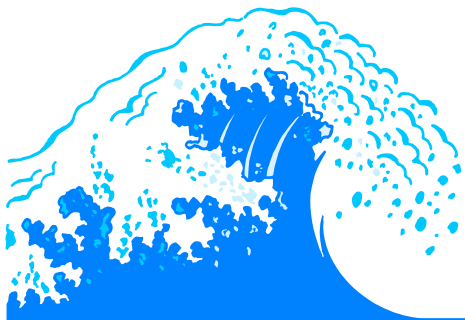


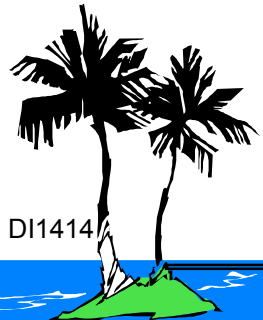
1998 DI USERS GROUP

MQSeries and DataInterchange

Richard Bennett
IBM Global Services
DataInterchange Development



DI/EDI RIDE THE WAVE!



DI1414

Agenda

- △ **Overview**
- △ **New MQSeries Queue Profile**
- △ **Reading/Writing Queues**
- △ **Syncpoint Control**
- △ **Which Files Can Use Queues**
- △ **Specifying a Queue**
- △ **Continuous Receive**
- △ **Automate Processes**
- △ **Examples of Use**



Overview

△ Requires:

- ✦ MQSeries for MVS/ESA 1.1.4 or greater
- ✦ DataInterchange 3.1

△ New MQSeries Queue Profile

△ Most Files Can Use Queues

△ Network Profile for Send/Receiving

△ Implement Real-time Translation Support



MQSeries Queue Profile

New Profile

**Associates Logical Name With a
Physical MQSeries Queue**

**Created/Maintained Like Other
Profiles**



MQSeries Queue Profile (cont.)

Development - MQSeries Queues - QSAMP1

General Comments

Queue Profile ID: QSAMP1

Full Queue Name: Q.SAMPLE.INQ

Queue Manager Name: QMGR.SAMPLE

Description: Sample Queue Profile

Maximum Message Length:

Destructive Reads

Syncpoint Control

MQSeries Queue Profile (cont.)

▲ Fields

- ✦ **Queue Profile ID**
 - Usual profile naming rules
- ✦ **Full Queue Name**
 - MQSeries naming rules
- ✦ **Queue Manager Name**
 - MQSeries naming rules
 - Optional
- ✦ **Description**
 - Limited to 30 characters
- ✦ **Maximum Message Length**
 - Defines maximum size of queue message
 - Used to allocate memory buffers in DataInterchange
 - Defaults to 32700



MQSeries Queue Profile (cont.)

△ Fields (cont)

✦ Destructive Reads

- Used to indicate whether messages read from a queue will be deleted by MQSeries after being read

✦ Default is Yes - Read is Destructive

✦ Syncpoint Control

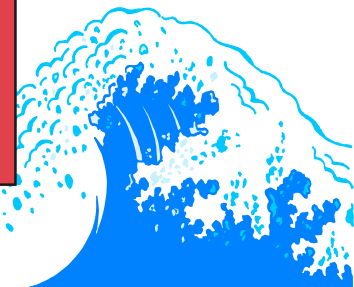
- Used to indicate whether Reading/Writing messages will be part of the unit of work
- Default is Yes



Reading/Writing Queues

Reading

<p>First Read MsgId and CorrelID will be set to nulls</p>
<p>Subsequent reads reuse MsgId obtained from first read, CorrelId is always reset to nulls</p>
<p>When "No messages available", MsgId set to nulls and read from queue again</p>
<p>Method ensures FIFO Processing within messages grouped by MsgId</p>
<p>Will process all message groups in a queue</p>



Reading/Writing Queues

Writing

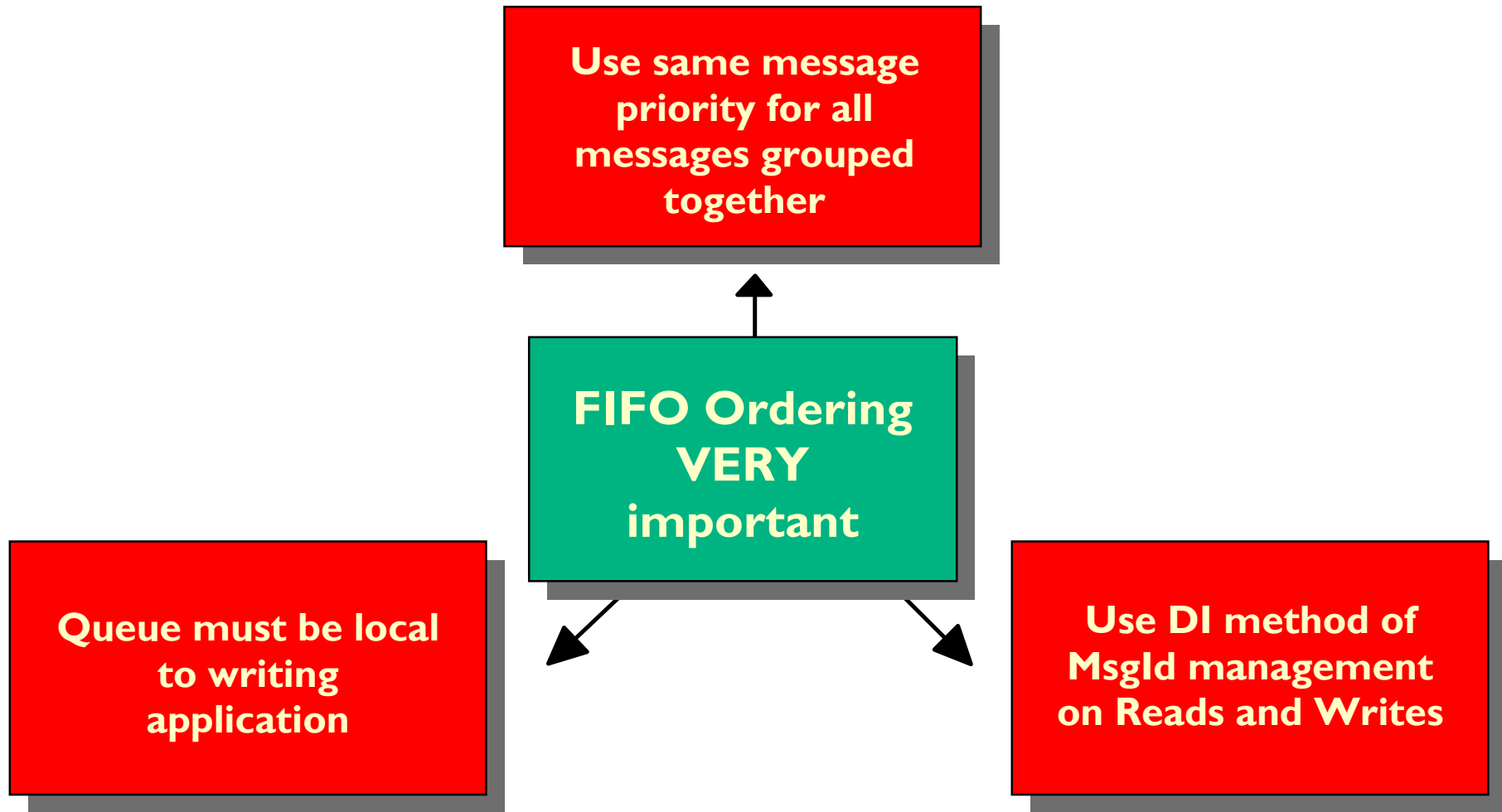
Correlld always set to Nulls

Msgld generated by MQSeries on First Write, then reused on subsequent writes to same queue



Reading/Writing Queues (cont.)

Message Ordering



Syncpoint Control

**MQSeries message
queues can participate
in unit of work**

**Works similar to DB2
in both MVS and CICS
environments**

**Controlled in
queue profile**

Which Files can use Queues

- ▲ **Print File**
- ▲ **Report File**
- ▲ **Query File**
- ▲ **Exception File**
- ▲ **Tracking File**
- ▲ **Send/Receive**
- ▲ **Application File**
- ▲ **Network Acknowledgement File**
- ▲ **Export/Import Control File**
- ▲ **Mapping Migration Control File**
- ▲ **SAP Status Extract File**
- ▲ **SYSIN (Command File)**

Envelope File Can Not Use a Queue



Specifying a Queue

- △ **Specify Queue Profile name in “File” field**
 - ✦ Applies to Panels, Keywords and Control Blocks
- △ **Specify Type “MQ” if “TYPE” field exists**
 - ✦ Applies to Panels, Keywords and Control Blocks
 - ✦ “TYPE” Fields now used in MVS and CICS
 - ✦ For example:
 - APPFILE = “QPROFMBR”
 - APPTYPE = “MQ”
- △ **When Fully Qualified Data Set Name is expected, Specify Queue Profile Member With “:MQ” Appended**
 - For Example: “QPROFMBR:MQ”
- △ **Use New JCL Parameters to specify a Queue Profile Member**
- △ **Send/Receive uses Network Profile**

Specifying a Queue

▲ Print File

✦ CICS Panels

- Use Queue Profile Member in Print File Field
- Specify the Print File Type as “MQ”
- (TF52) Specify Exception and Print File Panel in Transaction Store Facility
- (TF60) Receive Transactions Panel in Transaction Store Facility
- Transaction EDIW

✦ MVS Panels

- No “Type” Field
- Specify Queue Profile Member as Data Set Name With “:MQ” Appended to It
- (TF52) Specify Exception and Print File Panel in Transaction Store Facility
- (TF60) Receive Transactions Panel in Transaction Store Facility
- (FF11) DataInterchange MVS Utility Invocation

Specifying a Queue

△ Print File (cont)

- ✦ CICS Utility Control Block
 - Specify Queue Profile Member in PRTNAME
 - Set PRTTYPE to “MQ”
- ✦ New MVS JCL Parameter
 - Identifies Queue Profile Member
 - MQPRT=qprofmbr

△ Report File

- ✦ CICS Panels
 - Use Queue Profile Member in Report File Field
 - Specify the Report File Type as “MQ”
 - Transaction EDIW



Specifying a Queue

△ Report File (cont)

✦ MVS Panels

- No “Type” field
- Specify Queue Profile member as Data Set Name with “:MQ” appended to it
- (FF11) DataInterchange MVS Utility invocation

✦ CICS Utility Control Block

- Specify Queue Profile Member in RPTNAME
- Set RPTTYPE to “MQ”

✦ New MVS JCL Parameter

- Identifies Queue Profile member
- MQRPT=qprofnam



Specifying a Queue

△ Query File

✦ CICS Panels

- Use Queue Profile Member in Query File Field
- Specify the Query File Type as “MQ”
- Transaction EDIW

✦ MVS Panels

- No “Type” Field
- Specify Queue Profile Member as Data Set Name With “:MQ” Appended to It
- (FF11) DataInterchange MVS Utility Invocation



Specifying a Queue

△ Query File (cont)

✦ CICS Utility Control Block

- Specify Queue Profile Member in QRYNAME
- Set QRYTYPE to “MQ”

✦ New MVS JCL Parameter

- Identifies Queue Profile Member
- MQQUERY=qprofnam

✦ “Perform Query Profile” Command

- Use Queue Profile Member in OUTFILE Keyword
- Use Value “MQ” in OUTTYPE Keyword (Applies in MVS and CICS Environments)



Specifying a Queue

△ Exception File

✦ CICS Panels

- Use Queue Profile Member in Exception File Field
- Specify the Exception File Type as “MQ”
- (TF52) Specify Exception and Print File Panel in Transaction Store Facility
- Transaction EDIW

✦ MVS Panels

- No “Type” Field
- Specify Queue Profile Member as Data Set Name With “:MQ” Appended to It
- (TF52) Specify Exception and Print File Panel in Transaction Store Facility
- (FF11) DataInterchange MVS Utility Invocation



Specifying a Queue

△ Exception File (cont)

- ✦ CICS Utility Control Block
 - Specify Queue Profile Member in EXCPNAME
 - Set EXCPTYPE to “MQ”
- ✦ New MVS JCL Parameter
 - Identifies Queue Profile Member
 - MQEXCP=qprofnam

△ Tracking File

- ✦ CICS Panels
 - Use Queue Profile Member in Tracking File Field
 - Specify the Tracking File Type as “MQ”
 - Transaction EDIW



Specifying a Queue

△ Tracking File (cont)

✦ MVS Panels

- No “Type” Field
- Specify Queue Profile Member as Data Set Name With “:MQ” Appended to It
- (FF11) DataInterchange MVS Utility Invocation

✦ CICS Utility Control Block

- Specify Queue Profile Member in TRAKNAME
- Set TRAKTYPE to “MQ”

✦ New MVS JCL Parameter

- Identifies Queue Profile Member
- MQTRAK=qprofnam



Specifying a Queue

△ Application File

✦ MVS/CICS Panels

- Use Queue Profile Member in Application File Field
- Specify the Application File Type as “MQ”
- (TD02) Application Data Format
- (TP27) Receive Usage Trading Partner Override

✦ Client Windows

- Use Queue Profile Member in Application File Field
- Specify the Application File Type as “MQ”
- Data Format Editor, “General” Tab
- Receive Usage Editor, “Attributes” Tab



Specifying a Queue

△ Application File (cont)

✦ “Perform” Commands

- Use Queue Profile Member in APPFILE Keyword
- Use Value “MQ” in APPTYPE Keyword (Applies in MVS and CICS Environments)



Specifying a Queue

△ Network Acknowledgement File

✦ CICS Panels

- Use Queue Profile Member in Tracking File Field
- Specify the Tracking File Type as “MQ”
- Transaction EDIW

✦ CICS Utility Control Block

- Specify Queue Profile Member in FILEID
- Set FILETYP to “MQ”

✦ “Perform Process Network Acks” Command

- Use Queue Profile Member in ACKFILE Keyword
- Use Value “MQ” in ACKTYPE Keyword (Applies in MVS and CICS Environments)



Specifying a Queue

△ Export/Import Control File

✦ “Perform Export” Command

- Use Queue Profile Member in CTLFILE Keyword
- Use Value “MQ” in CTLTYPE Keyword (Applies in MVS and CICS Environments)

✦ “Perform Import” Command

- Use Queue Profile Member in CTLFILE Keyword
- Use Value “MQ” in CTLTYPE Keyword (Applies in MVS and CICS Environments)



Specifying a Queue

△ Mapping Migration Control File

- ✦ “Perform Mapping Migration” Command
 - Use Queue Profile Member in INCNTL and OUTCNTL Keywords
 - Use Value “MQ” in INTYPE and OUTTYPE Keywords (Applies in MVS and CICS Environments)

△ SAP Status Extract File

- ✦ “Perform Sap Status Extract” Command
 - Use Queue Profile Member in OUTFILE Keyword
 - Use Value “MQ” in OUTTYPE Keyword (Applies in MVS and CICS environments)



Specifying a Queue

▲ **SYSD (Command File)**

✦ **CICS Panels**

- **Use Queue Profile Member in Command File Field**
- **Specify the Command File Type as “MQ”**
- **Transaction EDIW**

✦ **MVS Panels**

- **No “Type” Field**
- **Specify Queue Profile Member as Data Set Name With “:MQ” Appended to It**
- **(FF11) DataInterchange MVS Utility Invocation**



Specifying a Queue

▲ **SYSIN (Command File) (cont)**

- ✦ **CICS Utility Control Block**
 - **Specify Queue Profile Member in CMDNAME**
 - **Set CMDTYPE to “MQ”**
- ✦ **New MVS JCL Parameter**
- ✦ **Identifies Queue Profile Member**
- ✦ **MQSYSIN=qprofnam**



Specifying a Queue

△ Send/Receive

✦ Create a Network Profile

- Specify Communication Routine as “VANIMQ”
- Use “EDIMQSR” as the Network Program
- In Network Parameters Field:
 - Set “SENDMQ=qprofmbr” for Send
 - Set “RECEIVEMQ=qprofmbr” for Receive
 - Use both (blank separated) for Send and Receive
- No Network Operations Profile Members



Specifying a Queue Send/Receive (cont)

Development - Network Profile - MQSAMP

General | Comments

Network ID: MQSAMP Description:

Comm. Routine: VANIMQ Network Name: Sample MQSeries Network

Network Program: EDIMQSR Network Parameters: SENDMQ=mqprof1 RECEIVEMQ=mqprof2

Input File: Input Record Length:

Trans Data Queue: QDATA Trans Rec Length:

Acks File: Output File:

Msg. Text Header: Msg. Handler: Network Sequence: 00000

Time Zone: GMT System Type: System Level:

Dial Connect: Script Name:

Specifying a Queue

▲ **Send/Receive (cont)**

✦ **To Send:**

- **Specify the MQSeries Network Profile in the Network Id Field of the Trading Partner Profile**

✦ **To Receive:**

- **Specify the MQSeries Network Profile in the Network Id Field of the Requestor Profile (Mailbox on the Client)**

✦ **Each Interchange Must be Contained in a Single Queue Message**

- **Set SEGMENTED to “N” in the Trading Partner Profile When Sending to a Trading Partner**
- **Make Sure Trading Partner Sends to You as a Single Queue Message**

Continuous Receive

- △ **Can Be Set Up to Automatically Process Interchange Data Received on a queue**
- △ **Any Number of Queues Can be Set Up**
- △ **Works Much Like Existing Continuous Receive Process**
- △ **Full End to End Recovery**
- △ **New Transaction "EDIQ"**



Continuous Receive (cont)

△ Define Queue Profile

✦ For Example:

- Set Queue Profile Name to "MQ1"
- Set Queue Name to "CICS1.EDIRECEIVE"

△ Define Continuous Receive Profile

✦ For Example:

- Set Continuous Receive Profile Name to "CR1"
- Leave Selection Fields Blank, i.e. Requestor ID



Continuous Receive (cont)

△ MQSeries Definitions

✦ Define an Initiation Queue to MQSeries

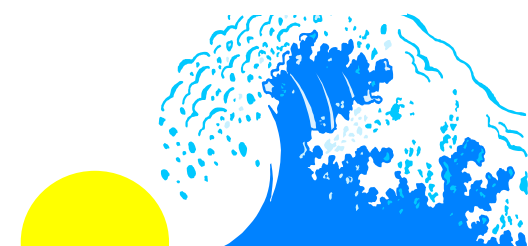
➤ DEFINE QLOCAL(CICS1.TRIGGER)

△ Define a "Process" to MQSeries

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

✦ Define the Queue that Will Receive Interchanges

➤ DEFINE QLOCAL(CICS1.EDIRECEIVE) INITQ(CICS1.TRIGGER)
PROCESS(EDIPROC) TRIGGER TRIGTYPE(FIRST)
TRIGDATA('CRPROF=CR1 MQPROF=MQ1')



Automate Processes

△ Can Fully Automate Translation Processes

- ✦ In MVS and/or CICS
- ✦ On Application Data Side
- ✦ On Send/Receive Side
 - Use Continuous Receive for CICS Receive Side
- ✦ Exception Handling
- ✦ Can Be Used to Implement Real-time EDI Processing on Both MVS and CICS

△ MVS

- ✦ Write Trigger Monitor Program to:
 - Monitor an Initiation Queue
 - When Message Arrives on Queue, Build Perform Commands, Submit Job to Start DataInterchange
- ✦ Define Queue Profile Similar to Continuous Receive
- ✦ MQSeries Definitions Similar to Continuous Receive
 - Specify Your Batch Job Instead of EDIQ
 - TRIGDATA is For Your Use



Automate Processes (cont)

▲ CICS

- ✦ **Write Transaction to Start DataInterchange**
 - **Will Be Started by Trigger Monitor (CKTI)**
 - **May Use Data Passed From Trigger Monitor**
- ✦ **Define Queue Profile Similar to Continuous Receive**
- ✦ **MQSeries Definitions Similar to Continuous Receive**
 - **Specify Your Transaction Instead of EDIQ**
 - **TRIGDATA is For Your Use**



Example of Use

▲ **Generate Data on AIX, Translate and Send**

- ✦ **Application Running on AIX Writes Data to Queue**
- ✦ **Data is Sent to Queue on MVS By Queue Manager**
- ✦ **Trigger Event is Generated Causing Trigger Monitor to Read Trigger Message**
- ✦ **Trigger Monitor Uses Trigger Message QName, ProcessName and TriggerData to:**
 - **Build DataInterchange Perform Commands**
 - **Start MVS Job That Will Invoke DataInterchange**
- ✦ **DataInterchange Reads and Translates Data in the Queue**



Example of Use (cont)

- ✦ **DataInterchange Then Places Translated Data Into Message Queue**
- ✦ **Queue Manager Sends Translated Data to Trading Partner**

▲ **Exception Handling**

- ✦ **DataInterchange Generates an Exception and Places on a Queue**
- ✦ **Exception is Sent by Queue Manager to a Queue on a Windows NT System Resulting in a Trigger Event**
- ✦ **Trigger Application on Windows NT System Starts Process for Handling the Exception**



Conclusion

▲ Examine Data Flow

- ✦ MQSeries Good Choice When Mixing MVS and CICS
 - ✦ MQSeries Good Choice When Applications on Other Platforms are Involved
 - ✦ MQSeries Good Choice for Automating Processes
- ## ▲ MQSeries Excellent Method for Real-time EDI Processing
- ## ▲ Use Syncpoint Control, Especially in CICS

