



IBM Software Group

IBM WebSphere® Data Interchange V3.3

CICS Continuous Receive



@business on demand.

© 2007 IBM Corporation

This is the WDI version 3.3 CICS Continuous Receive presentation.

Agenda

- Provide an overview of Continuous Receive
- Explain what's new since Base 3.2
- Review Response Programs
- Review Expedite/CICS Continuous Receive
- Review MQ Continuous Receive
- Summary



This is more of a review of Continuous Receive and related items than anything else. However, there are some new things about Continuous Receive since base version 3.2 that will be covered. We will also talk a little about response programs and Expedite and MQ Continuous Receives.

Continuous Receive Overview

- Continuous Receive allows an automated way for WDI to receive and process messages in a CICS environment
- Continuous Receives can occur three ways
 - ▶ Expedite/CICS and Information Exchange
 - ▶ Non-Expedite/CICS
 - Link to program EDICRIN with a COMMAREA that includes the name of the Continuous Receive profile
 - ▶ WebSphere MQ



For those unfamiliar with CICS Continuous Receive, it basically is an automated way for WDI to receive and process messages. No human intervention is required. No batch jobs to submit. Continuous Receive can occur three ways: via Expedite/CICS and Information Exchange, via user-written applications that link to program EDICRIN, and via WebSphere MQ.

Continuous Receive Overview

- Automatically receives and processes incoming messages
 - ▶ Responses can be automated
 - Example: A 270 (eligibility inquiry) is received. A 271 (eligibility information) can be created, enveloped, and automatically sent back.
- Automatically receives network status information
 - ▶ Expedite/CICS and Information Exchange
 - ▶ Updates WDI's Transaction Store statuses
 - From "Sent to network" to "Delivered by network"



An example of Continuous Receive might be an incoming 270 (health care eligibility/benefit inquiry) EDI transaction is received and automatically translated. A user-written response program recognizes the eligibility request, accesses company databases, formats a reply, and then invokes WDI to translate, envelope, and send the reply (the 271 transaction) back. Also, when a transaction has been sent, its Transaction Store status is set to "Sent to network". Continuous Receives can be set up to process network acknowledgments from Expedite/CICS and IE that will change the Store status to "Delivered by Network" when IE delivers the message to the trading partner's mailbox.

Continuous Receive Overview - continued

- Base WDI version 3.2 supports four types of Continuous Receive processing
 - ▶ Deenvelope
 - PERFORM DEENVELOPE WHERE ...
 - ▶ Deenvelope and translate
 - PERFORM DEENVELOPE AND TRANSLATE WHERE ...
 - ▶ Execute response program only
 - ▶ Process network acknowledgements
 - PERFORM PROCESS NETWORK ACKS WHERE ...



In Base 3.2 four types of events can occur with Continuous Receive. Deenvelope only, deenvelope and translate to application, process network acknowledgments, and execute a response program only. For those familiar with WDI's PERFORM commands, a Continuous Receive equates to an automatic receive followed by execution of an optional PERFORM command. The keywords on the PERFORM command would be populated by values set in the Continuous Receive profile.

Continuous Receive Overview - continued

- Continuous Receive profile
 1. Select the mailbox to monitor
 2. Specify selection criteria for the mailbox
 - For example: only select messages from a certain trading partner
 3. Specify Utility control information (FFUS block)
 - For example: the name of the print file
 4. Select the type of PERFORM command to be executed when a message is received from the mailbox
 5. Select keywords and values to go on the PERFORM command
 6. Name a response program that is to be executed after the PERFORM command completes



Essentially these are the six things that make up a Continuous Receive profile. You select the mailbox to monitor and the selection criteria for that mailbox. You specify the Utility control information. For example: the name of the print file. You select the PERFORM command to be executed, and the keywords that go on the PERFORM command. And finally, you specify the name of the response program to be executed after the PERFORM command completes.

New Since Base 3.2

- Support for three new types of Continuous Receive processing
 - ▶ DT translations
 - PERFORM TRANSFORM WHERE ...
 - ▶ Process via a Service profile
 - PERFORM PROCESS WHERE ...
 - ▶ Receive from an MQ and process via a Service profile
 - PERFORM RECEIVE AND PROCESS WHERE ...
- Keywords for each PERFORM command
- XML print file, ADF print file, Tracking file



New since Base 3.2, we added three new PERFORM commands that can be executed during Continuous Receive. PERFORM TRANSFORM and the two PERFORM PROCESS commands. We added these command choices to the Continuous Receive profile, as well as support for all associated keywords. We also added support for XML print file, ADF print file, and Tracking file names and types. &UNIQUE may be used for the names. In which case each execution of WDI would generate unique TSQ names.

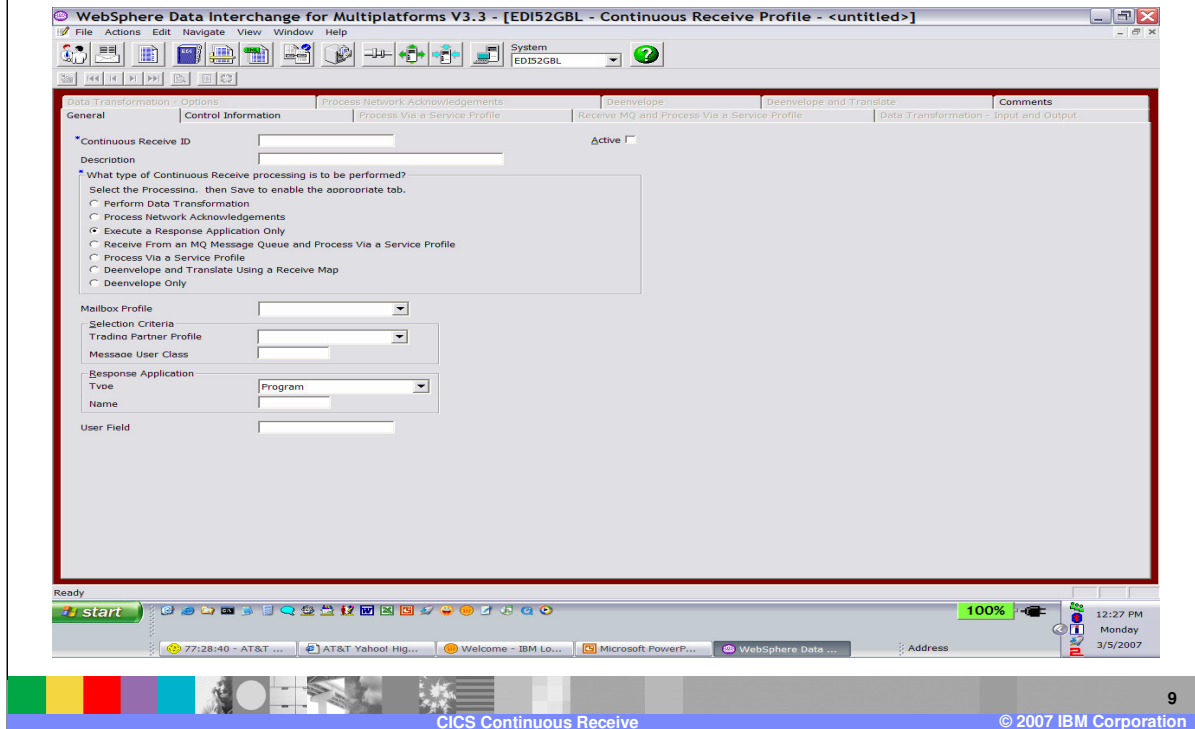
New Since Base 3.2

- The Continuous Receive profile dialog in the WDI Client has been updated to support the new PERFORM commands and their keywords
 - ▶ Within the Client Continuous Receive profile there are now separate tabs for each type of processing
 - In other words, for each type of PERFORM command
 - ▶ When a PERFORM command is selected from the general dialog, the appropriate tab becomes active
 - ▶ Within each tab are fields that correspond to that PERFORM command's keywords



The look of the Continuous Receive profile has changed. There are many more tabs now. There is a separate tab for each type of PERFORM command. Within those tabs are fields that correspond with that PERFORM command's keywords. On the main dialog you select the type of PERFORM command to execute. Then the associated tab becomes active. You can then go to that tab to specify the PERFORM command keyword values.

Client Continuous Receive Profile



This screen shot may be hard to see. It is the general tab of the Continuous Receive profile. Here you select the type of PERFORM command to be executed when a receive occurs. Doing so will activate the associated tab, where you can enter values associated with the PERFORM command's keywords.

Let's Review some Related Items

- Response Programs
- Expedite/CICS Continuous Receive
- MQ Continuous Receive



That's it for the new stuff. For those unfamiliar with CICS Continuous Receive, we thought a short review of some related items might be helpful. The next several slides review response programs, Expedite/CICS Continuous Receive, and MQ Continuous Receive.

Response Programs

- User-written CICS programs - like user exits
- These programs are passed control at certain points in the translation process
- The Utility Control Information block (the FFUS block) is the interface passed to response programs
- If the WDI Utility is invoked asynchronously (EXEC CICS START TRANSID('EDIB') or Continuous Receive is used, response programs are an essential part of the complete processing picture
- Useful when you need to know the outcome of WDI processing



CICS response programs are user-written programs. The Utility Control Information block (FFUS block) is the interface. For example, a response program might get invoked after WDI completes a translation. In which case, the translation return codes would be in the FFUS block. The response program could then take action depending on the return codes. Where WDI/CICS is concerned, response programs are often an essential part of the processing model. The FFUS block and response programs are described in the CICS chapter of the Programmer's Reference.

Response Programs - continued

- May be specified as programs or CICS transaction IDs
- Type = PG
 - ▶ WDI will EXEC CICS LINK PROGRAM (your-program) COMMAREA(FFUS) LENGTH(300)
 - ▶ These programs should not handle abends because WDI needs to free ENQs and other resources
- Type = TX
 - ▶ WDI will EXEC CICS START TRANSID(your-transaction) FROM(FFUS) LENGTH(300)
 - ▶ FFUS block obtained by doing an EXEC CICS RETRIEVE



Response programs may be specified as being programs or CICS transaction IDs. WDI will EXEC CICS LINK to programs, and EXEC CICS START transaction IDs. Programs would be synchronous with WDI processing and transactions would be asynchronous.

Response Programs - continued

- Three types of response programs
 - ▶ Utility level response programs
 - ▶ EDI transaction level response programs
 - PERFORM TRANSLATE TO APPLICATION
 - PERFORM DEENVELOPE AND TRANSLATE
 - PERFORM RECEIVE AND TRANSLATE
 - ▶ Continuous Receive response programs



As described in the Programmer's Reference CICS chapter, there are three types of response programs. There are Utility level, transaction level, and Continuous Receive response programs. Transaction level response programs pertain to the translate to application PERFORM commands seen here. Each of these types of response programs will be further explained.

Response Programs - continued

- Utility level response programs
 - ▶ Specified in the FFUS block when the WDI Utility is invoked with a PERFORM command
 - ▶ Passed control after the PERFORM command completes
- EDI transaction level response programs
 - ▶ Specified in the Document Destination field in the Receive Usage or Data Format
 - ▶ Passed control after WDI translates each EDI transaction



Utility level response programs are specified in the Utility Control Information block (the FFUS block) when the WDI Utility is invoked. The response program would get executed when the PERFORM command completes. Transaction level response programs are specified in the receive usage or data format. These response programs get executed for each EDI transaction. The name of a unique TSQ that holds the translated application data is passed to the response program in the FFUS block, along with the return codes from the translation.

Response Programs - continued

- Continuous Receive response programs
 - ▶ Specified in the Continuous Receive profile
 - ▶ Passed control after the PERFORM command completes (or instead of any PERFORM command)
 - ▶ Because Continuous Receives are real-time, background events, these programs are essential in order to know the outcome of translations, etc.



Continuous Receive response programs are specified in the Continuous Receive profile. These response programs are executed after the PERFORM completes (or instead of a PERFORM command). Because Continuous Receives are real-time, background events, these programs are essential in order to know the outcome of translations, etc.

Response Programs - continued

- Why a response program might be used:
 - ▶ Check translation return codes
 - ▶ If necessary, invoke exception handling
 - ▶ Subsequently process input and/or output files
 - ▶ Commit or rollback unit-of-work
 - When WDI is instructed not to issue SYNCPOINTs and the response program is linked to



As we have seen, response programs are essential in some cases in order to maintain integrity of WDI/CICS processing. As we saw in the 270/271 example, response programs can be very powerful as a means of triggering subsequent processes. Trading partners can actually ping pong back and forth sending and receiving transactions using Continuous Receive response programs. We actually showed a live demo of this at the DI User's Conference in San Francisco in 1997.

Expedite/CICS Continuous Receive

- IE to Expedite/CICS to WDI
- Start and stop Continuous Receives
 - ▶ EDIR (CICS transaction ID to start CR)
 - ▶ EDIS (CICS transaction ID to stop CR)
 - ▶ PERFORM START CONTINUOUS RECEIVE
 - ▶ PERFORM STOP CONTINUOUS RECEIVE
- Continuous Receive statuses
 - ▶ PERFORM REPORT CONTINUOUS RECEIVE STATUS



WDI Continuous Receive was originally designed to work with Expedite/CICS and Information Exchange. Continuous Receive is actually an Information Exchange feature that WDI takes advantage of via Expedite/CICS. The idea is that you tell Expedite that you want a certain IE mailbox monitored. When mail is delivered to that mailbox, Expedite/CICS automatically receives it and passes it on to an application. That application is WDI, in this case. The flow is always IE to Expedite/CICS to WDI. You use WDI to start and stop these types of receives. There is a report you can generate that tells the Continuous Receive profile name, the associated mailbox (Account / User ID), and whether the Continuous Receive is started or not.

Expedite/CICS Continuous Receive - continued

- Continuous Receive Flow
 1. Message delivered to IE mailbox
 2. IE automatically triggers Expedite/CICS transaction ISC2, which receives the message from IE
 3. ISC2 then starts transaction IMR1, which links to WDI (program EDICRIN)
 4. EDICRIN starts the WDI Utility transaction EDIB
 5. The Utility does what is instructed in the Continuous Receive profile

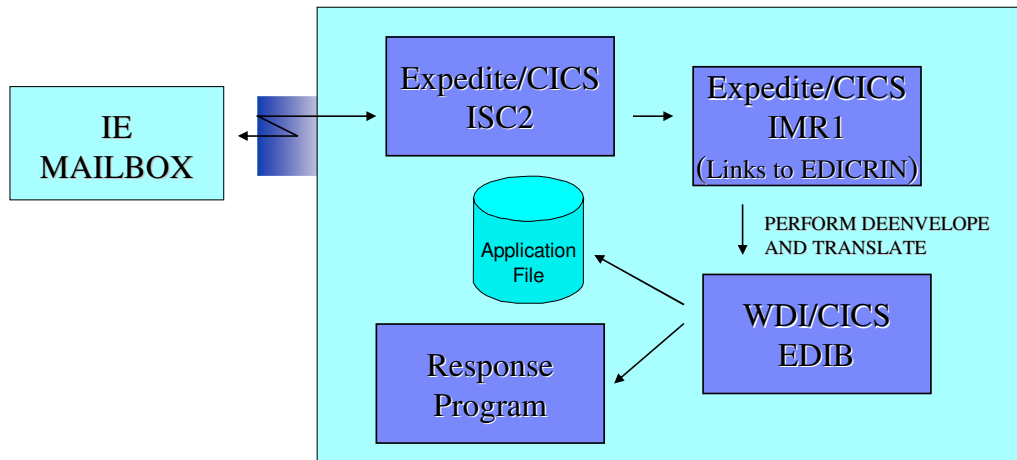


More technically, this is the Expedite/CICS Continuous Receive flow. The idea as far as WDI is concerned, is that the WDI Utility transaction, EDIB, gets invoked with a PERFORM command constructed from information in the Continuous Receive profile.

Expedite/CICS Continuous Receive - continued

- Continuous Receive Flow

- IE → Expedite/CICS → WDI



19

CICS Continuous Receive

© 2007 IBM Corporation

This illustrates the same flow as the previous slide. The example here shows WDI getting invoked with a PERFORM DEENVELOPE AND TRANSLATE command. Translated application data gets written to a file, and then the response program named in the Continuous Receive profile gets executed. This response program would check translation return codes and take appropriate action.

MQ Continuous Receive

1. Define a WDI Queue profile
 - Set profile name to MQI
 - Set queue name to CICS.EDIRECEIVE
2. Define a WDI Continuous Receive profile
 - Set profile name to CRI
 - Leave selection fields blank (i.e. Mailbox)



All three types of Continuous Receive are similar in that the WDI Utility automatically gets invoked with a PERFORM command constructed from information in a Continuous Receive profile. MQ can be set up in CICS to automatically trigger processing when a message is written to a queue. This and subsequent slides explain the necessary setup. First, you define a WDI Queue profile identifying the data queue to monitor. Then you define a WDI Continuous Receive profile identifying the type of WDI processing you want to occur. There is no need to select a mailbox here.

MQ Continuous Receive - continued

3. Define MQ objects

- Define an initiation queue

```
DEFINE QLOCAL(CICSI.TRIGGER)
```

- Define a process

```
DEFINE PROCESS(EDIPROC) APPLICID(EDIQ) APPLTYPE(CICS)
```

- Define the data queue that will receive messages

```
DEFINE QLOCAL(CICSI.EDIRECEIVE) INITQ(CICSI.TRIGGER)  
PROCESS(EDIPROC) TRIGGER TRIGTYPE(FIRST)  
TRIGDATA('CRPROF=CRI MQPROF=MQI')
```

Outside of WDI you need to define several MQ objects. You define an initiation queue, a process, and the data queue that will receive messages. It will be explained later how these objects relate.

MQ Continuous Receive - continued

4. Optionally, set the default queue manager name and initiation queue name in the CICS SIT.

```
INITPARM=(CSQCPARM='SN=MQ65,TN=001,IQ=CICSI.TRIGGER')
```



You may optionally specify the default queue manager name (subsystem name) and the default initiation queue name in the CICS SIT (System Initialization Table). If you connect to MQ during CICS system initialization, this is the queue manager name used. This is an optional step, but it can make subsequent steps a tad easier.

MQ Continuous Receive - continued

5. Start an MQ connection in CICS

- The CICS adapter control panel (transaction CKQC)
 - ▶ Select "Connection" then "Start"
 - ▶ Displays screen where you enter connection parameters
- The CICS command line
 - ▶ CKQC START Y MQ65 001 CICS.TRIGGER
- A CICS application program
 - ▶ EXEC CICS LINK PROGRAM('CSQCQCON') . . .
- CICS SIT parameter -or- CICS initialization PLT entry
 - ▶ SIT Parameter: MQCONN=YES
 - ▶ PLTPI Entry: DFHPLT TYPE=ENTRY,PROGRAM=CSQCCODF

There are several ways to connect MQ to CICS. You may use transaction CKQC. This allows you to enter connection parameters on screens that are displayed. You may enter CKQC START on a blank CICS screen. If explicit parms are included here, they are positional and must each be the maximum size. You may start a connection by linking to the adapter connect program, CSQCQCON, from a CICS application program. And finally, you may make the MQ connection during CICS system initialization, either by way of a SIT parameter or an initialization PLT entry.

MQ Continuous Receive - continued

6. Start an instance of the MQ trigger monitor CKTI

- The CICS adapter control panel (transaction CKQC)
 - ▶ Select "CKTI" then "Start"
 - ▶ Displays screen where you enter initiation queue name
- The CICS command line
 - ▶ CKQC STARTCKTI CICS.TRIGGER
- A CICS application program
 - ▶ EXEC CICS LINK PROGRAM('CSQCSSQ ') INPUTMSG('CKQC STARTCKTI CICS.TRIGGER')



You need to start an instance of the MQ trigger monitor. CKTI is used in CICS to start a transaction when trigger conditions on any of its associated queues are met. You may use transaction CKQC. This allows you to enter the name of the initiation queue on a screen that is displayed. You may enter CKQC STARTCKTI on a blank CICS screen with or without an overriding initiation queue name. Within a CICS application you may link to the adapter task initiation program, CSQCSSQ, with appropriate INPUTMSG data.

MQ Continuous Receive - continued

- Continuous Receive Flow
 - A. Data is written to queue CICS.EDIRECEIVE
 - B. TRIGDATA is written to queue CICS.TRIGGER
 - C. The MQ trigger monitor (CKTI) starts application EDIQ, which is specified in process EDIPROC

```
DEFINE QLOCAL(CICS.TRIGGER)
```

```
DEFINE PROCESS(EDIPROC) APPLICID(EDIQ) APPLTYPE(CICS)
```

```
DEFINE QLOCAL(CICS.EDIRECEIVE) INITQ(CICS.TRIGGER)  
PROCESS(EDIPROC) TRIGGER TRIGTYPE(FIRST)  
TRIGDATA('CRPROF=CRI MQPROF=MQI')
```

25

CICS Continuous Receive

© 2007 IBM Corporation

The words in brown are the MQ definitions shown a few slides ago. They are here so you can see how these objects relate. CKTI has previously been started to monitor queues associated with CICS.TRIGGER. The flow starts when a message is written to the data queue, which is being monitored by CKTI. The trigger condition is met (FIRST), so CKTI writes the TRIGDATA to the initiation queue, and starts the CICS transaction ID specified in the PROCESS.

MQ Continuous Receive - continued

- Continuous Receive Flow - continued
 - D. EDIQ does an EXEC CICS RETRIEVE to get the TRIGDATA
 - E. EDIQ then LINKs to program EDICRIN with commarea EXICOMM
 - Continuous Receive profile name ('CRI') in EXIFNAME
 - MQSeries profile name ('MQI') in EXIMQNAM

Note: EXIFNAME and EXIMQNAM are two fields in the EXICOMM data structure



EDIQ does an EXEC CICS RETRIEVE to get the TRIGDATA ('CRPROF=CRI MQPROF=MQI'). EDIQ parses TRIGDATA to determine the Continuous Receive profile name and the WDI Queue profile name (this is the WDI Queue profile that is associated with the MQ data queue). These two names are placed in their respective EXICOMM fields. EXICOMM is an Expedite/CICS-defined data structure. EDIQ then EXEC CICS LINKs to program EDICRIN with commarea EXICOMM.

MQ Continuous Receive - continued

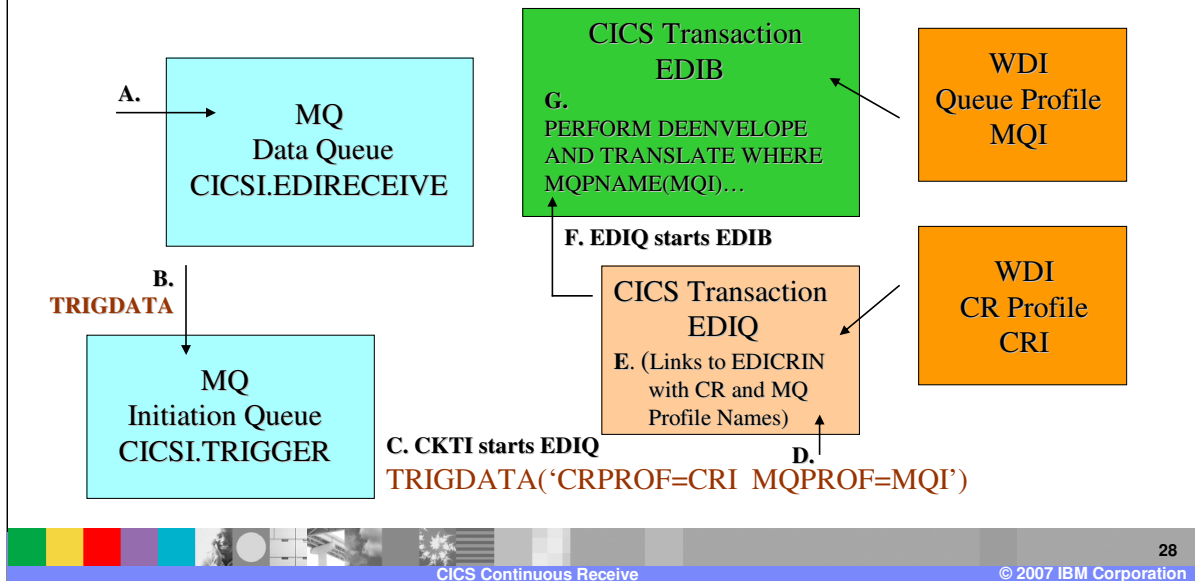
- Continuous Receive Flow - continued
 - F. EDICRIN starts the WDI Utility transaction EDIB
 - An FFUS block is passed along with a PERFORM command containing keyword MQPNAME and the name of the WDI Queue profile:
MQPNAME(MQI)
 - G. The Utility does what is instructed in the Continuous Receive profile
 - The queue named in the WDI Queue profile is processed



Program EDICRIN reads the Continuous Receive profile and constructs a Utility Control Information block and a PERFORM command. On the PERFORM command is placed the MQPNAME keyword with the WDI Queue profile name specified in TRIGDATA. EDICRIN then starts CICS transaction EDIB, which is the WDI Utility transaction. EDIB processes the PERFORM command.

MQ Continuous Receive - continued

- Continuous Receive Flow



This slide pictorially shows the MQ Continuous Receive flow. The letters are associated with the steps just mentioned. The flow starts when a message is written to the data queue. Steps A through G are executed. And it ends when EDIB executes the PERFORM DEENVELOPE AND TRANSLATE command. The next slide has a follow-up on this.

MQ Continuous Receive - continued

- The WDI Utility is invoked with PERFORM command:
`PERFORM DEENVELOPE AND TRANSLATE WHERE
MQPNAME(MQI)...`
- MQI is the name of the WDI Queue profile. Within this profile is MQ queue name CICS.EDIRECEIVE
- The Utility deenvelopes and translates the data in CICS.EDIRECEIVE
 - ▶ Data was written to CICS.EDIRECEIVE back in Step A. This triggered the Continuous Receive process which ended with the data being deenveloped and translated.



The input for the PERFORM DEENVELOPE AND TRANSLATE command is determined by the presence of the MQPNAME keyword. The value in this keyword is the name of a WDI Queue profile. Within that profile is the name of an MQ queue, which is processed as the input to the PERFORM command.

Summary

- Three new types of Continuous Receive processing for WDI 3.3
 - ▶ PERFORM TRANSFORM WHERE ...
 - ▶ PERFORM PROCESS WHERE ...
 - ▶ PERFORM RECEIVE AND PROCESS WHERE ...
- XML print file, ADF print file, and tracking file added to Continuous Receive profile
- Client Continuous Receive profile changed
 - ▶ Tabs for each type of processing (PERFORM command)
 - ▶ Within each tab are fields associated with the PERFORM command's keywords

With regard to 3.3 changes, we saw that three new types of processing can occur via Continuous Receive. We saw that the Continuous Receive profile now accommodates XML, ADF, and tracking file names and types. And we briefly talked about the Client Continuous Receive profile's new look. There are now many more tabs. One for each type of processing. Within the tabs are fields associated with that PERFORM command's keywords. We also reviewed response programs and Expedite and MQ Continuous Receives. Thank you.

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	WMO	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 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.

