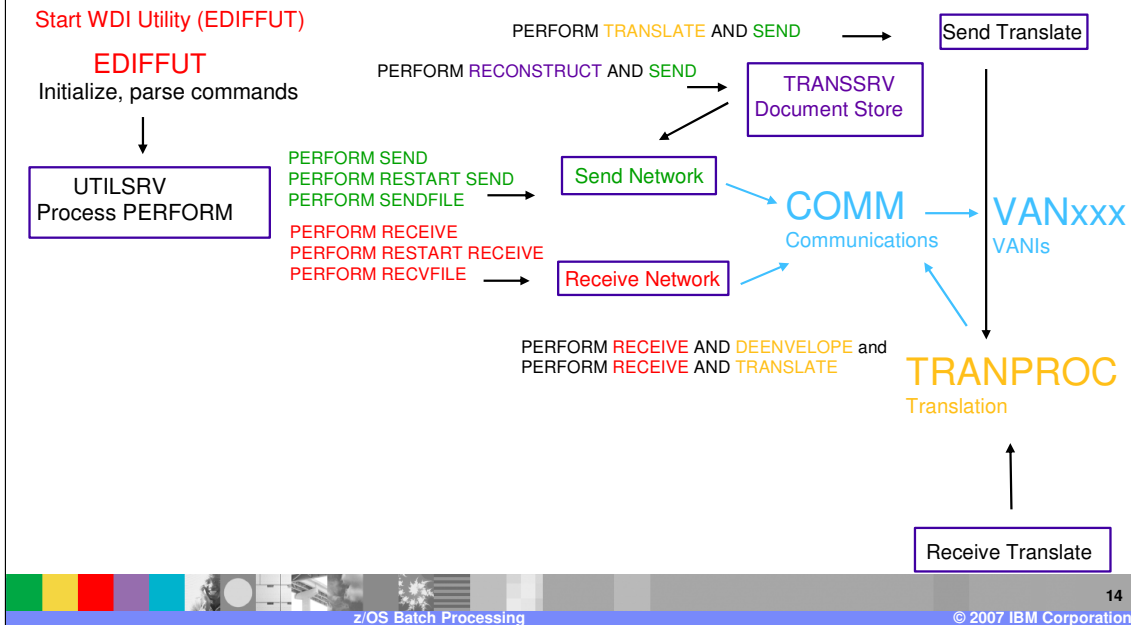
7. If the network output file indicates that data has been received, the message handler should set the FILERCVD field in the communications control block to **Y** and put the account number and user ID of the trading partner sending the data into the trading partner data block.
8. The communication routine uses a reverse lookup on the trading partner profile members to locate the entry belonging to the returned account number and user ID. The trading partner nickname is returned to the application program so that it knows data was received and from what trading partner. If any errors were encountered, the communication routine writes an entry to the event log indicating the type of error that occurred.

Generalized Networks



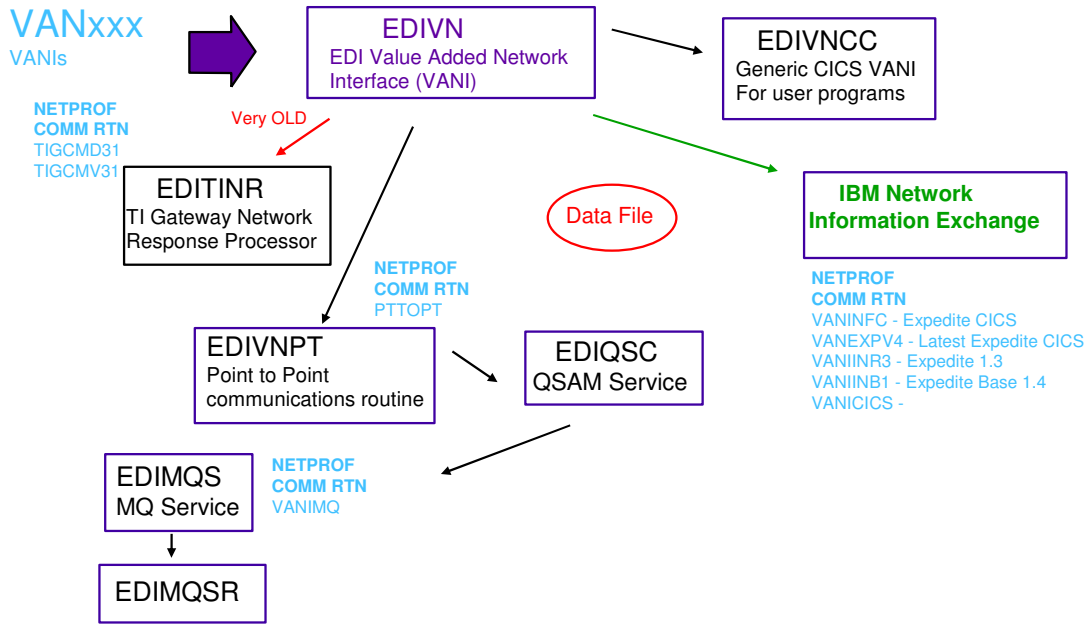
9. The application regains control with the results of the operation indicated by the ZCCBRC and ZCCBERC fields of the CCB.

Generalized Networks

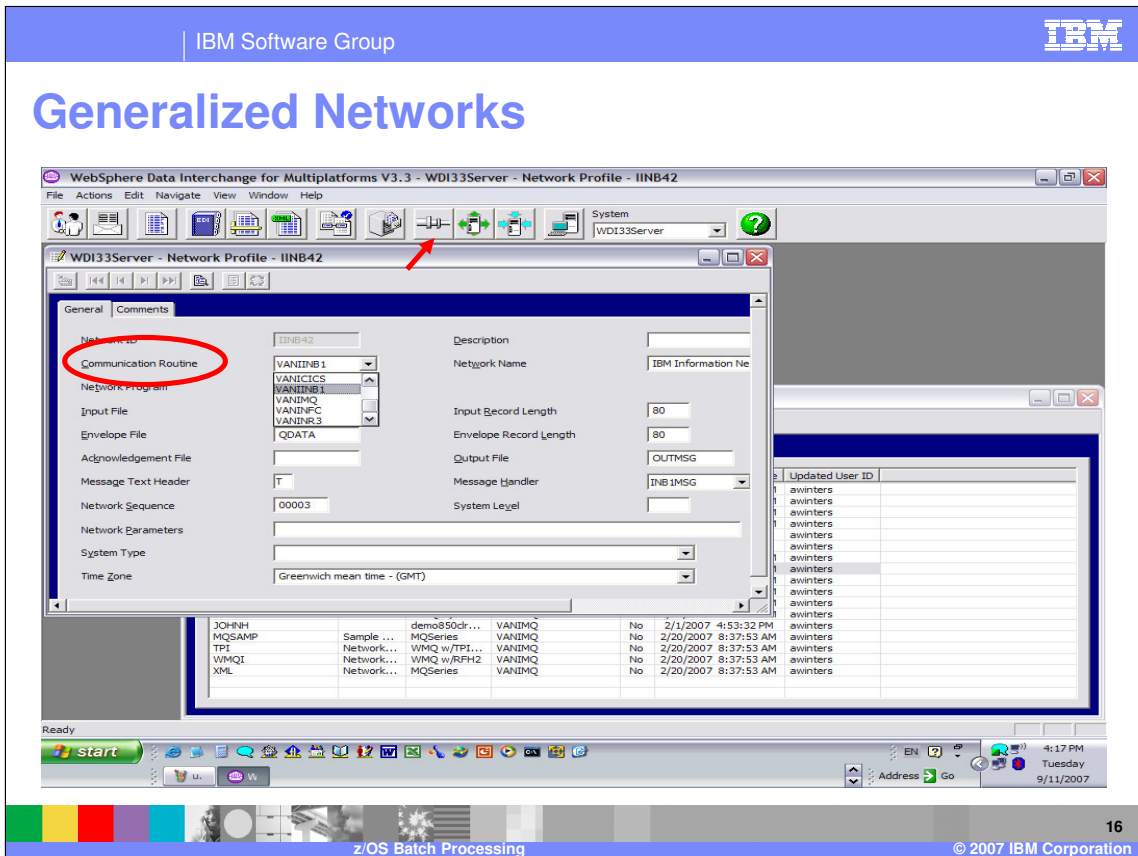


When a communication request is identified via a WebSphere Data Interchange Utility PERFORM command. The Utility uses the communication API to process the request.

Generalized Networks



These are the communication routines provided by WebSphere Data Interchange.

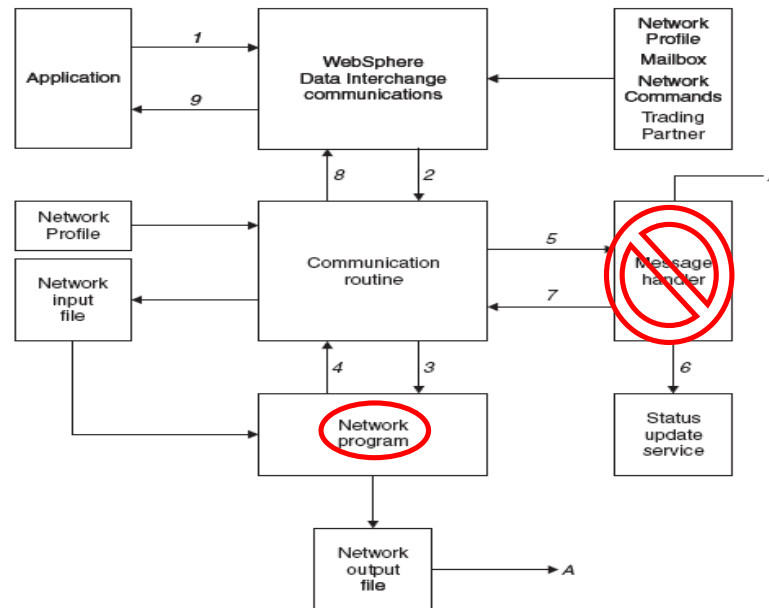


The communications routine to be used is a selectable field in the Client Network Profile. The message handler program is identified in the Message handler field of the network profile. The message handler is architecturally the same as a user exit and, therefore, has the same language and linkage edit considerations as user exits. The value in the Message handler field is the logical name for the exit. The physical load module name and the implementation language for the message handler are defined in the user program information profile (User Exits). An User Exit entry is not needed for the network profile entries that are distributed by WebSphere Data Interchange as these programs are defined in internal WebSphere Data Interchange tables.

Section

Point to Point Networks

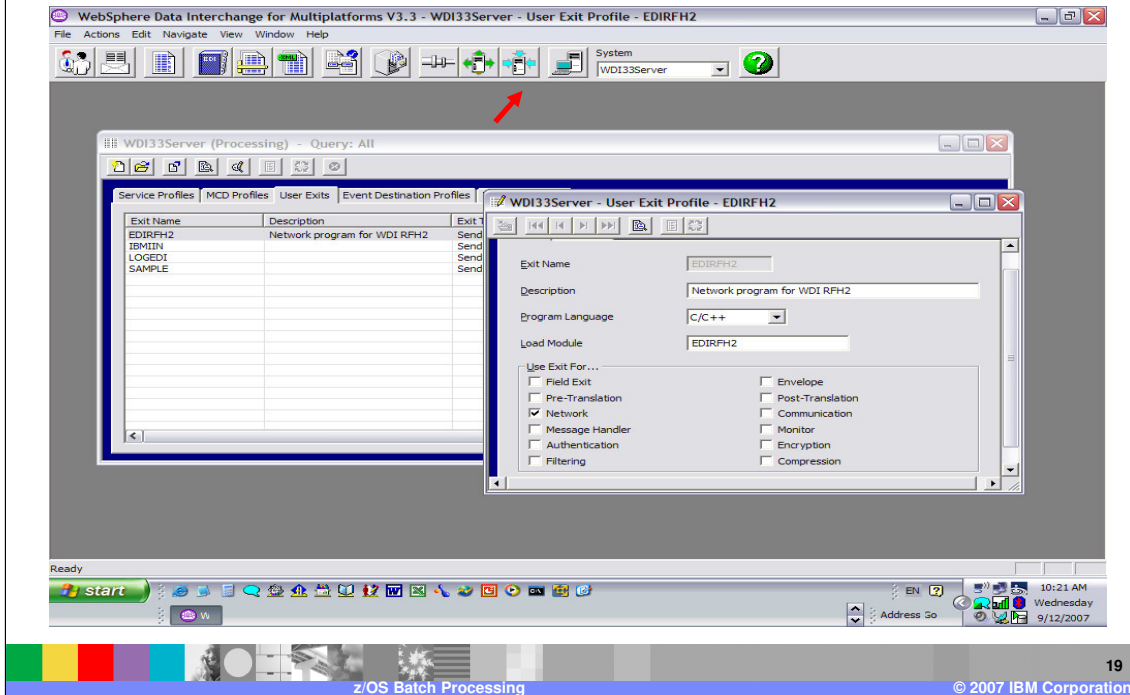
Point to Point Networks



The flow for a point-to-point network is similar to the flow for the generalized network. with the following exceptions:

1. The network program is called as a WebSphere Data Interchange exit rather than using an z/OS ATTACH. You must define the program in the User Exits (ADAMCTL) profile and write the program in a language supported by WebSphere Data Interchange. The parameters passed to the point-to-point network program are the same as those passed to the communications routine.
2. The network program receives the same parameters as the communications routine (a programming interface rather than a file interface).
3. There is no message handler for a point-to-point network. The network program is responsible for processing the requests and for processing the results expected by the API.

Point to Point Networks



User exit profiles are located in the Client Processing functional area. A User Exit profile contains information defining a user program or exit routine to WebSphere Data Interchange. User Exit profiles can be viewed and changed using the editor.

There are two formats of the User Exit Editor - one supports Data Transformation Exits and the other supports Send or Receive Exits. **Data Transformation Exits** are currently used in Data Transformation Maps, Validation Maps and Functional Acknowledgement Maps. The other type of exit, known as **Send or Receive Exits**, are used in Send Maps, Receive Maps, network communications, network security, pre-translation, and post-translation.

Section

Parameters Communications Routine

Parameters

- Service Name Block (SNB)
- Common Control Block (CCB)
- Function Control Block (FCB)
- Communication Interface Control (CMCB)
- Trading Partner Profile (TPPDB)
- Network Profile Block (NPDB)
- Mailbox (Requestor) Profile Block (REQDB)



Communications invokes the communications routine to process each request from an application that requires network activity, such as requests to send or receive transactions. Your application program supplies the required values before calling communication. The data supplied by the application is combined with information from the profiles, and the parameters listed below are passed to the communications routine. For point-to-point networks (Communication routine has a value of **PTTOPT**), these same parameters are sent to the network program.

IBM Software Group IBM

Network Commands

Network Profile	Description	Network Command	Sequence Number	Lock	Updated Date and Time	Updated User ID
INB41		CANCEL	6 7	No	9/10/2007 7:47:23 PM	RPOPE
INB41		CANCEL	7 1	No	9/10/2007 7:47:23 PM	RPOPE
INB41		FILEID	1 1	No	9/10/2007 7:47:23 PM	RPOPE
INB41		FILEID	1 2	No	9/10/2007 7:47:23 PM	RPOPE
INB41		FILEID	1 3	No	9/10/2007 7:47:23 PM	RPOPE
INB41		FILEID	1 4	No	9/10/2007 7:47:23 PM	RPOPE
INB41		FSUPPORT		No	9/10/2007 7:47:23 PM	RPOPE
INB41		FSUPPORT	1 4	No	9/10/2007 7:47:23 PM	RPOPE
INB41		RECVEDI	1 1	No	9/10/2007 7:47:23 PM	RPOPE
INB41		RECVEDI	3 1	No	9/10/2007 7:47:23 PM	RPOPE
INB41		RECVEDI	3 2	No	9/10/2007 7:47:23 PM	RPOPE
INB41		RECVEDI	4 1	No	9/10/2007 7:47:23 PM	RPOPE
INB41		RECVEDI	5 1	No	9/10/2007 7:47:23 PM	RPOPE
INB41		RECVEDI	5 2	No	9/10/2007 7:47:23 PM	RPOPE
INB41		RECVEDI	5 3	No	9/10/2007 7:47:23 PM	RPOPE
INB41		RECVEDI	5 4	No	9/10/2007 7:47:23 PM	RPOPE
INB41		RECVEDI	5 5	No	9/10/2007 7:47:23 PM	RPOPE
INB41		RECVEDI	5 6	No	9/10/2007 7:47:23 PM	RPOPE
INB41		RECVEDI	5 7	No	9/10/2007 7:47:23 PM	RPOPE

1206 rows

10:31 AM
Wednesday
9/12/2007

z/OS Batch Processing © 2007 IBM Corporation

A communications routine (such as VANIINB1) uses a network commands profile to build command records that it places in a network input file (such as INMSG). The information needed to build the command records is in the control blocks passed to the communication routine. Entries in the network commands profile tell the communication routine what data from the control blocks should be used in building the commands for the network program.

Network Commands

The screenshot displays the WebSphere Data Interchange for Multiplatforms V3.3 interface. The main window is titled "WDI33FVT - Network Command Profile - IINB41 FSUPPORT". It features a menu bar (File, Actions, Edit, Navigate, View, Window, Help) and a toolbar. The interface is divided into several panes:

- Mailboxes:** A list of network profiles with columns for Network Profile, Description, and Network Command. The list includes entries like IINB41 CANCEL, IINB41 FILEID, IINB41 FSUPPORT, and IINB41 RECVEDI.
- Network Profiles:** A pane showing the selected profile details.
- Network Command Profile - IINB41 FSUPPORT:** A configuration window with the following fields:
 - Network Profile: IINB41
 - Network Command: FSUPPORT
 - Sequence: (empty)
 - Description: (empty)
 - Data Source: (empty)
 - Block Name: (empty)
 - Block Position: (empty)
 - Literal Value: YYYYYNNYYYYNN
 - Move Data to: Line, Position, Length (all empty)

The system tray at the bottom shows the Start button, system icons, and the date/time: 10:36 AM Wednesday 9/12/2007. The footer of the slide includes "z/OS Batch Processing" and "© 2007 IBM Corporation".

Communications determines the support provided by a network before calling the communications routine for the network. It does this by reading the network commands profile using the key network id and network command FSUPPORT. The literal in the command field value (**LITERAL VALUE**) of the member indicates which functions are supported. The first byte corresponds to FQUEUED; the second byte corresponds to FMSGs, and so on. Values entered in the FSUPPORT member must be in uppercase.

Section

CICS

CICS

- VANICICS
- LINK interface
- Provides Interchange queuing function
- Management Reporting



A special communications routine for the CICS environment (logical name VANICICS) uses a LINK interface to pass control to the network program and the message handler. If you do not plan to use GXS Expedite/CICS to communicate with trading partners, this special communications routine meet your needs. VANICICS also provides the interchange queuing function. Whenever an interchange is generated through an ENVELOPE function, the translator indirectly invokes the associated communication routine to write the interchange to the appropriate TS queue. Management reporting functions are also provided. Whenever an EDI data receive occurs, VANICICS invokes management reporting to update the appropriate statistics.

CICS

- User defined
 - ▶ Network program
 - ▶ Message handler
 - ▶ CICS COMMAREA
 - Pointers to control blocks
 - Point to Point Communications Routine
 - Results



You can provide your own network program and message handler with a CICS interface. VANICICS passes control information to the network program and message handler by way of the CICS COMMAREA. The network program you supply receives control information via the CICS COMMAREA.

The COMMAREA is made up of a set of 4-byte addresses pointing to control blocks. These control blocks contain all the critical information your program requires to process the request. While most of the communication routines in WebSphere Data Interchange use the network commands profile member, VANICICS does not. The interface to your network program is similar to the z/OS point-to-point communication routine. Control blocks with critical information are passed to the program instead of the command being built in the COMMAREA. This method of passing control blocks gives your program greater flexibility

Your program is expected to overwrite the COMMAREA with the resulting information, and VANCICS will log errors if your program determines the execution was unsuccessful.

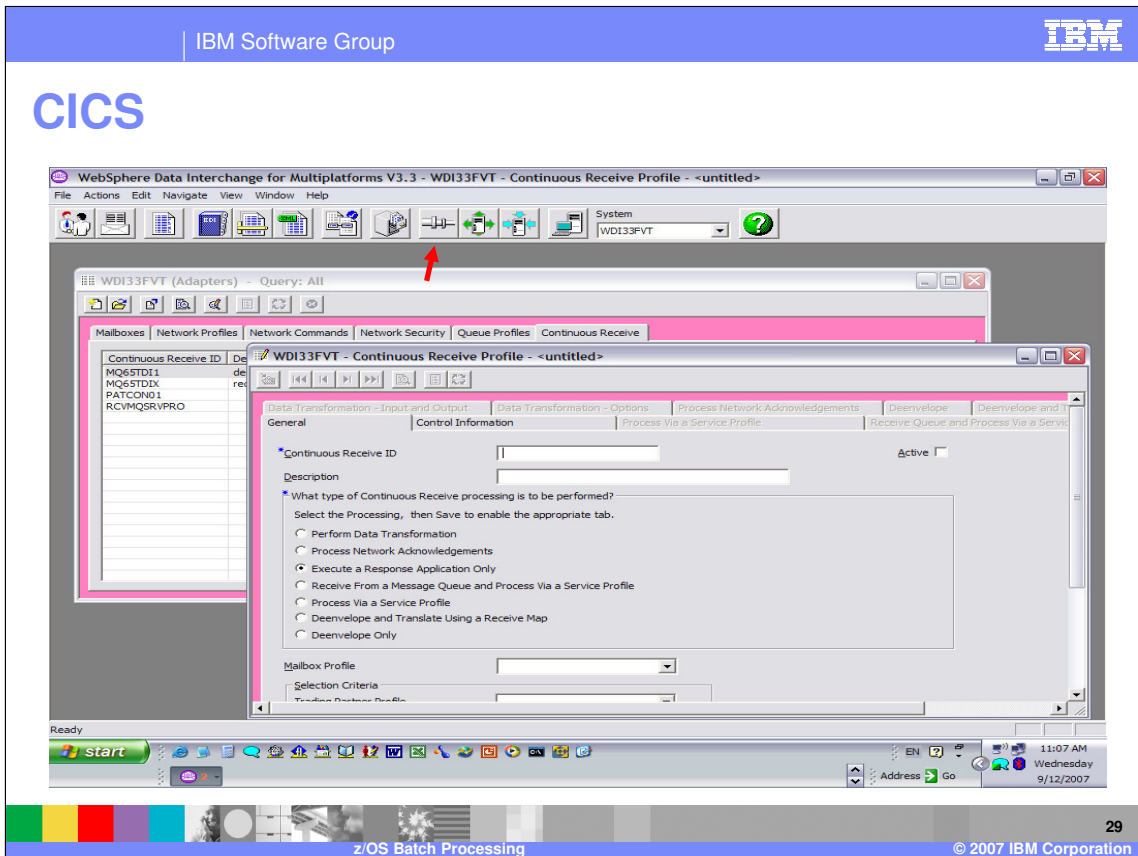
CICS

▪ Continuous Receive

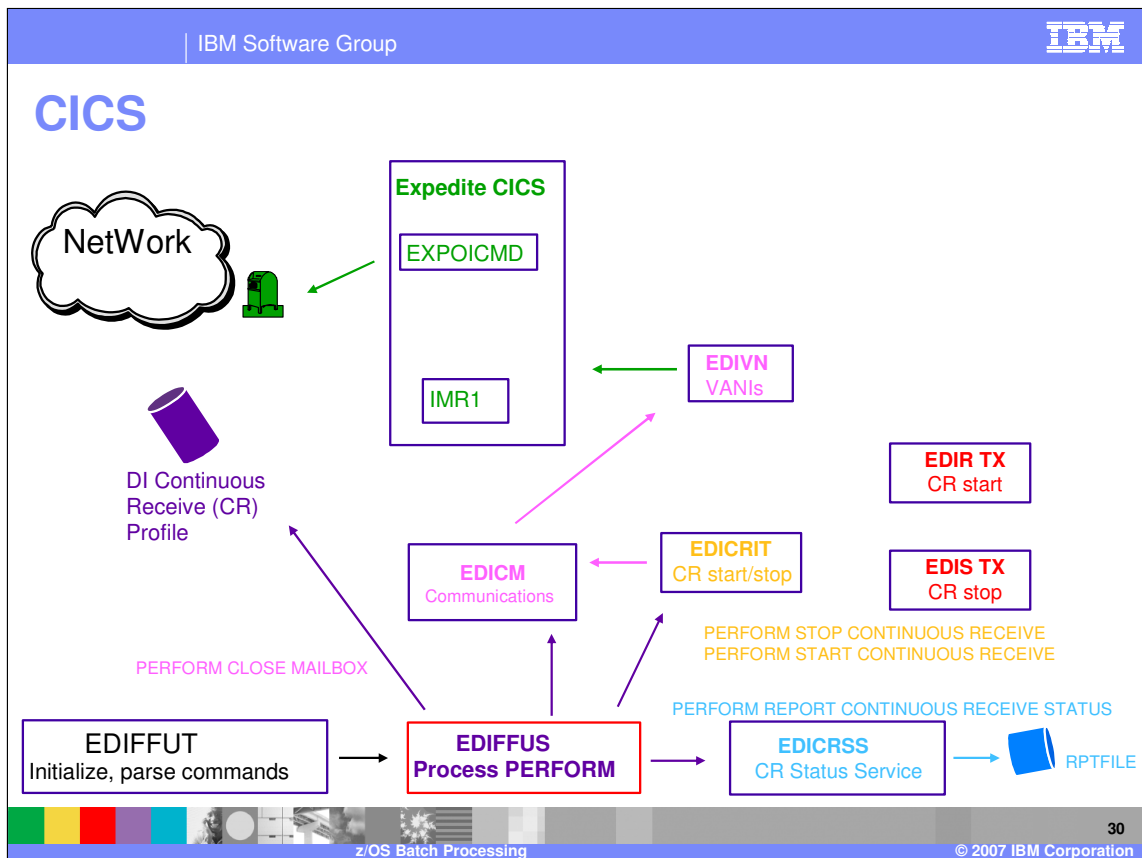
- GXS Expedite/CICS and Information Exchange
- WebSphere MQ
- Continuous Receive Profile
- Event-driven translation



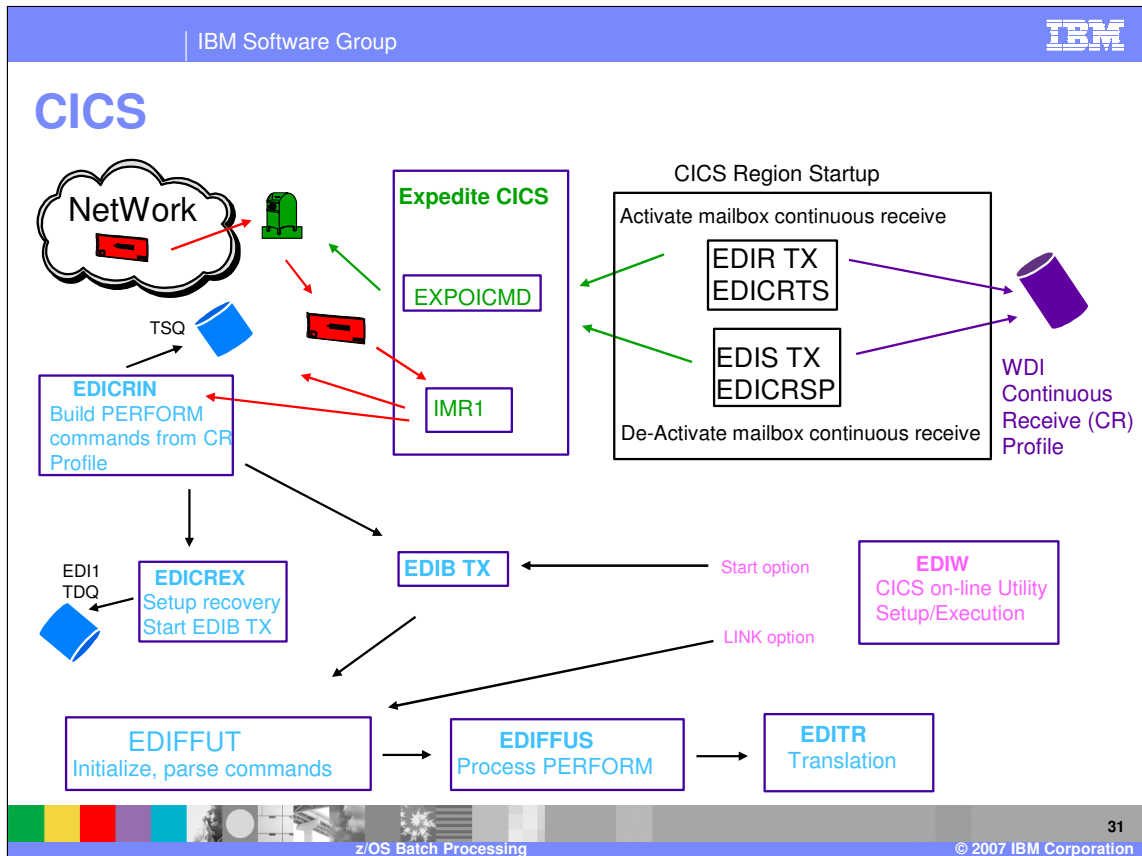
The continuous receive facility was designed to work with GXS Expedite/CICS and Information Exchange, but it can be used with user-written applications. The intent of this interface is to house WebSphere Data Interchange processing information in the continuous receive profile, away from the receiving application's logic. The user-supplied receive application does not need to be concerned with the WebSphere Data Interchange processing that should be performed. This information is contained in the continuous receive profile member. It provides options and parameters used when you initiate *event-driven translation* using WebSphere Data Interchange/CICS and either GXS Expedite/CICS, WebSphere MQ, or your own user written communication routines. You can automate the process of receiving and processing documents sent to your mailbox or a message queue by trading partners.



The Continuous Receive profile is located in the Client Adapters functional area. The Continuous Receive profile allows you to specify selection criteria when using GXS Expedite/CICS. Selection criteria is not used when using WebSphere MQ with continuous receive. If using another network with continuous receive, use of selection criteria is dependent on the other network. By defining selection criteria, you can set up multiple continuous receive sessions to receive and process documents from specific trading partners and specific document types. You can also set up a Continuous Receive profile that will cause only network acknowledgements to be received and processed.



The continuous receive facility of WebSphere Data Interchange provides two CICS terminal transactions: EDIR to start a continuous receive session, and EDIS to stop a continuous receive session. These two CICS transactions can be used at any time and are complemented by the WebSphere Data Interchange Utility commands START CONTINUOUS RECEIVE and STOP CONTINUOUS RECEIVE. Also, the WebSphere Data Interchange Utility provides the REPORT CONTINUOUS RECEIVE STATUS command for reporting the statuses of your continuous receives.



Only continuous receives associated with GXS Expedite/CICS are started and stopped with CICS transactions EDIR and EDIS, or with PLT programs EDICRTS and EDICRSP. Continuous receives associated with user-written receive applications are not started or stopped through WebSphere Data Interchange. The user-written receive application passes the name of the continuous receive profile member to WebSphere Data Interchange when a continuous receive occurs. Since this is all the control information WebSphere Data Interchange requires to perform the appropriate processing, starting and stopping is not necessary. Any request to start or stop a continuous receive associated with a user-written receive application is ignored by WebSphere Data Interchange.

Section

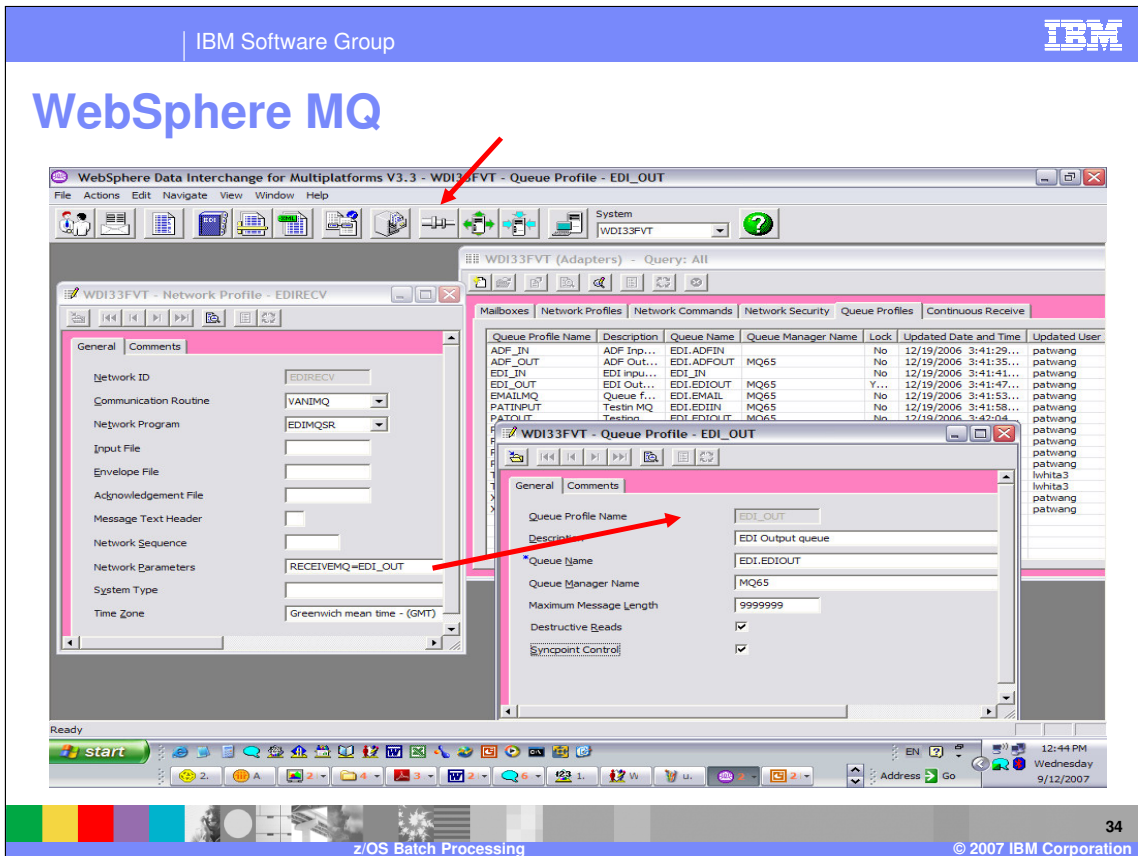
WebSphere MQ

WebSphere MQ

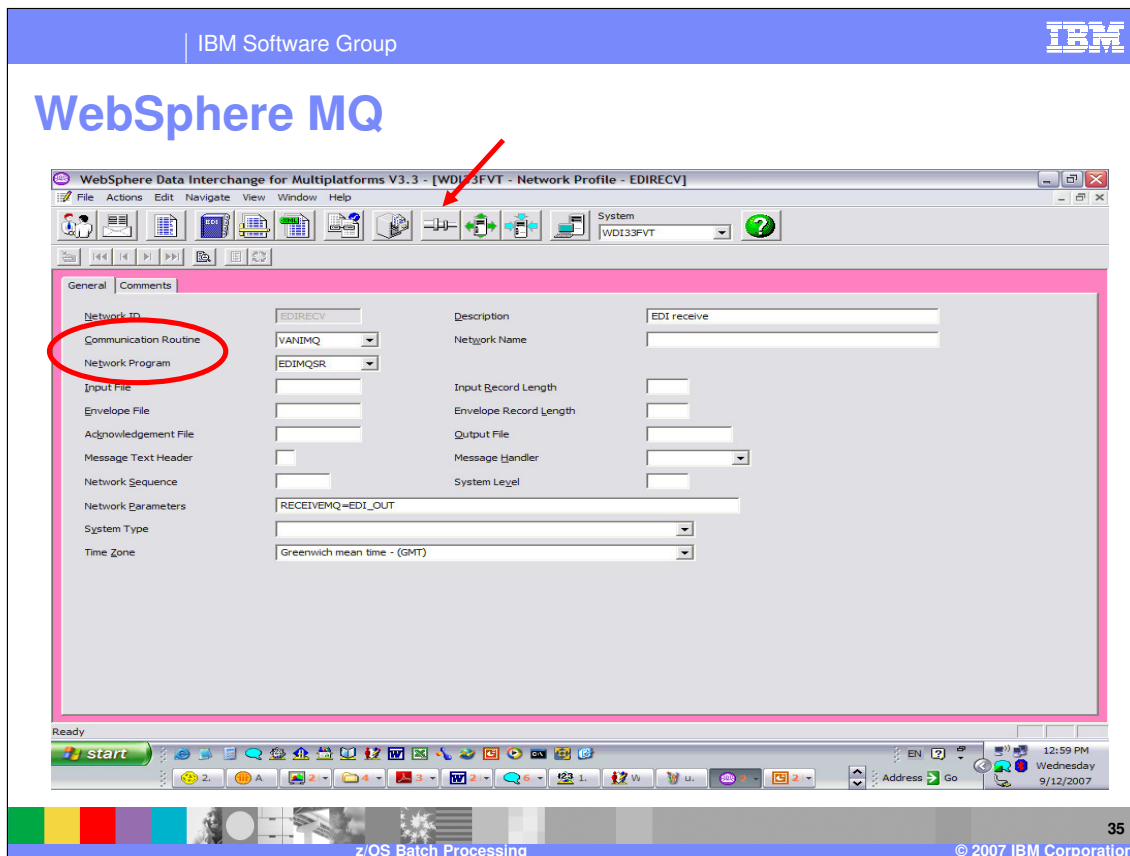
- Trading Partner Profile
- Network profile
- Queue profile
 - ▶ Associates a logical name with a physical WebSphere MQ message queue

You can exchange data with your trading partners using WebSphere MQ queues in network profiles. The Trading Partner profile identifies the Network profile to be used. WebSphere Data Interchange can use an WebSphere MQ message queue for most of its input and output files. To use a WebSphere MQ message queue with WebSphere Data Interchange, you must first define the queue to WebSphere MQ and then identify it to WebSphere Data Interchange. Identifying a message queue to WebSphere Data Interchange is accomplished through a **Queue profile**.

WebSphere Data Interchange prohibits the use of WebSphere MQ queues as the target of the envelope process or as input to the develope process. WebSphere MQ queues can be used as part of a logical network where enveloped data is sent to and received from WebSphere MQ queues. WebSphere Data Interchange provides a sample network profile member named *MQSAMP* to assist you with the setup of this local network,



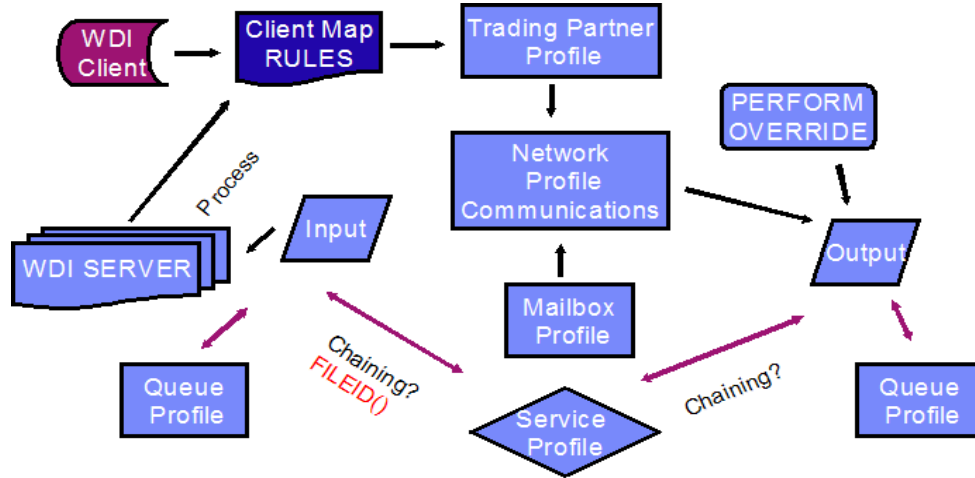
The Network and Queue profiles are located in the Adapters functional area. Network parameters identify the logical names of the queues used to send and receive messages to a trading partner when WebSphere MQ queues are used to communicate with the trading partner. WebSphere Data Interchange associates each external trading partner with a network. If WebSphere MQ queues are used to communicate with the trading partner, then the trading partners "network" consists of two WebSphere MQ queues; a queue used to send to the trading partner (**SENDMQ**) and a queue that the trading partner uses to send messages to you (**RECEIVEMQ**). This field must match the WebSphere Data Interchange WebSphere MQ queue profile member names.



In the Network profile, the name of the communication routine that builds network commands and invokes the network's send and receive program to process the commands should be VANIMQ.

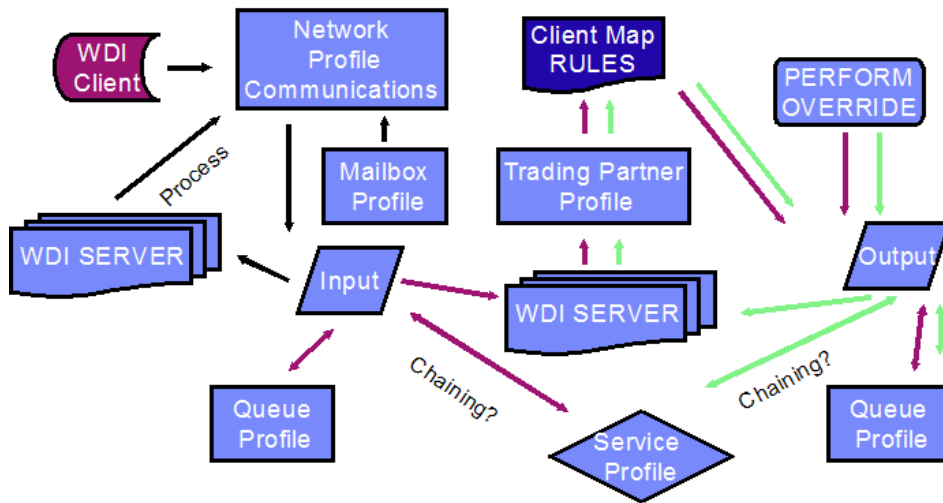
The Network program can be **EDIMQSR**, **EDIRFH2** or **EDICYCL**. This is the physical name of the send and receive program. This program is invoked by the communication routine to process requests. If messages are to be MQPUT to a queue with only an MQMD header, then specify the **EDIMQSR** network program. If messages are to be MQPUT to a queue with both the MQMD header and an RFH2 header, then specify **EDIRFH2**. WebSphere Data Interchange will fill out the <mcd> folder and add values like the sender and receiver ID's to the <usr> folder of the RFH2 for each message MQPUT to the queue. **EDIRFH2** should also be used for queues going to the iSoft EDI over the Internet (EDI-INT) Gateway product. If the queue goes to a product using the JMS API, such as the IBM Trading Partner Interchange product, Cyclone Interchange Server, or the IBM Web Services Gateway, then specify **EDICYCL**. This will cause WebSphere Data Interchange to format the RFH2 correctly for these products.

WebSphere MQ



This diagram shows the Client profile relationships for an outbound process.

WebSphere MQ



This diagram shows the Client profile relationships for an inbound process.

References

- WebSphere Data Interchange V3.3 Programmer's Reference Guide

More information 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
e/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 NON-INFRINGEMENT. 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.