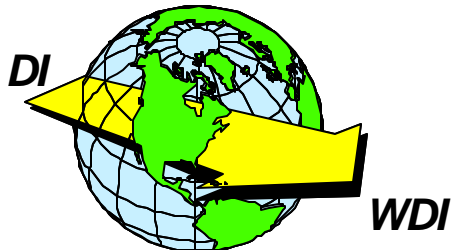


2002 Users Conference

MQ and DI

The Next Generation



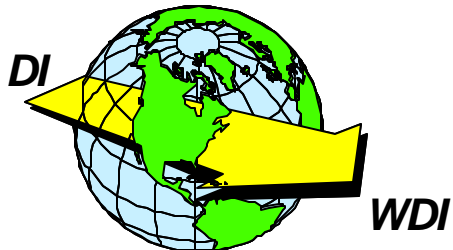
2002 User
Conference

Lee Whitaker
IBM Corporation

WebSphere Data Interchange for Multiplatforms

part of the WebSphere MQ strategy

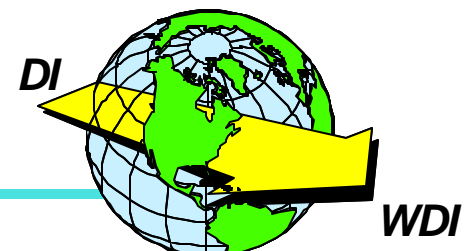
The Next Generation



2002 User
Conference

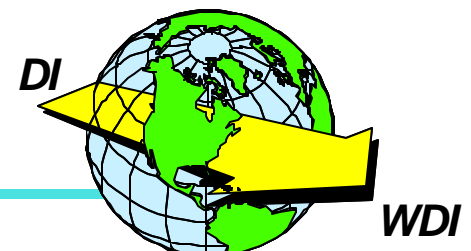
Agenda

- ★ Traditional DataInterchange
- ★ Introducing MQ Series to EDI
- ★ Maturing the MQ to EDI Relationship
- ★ Joining the WebSphere MQ team
- ★ A Vision of the Future

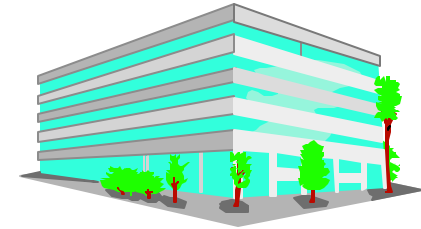
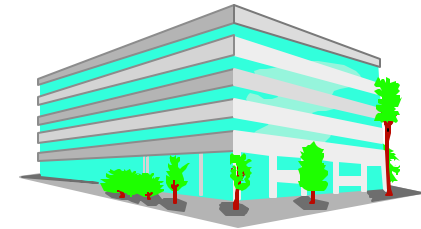
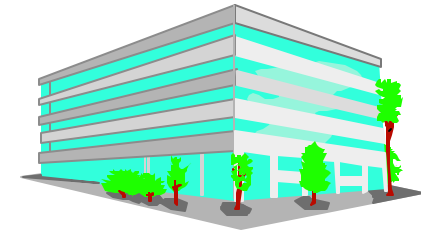
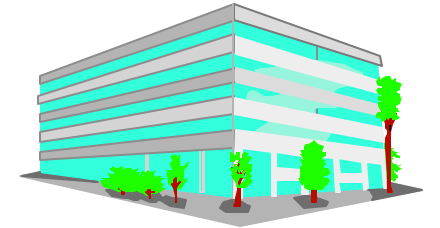
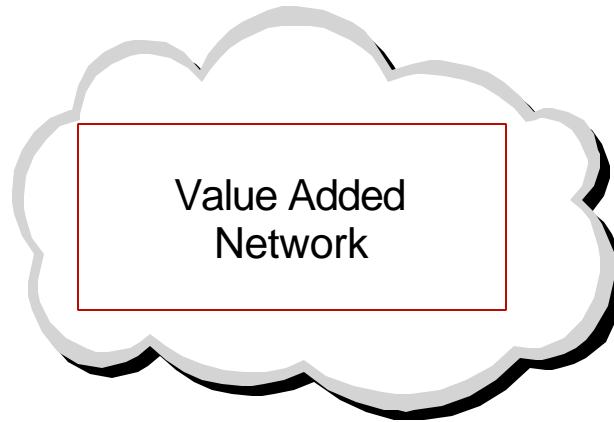


Objectives

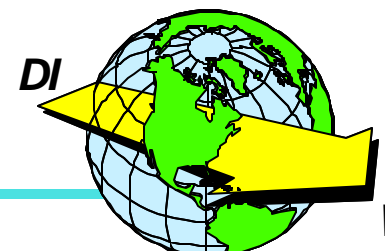
- ★ Experience the EDI and MOM models
- ★ Know how the original DataInterchange charter shaped the WMQ interface
- ★ Expose some messaging concepts and jargon
- ★ Understand how WDI works with WMQ and WMQSI
- ★ Sense the synergy of bringing these models together



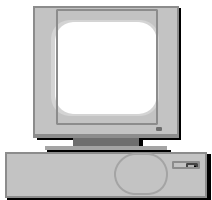
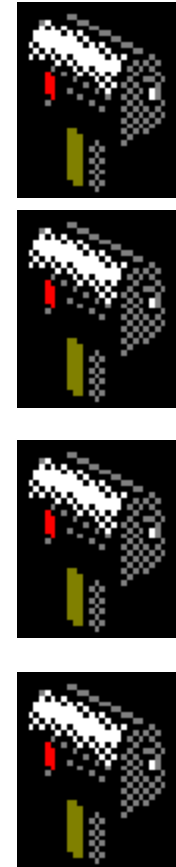
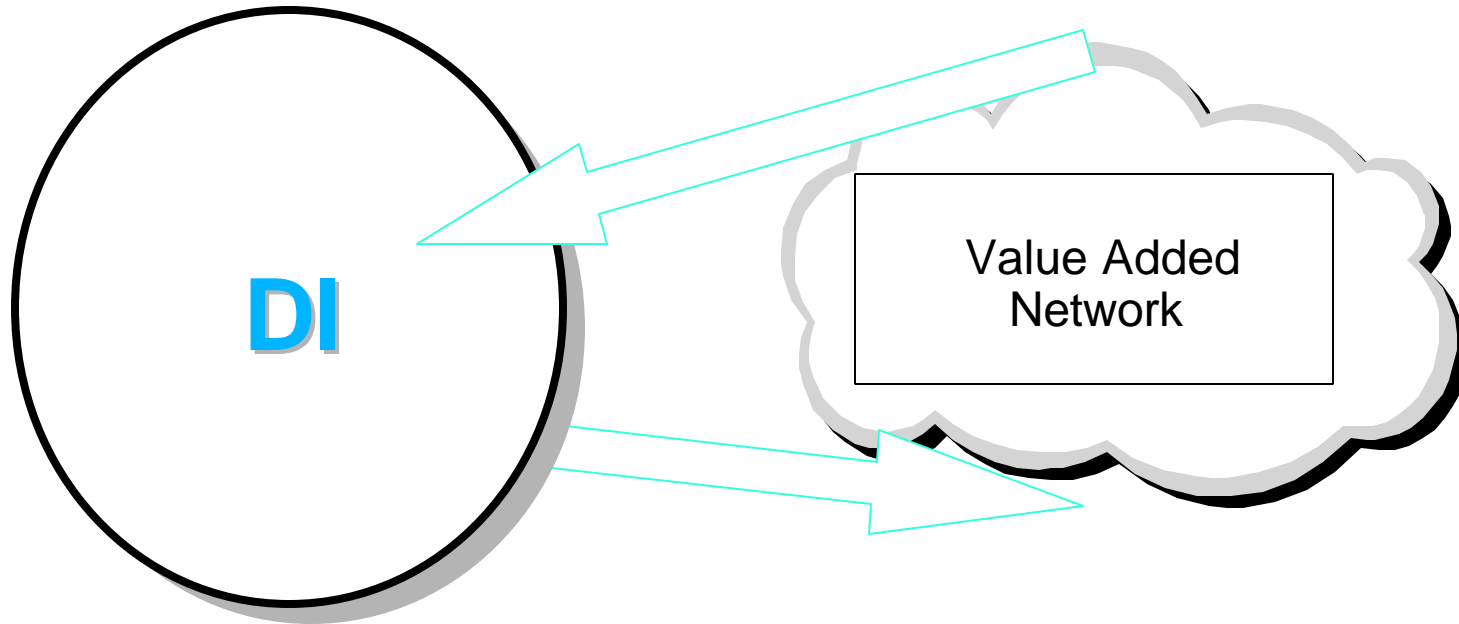
Traditional DataInterchange



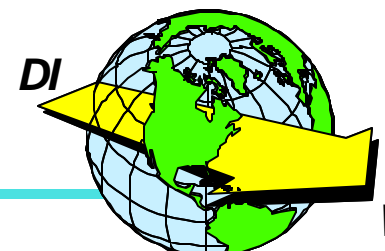
Moving Data In and Out



EDI and the VAN

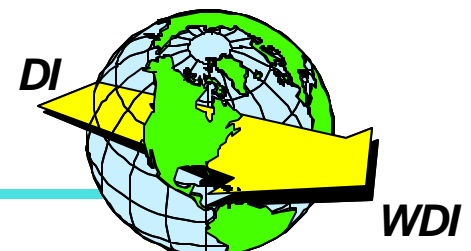


PERFORM RECEIVE WHERE REQID()

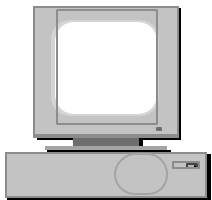
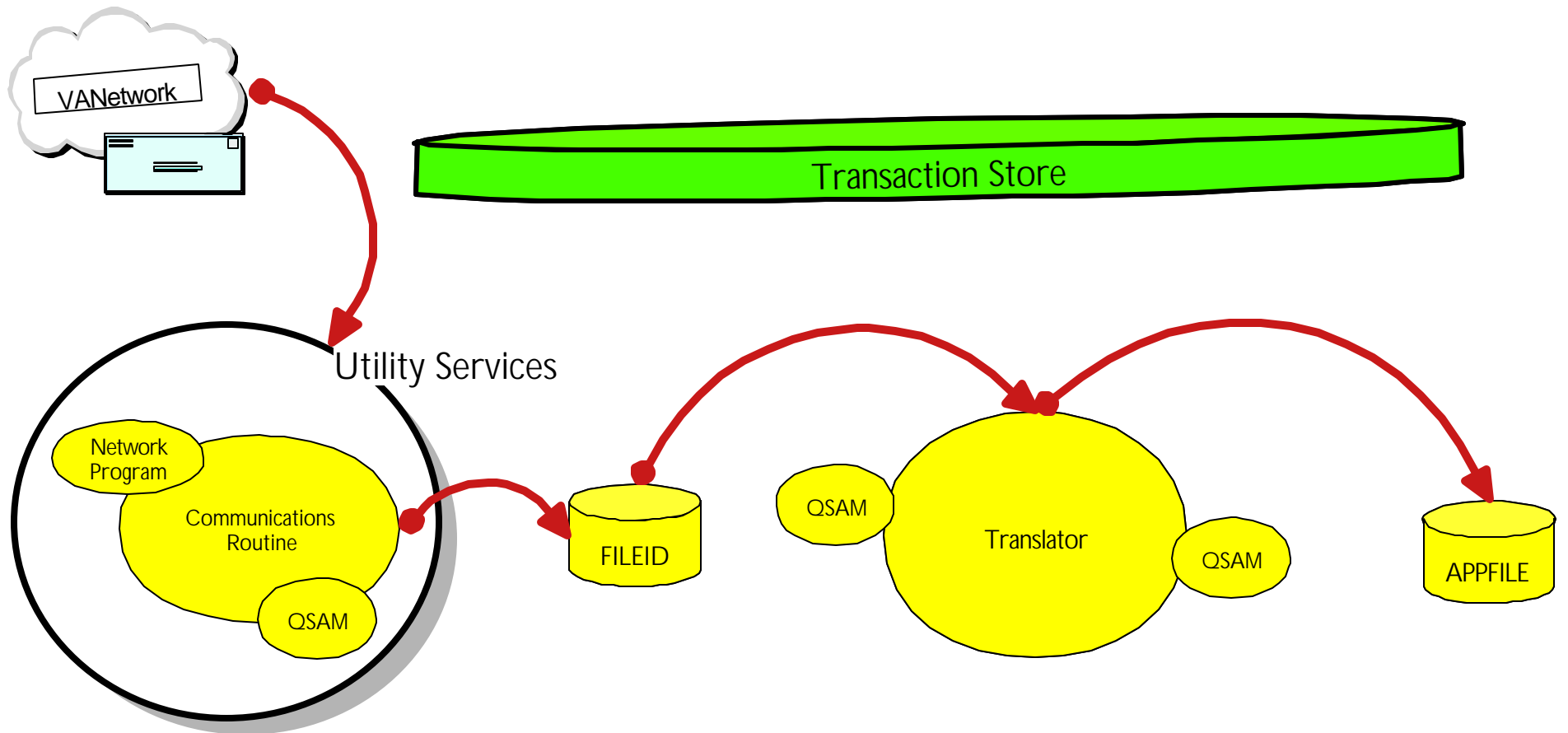


EDI and the VAN

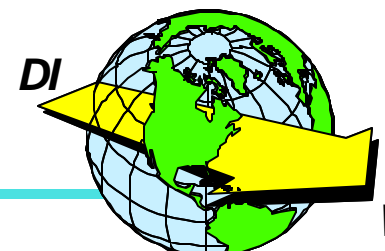
- ★ MAILBOX (a.k.a. REQID) contains connection and data filtering criteria limiting input and defining output
- ★ The communications program will receive control during a PERFORM, and it is expected to pull all data for the given criteria and buffer it in a sequential file.
 - userid / password
 - message class
 - customized command file
- ★ Mailbox is REQUIRED for all VAN connections
- ★ VAN uses this and information embedded in EDI transaction to perform routing



Batch Throughput

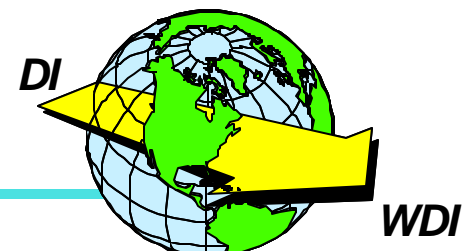


PERFORM RECEIVE AND TRANSLATE



Batch Throughput

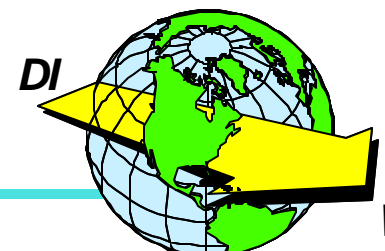
- ★ Single invoked instance
- ★ All data flows Inbound OR Outbound
- ★ Intermediate repository are used
 - transaction store
 - data files
- ★ QSAM service is integral
 - abstracts MVS and CICS data stores



Introducing MQSeries to EDI

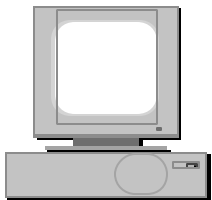
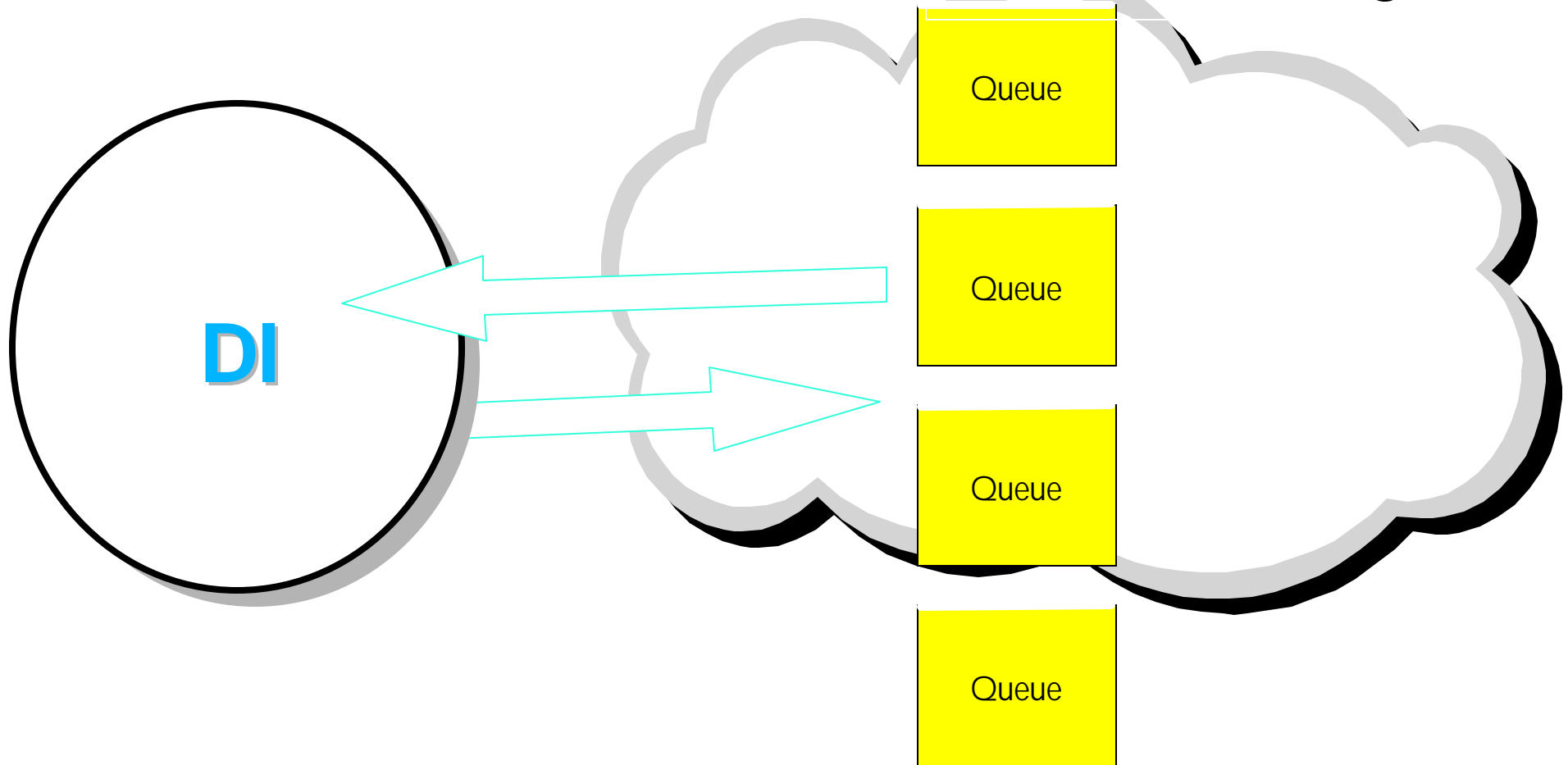


Incorporating MQSeries into DI

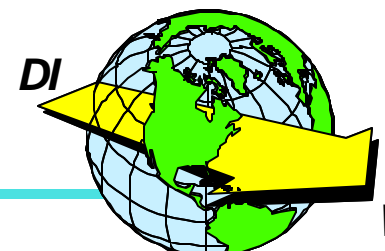


MQSeries First Impression

Queue Manager



PERFORM RECEIVE WHERE REQID()



Mechanisms for Integration

★ MQ Queue Manager modeled as a Network

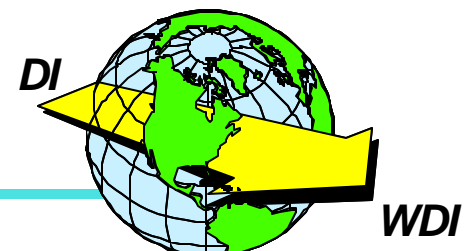
- Using the Network profile
 - ✎ VANIMQ as the Communications Routine
 - ✎ EDIMQS as the Network Program
- the REQID could point to a MQ Network
- separate inbound and outbound
 - ✎ RECEIVEMQ
 - ✎ SENDMQ

★ MQ Queue modeled as a File

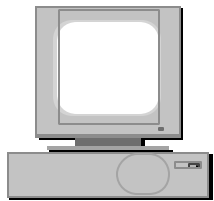
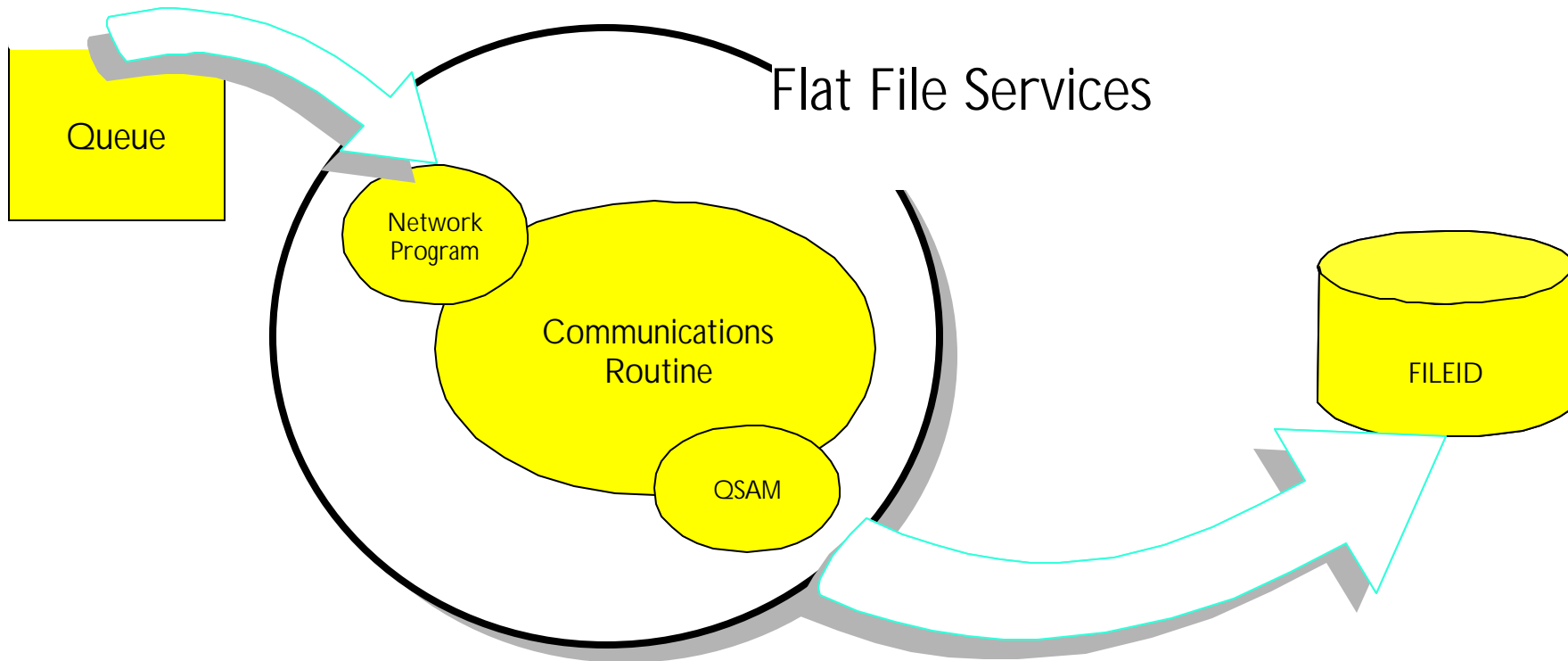
- changes to QSAM Services
- possible wherever FILETYPE could be specified

★ New MQSeries Queue profile

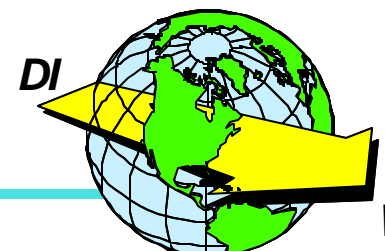
- expands 8 character identifier
- provides QMgrName and QName



Changing the Network

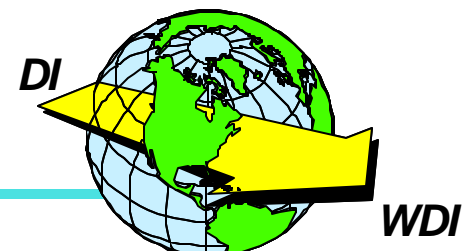


PERFORM RECEIVE WHERE REQID()

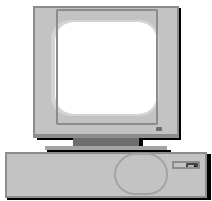
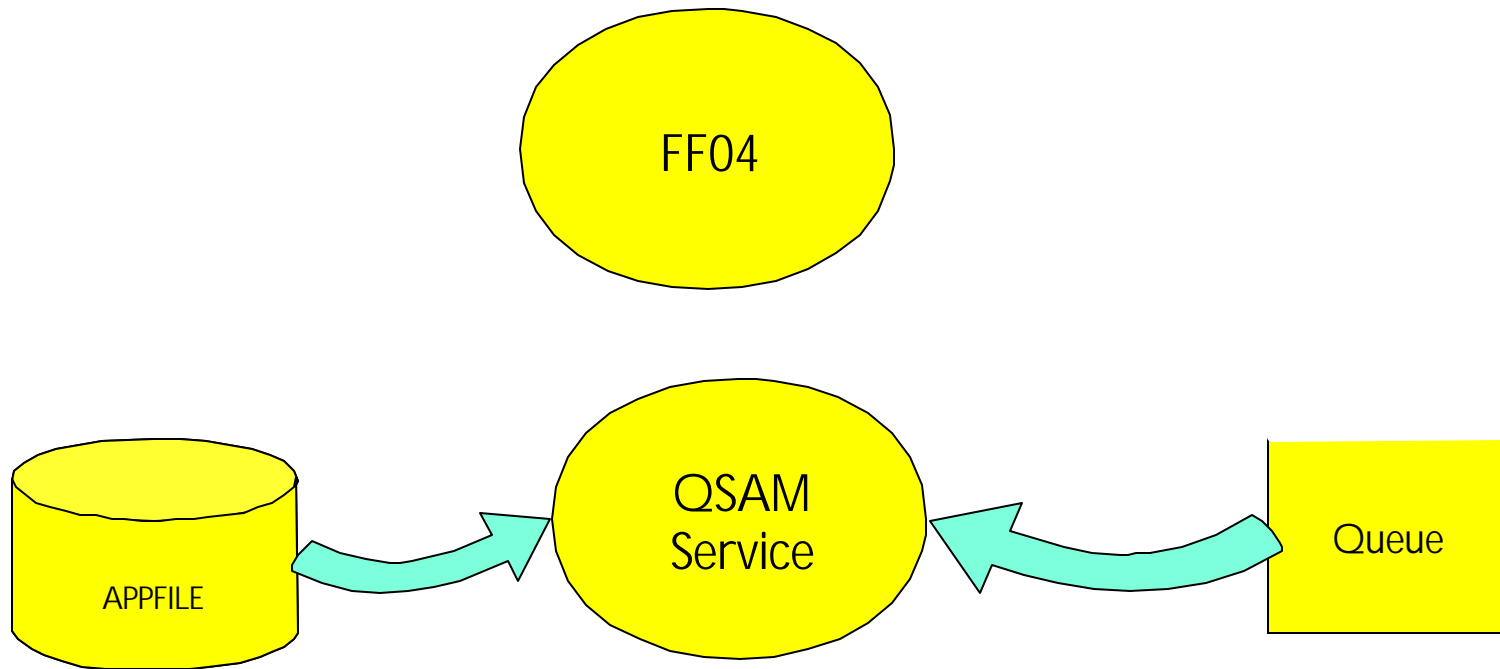


Keys in the MQ network model

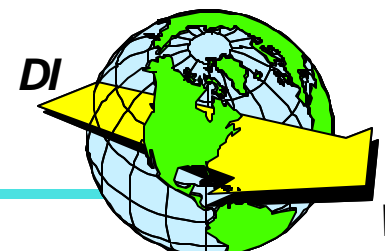
- ★ REQID (Mailbox) identifies one MQ Point to Point Network which identifies a SEND / RECEIVE Queue pair
- ★ Queue profile identifies the full queue name based on an eight character identifier used elsewhere.
- ★ All data is drained from the queue into FILEID.
- ★ On output, all data are aggregated into as few messages as possible.



Changing QSS

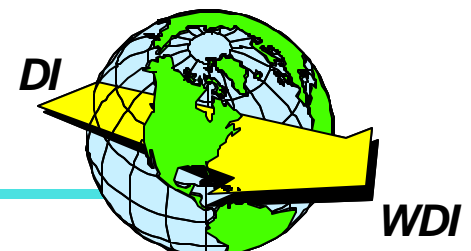


WHERE FILETYPE(MQ)



Keys to the QSAM model

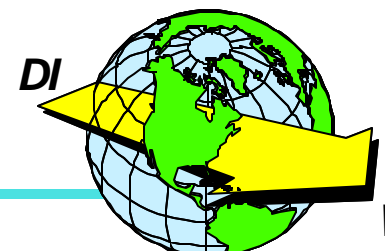
- ★ Data processing is the same for sequential file or MQ Queue
- ★ Record-oriented paradigm



Maturing the MQ to EDI Relationship



Using DI in an MQSeries environment



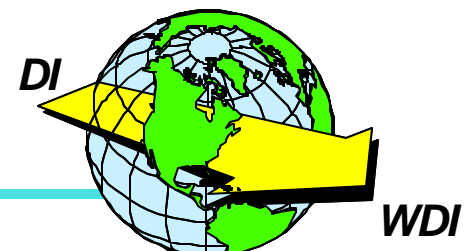
Feedback from MQ Users

★ Concerns with the Network model

- tweaking the profile relationships
 - ✍ limited need for the Mailbox
 - ✍ many queues for 1 queue manager
- messages are atomic and independent
- no way to associate input and output messages

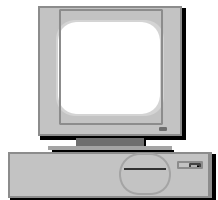
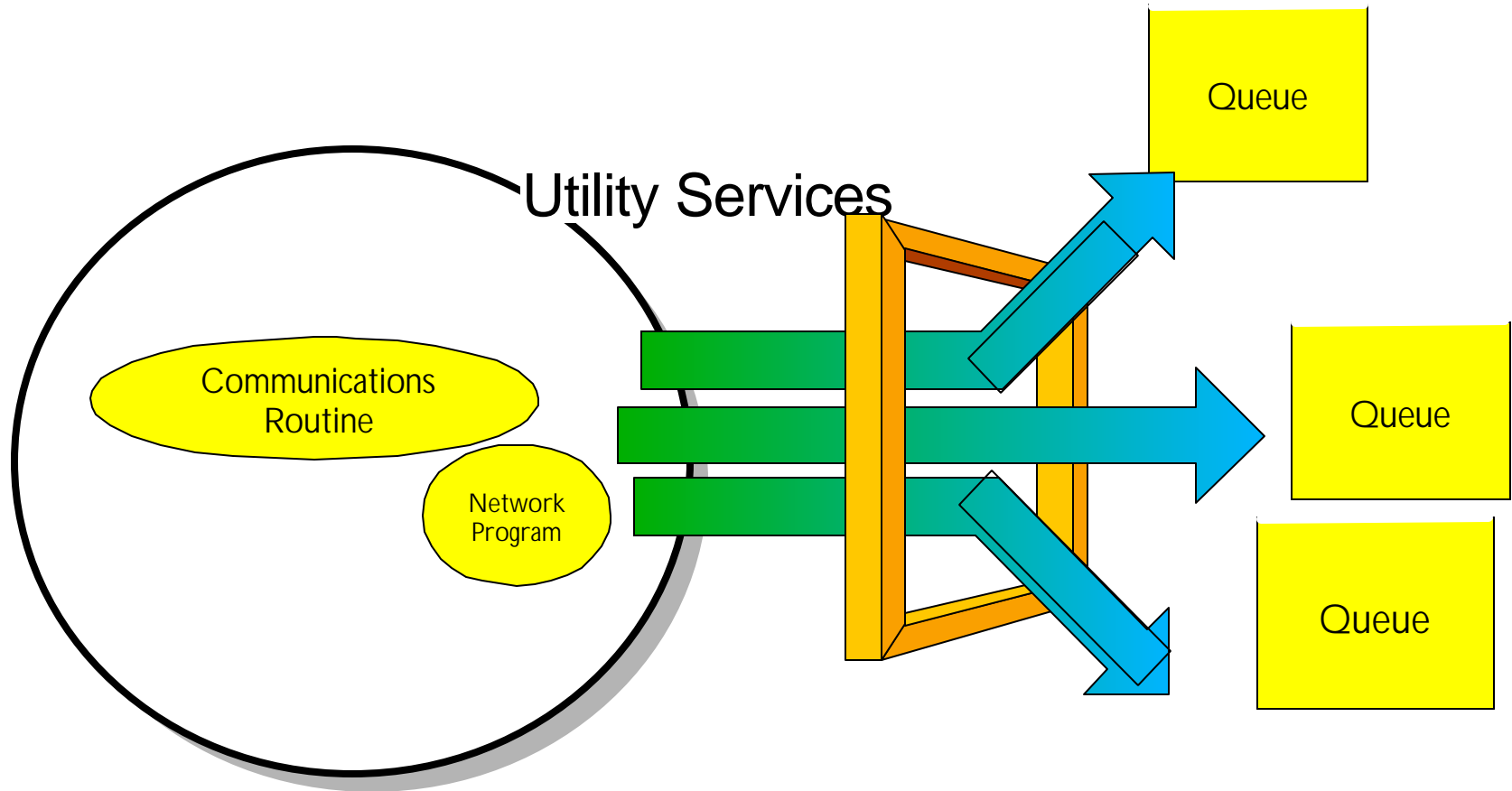
★ Concerns with the QSAM model

- a MQ message is not a file record
 - ✍ message context is important
 - ✍ a document may be made up of many records
 - ✍ a message may contain many documents

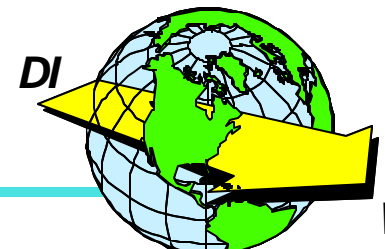


No More Mailbox

★ Not required on a send

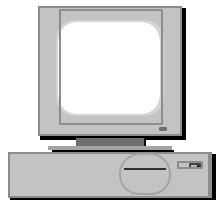


PERFORM TRANSLATE AND SEND

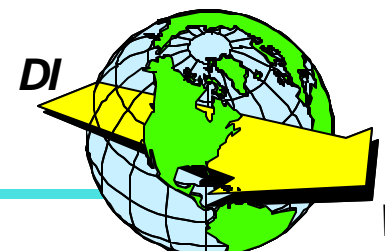


One Message

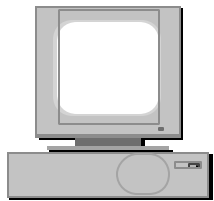
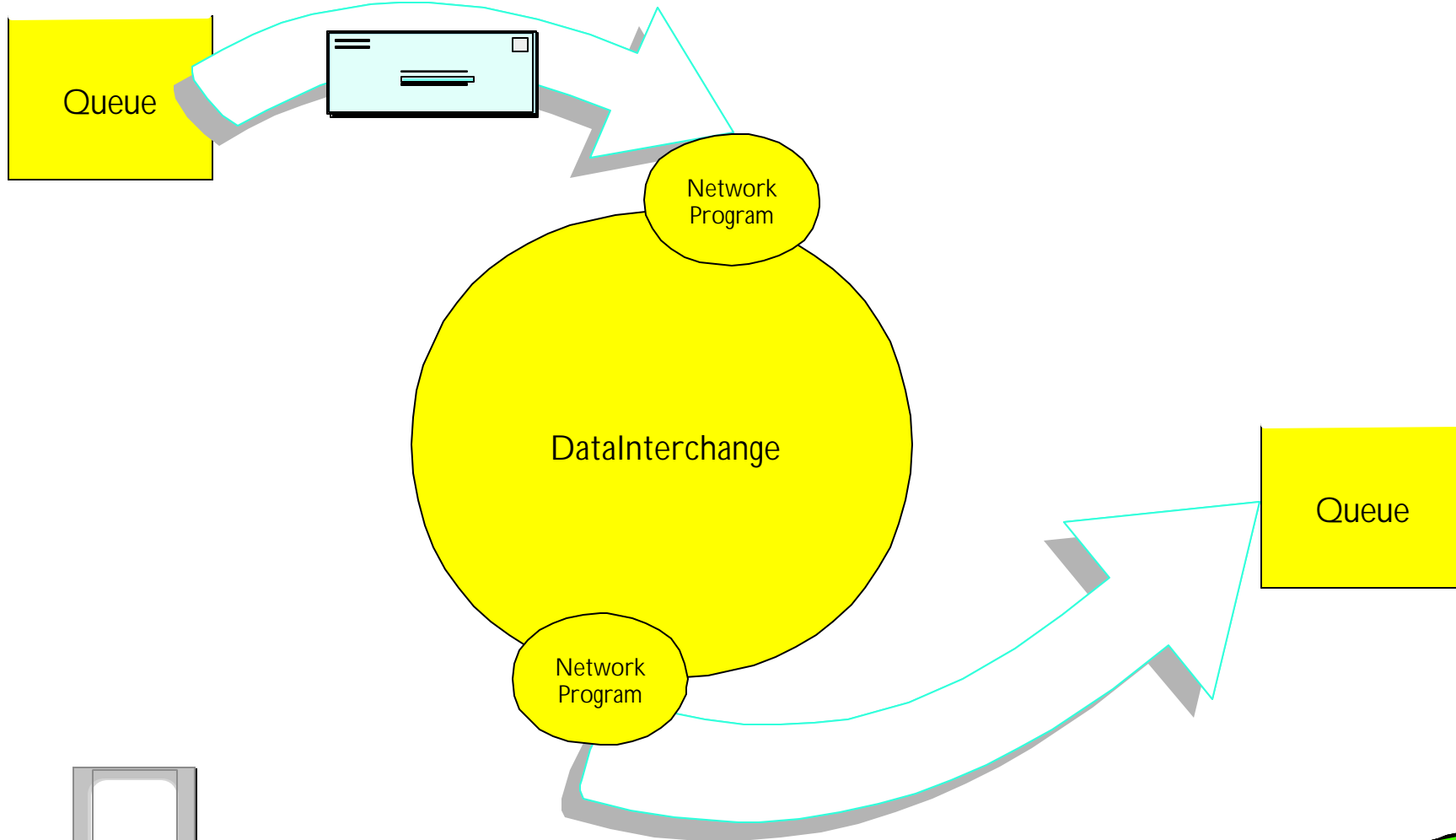
- ★ New key word ONEMSG(Y)
- ★ Single message read from queue
- ★ Logical group of messages processed together
- ★ Queue used in shared mode



WHERE ONEMSG(Y)

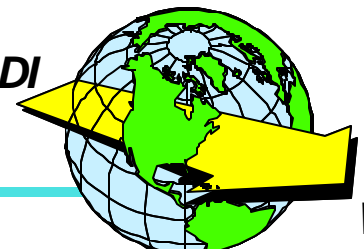


EDI to Data Format



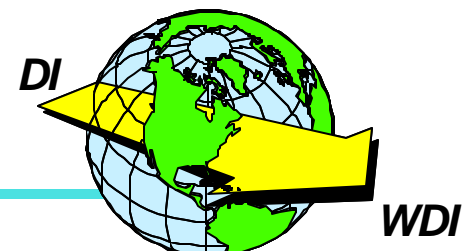
PERFORM RECEIVE AND SEND WHERE REQID() APPFILE()

DI

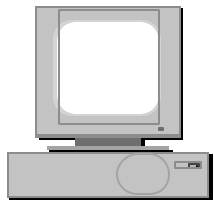
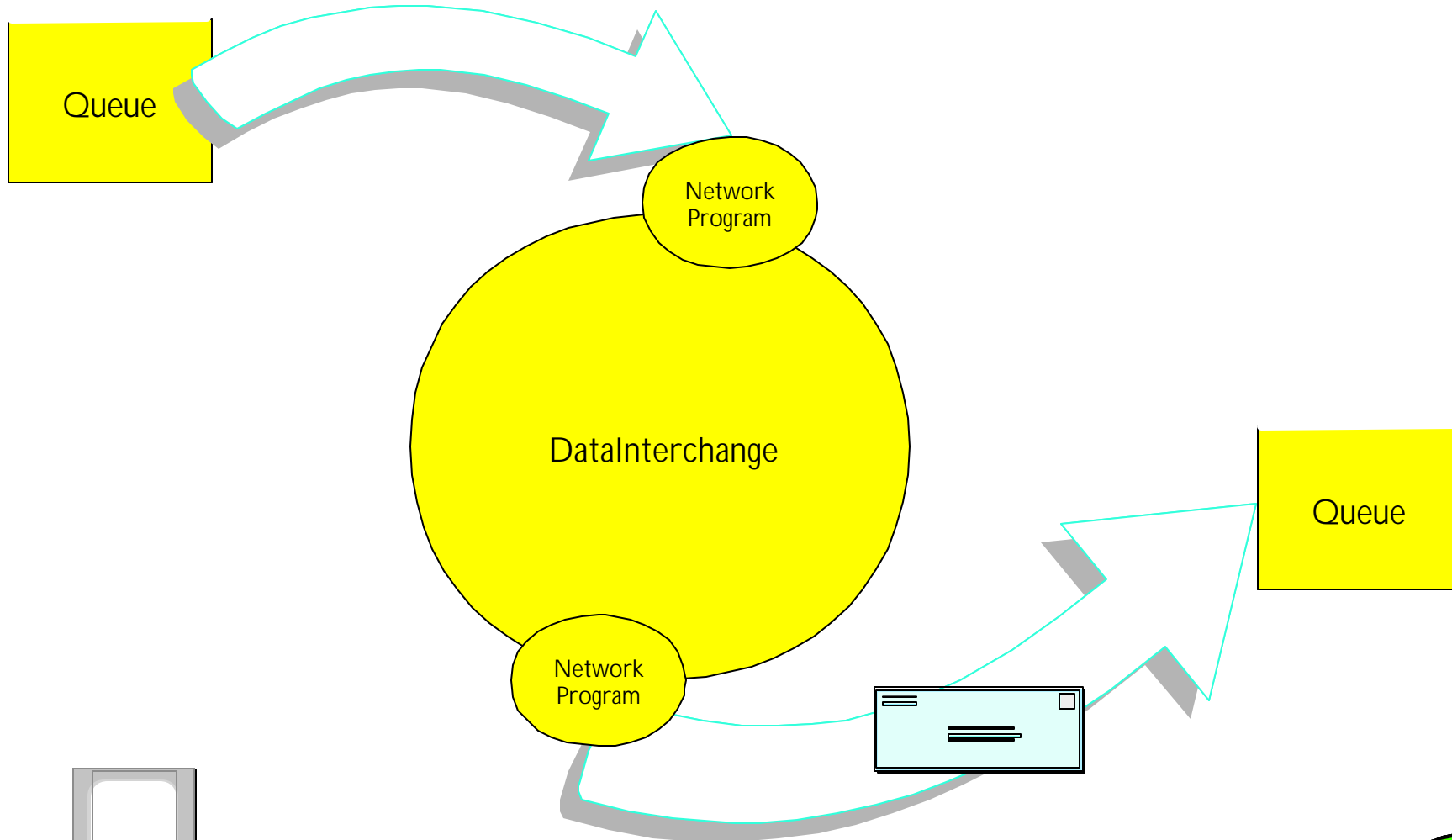


Message Context

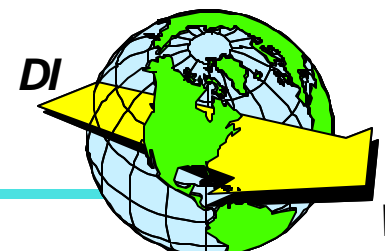
- ★ All context retained from input message
 - message id
 - correlation id
 - group id
- ★ Propagated to all output messages
- ★ Output allows REPLYTO fields
 - propagated with other context
 - used from RECEIVEMQ



Data Format to EDI



PERFORM RECVFILE AND SEND WHERE REQID()
RAWFMTID()



Combined Commands

This Perform command also uses the network interface for both input and output. It differs from the previous command because it expects Application Data as input with EDI resulting output. This command works with the XML PTF to either accept XML as input or produce XML output from Application Data.

Keywords

REQID

This option is used to identify the network source for the data, and it is required.

RAWFMTID

This identifies the ADF format of the input data. It must be supplied whenever the input is non-XML . When the translation target is XML, this keyword is still required.

XML(Y)

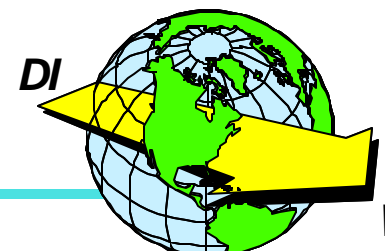
This flag signals DI to invoke the XML PTF. There are several other key words required with this option as noted in the PTF. Using XML input reverses the behavior of FILEID and APPFILE.

FILEID

This specifies the logical file which will contain translated EDI data before it is sent to the network. This option forces all data into a single file which must be sent to a single network target. If this option is omitted, the logical file name is taken from the Tran. Data Queue on the Network Profile. If this is not present, a file name is determined from the Trading Partner nickname. For XML(Y) translations, this is the logical file name where data is stored after it is received from the network.

APPFILE

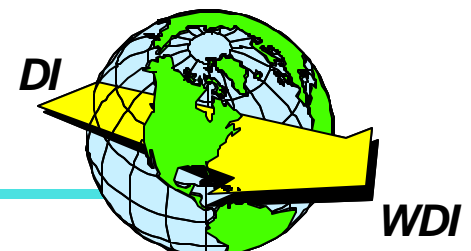
This is the name of the logical file into which data is received from the network. This file name must be supplied for non-XML translations. If it is absent, data will be written to the logical file name specified in the Receive File attribute of the Mailbox profile, but it will not be translated, and an FF0101 error will result. For XML input, this logical file defines where data is queued for subsequent delivery to the output network, and generally should be omitted as with FILEID on a non-XML translation.



Joining the WebSphere MQ team



WDI as part of a WMQ network

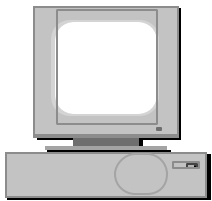


Data Format to EDI

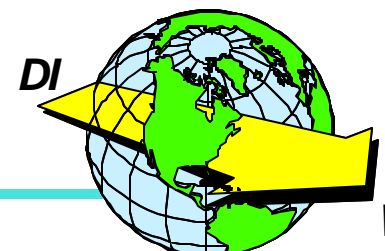
Perform command must identify format

Option 1: each Queue serves a single Data Format

Option 2: use the RFH2 Network Program

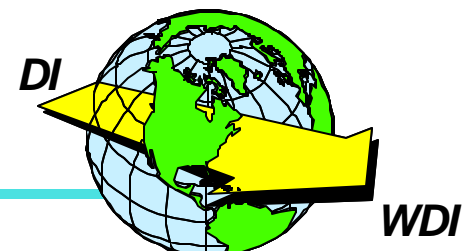


PERFORM RECVFILE AND SEND WHERE REQID()
RAWFMTID()



WMQSI and RFH2

The MQRFH2 structure contains information about the structure and intended consumers of a message that allows an MQSeries Integrator broker to process it and to deliver or publish it to those consumers.



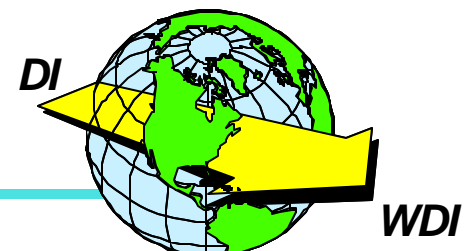
MQRFH2 Message

This message header uses an XML-like syntax to carry information about the message itself.

The flexible *NameValueData* is organized into folders, groups and properties.

```
<folder><group><property>VALUE</property></group></folder>
```

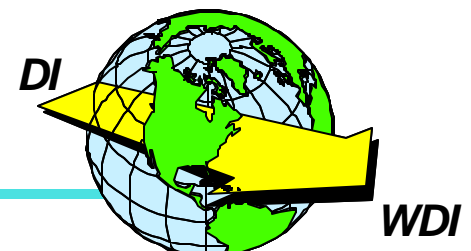
The header may be followed by data or other MQ headers.



MQRFH2 Message

The following folder names are defined for use by MQ products:

- <mcd> Message content descriptor
- <psc> Publish/subscribe command
- <pscr> Publish/subscribe command response
- <usr> Application (user) defined properties



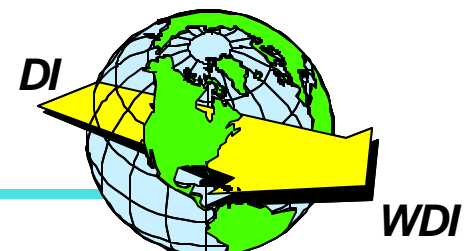
The <mcd> folder

Use in WMQSI

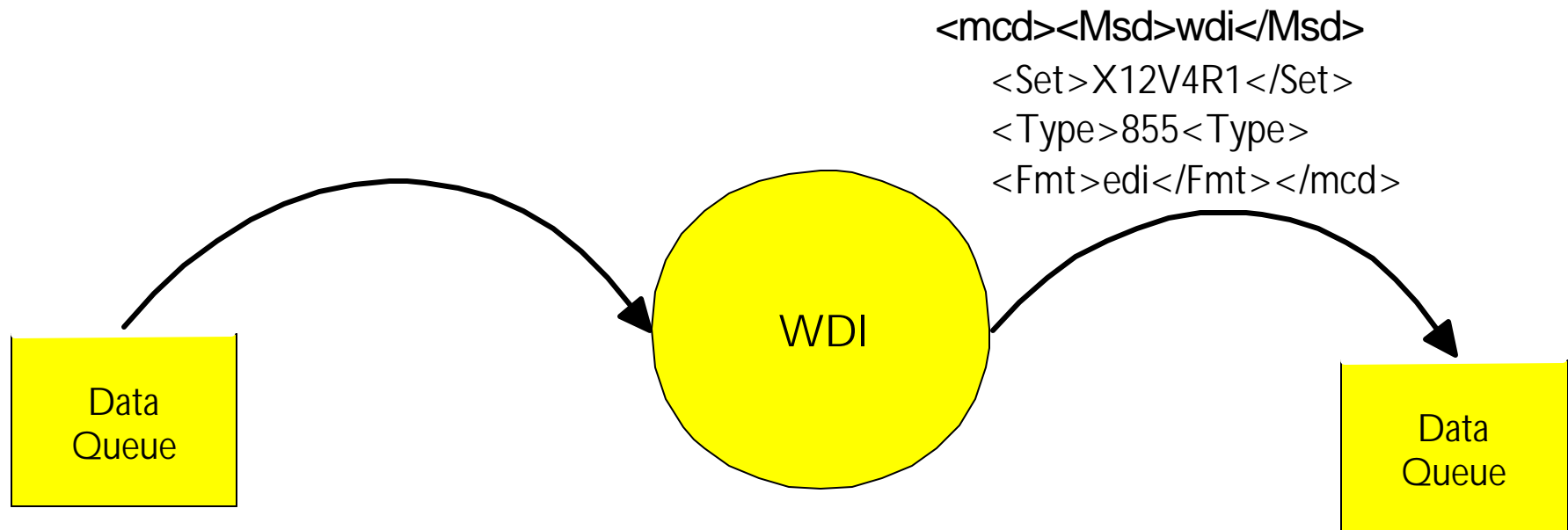
<Msd>	Message service domain
<Set>	Message set
<Type>	Message type
<Fmt>	Message format

"Native" use in WDI

<Msd>	"wdi"
<Set>	Dictionary
<Type>	Transaction Format Document
<Fmt>	Syntax (adf, edi, xml)

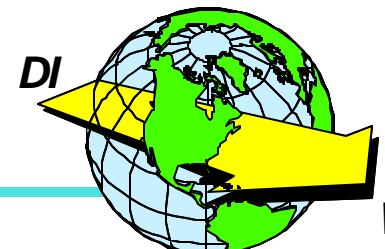


WDI and RFH2

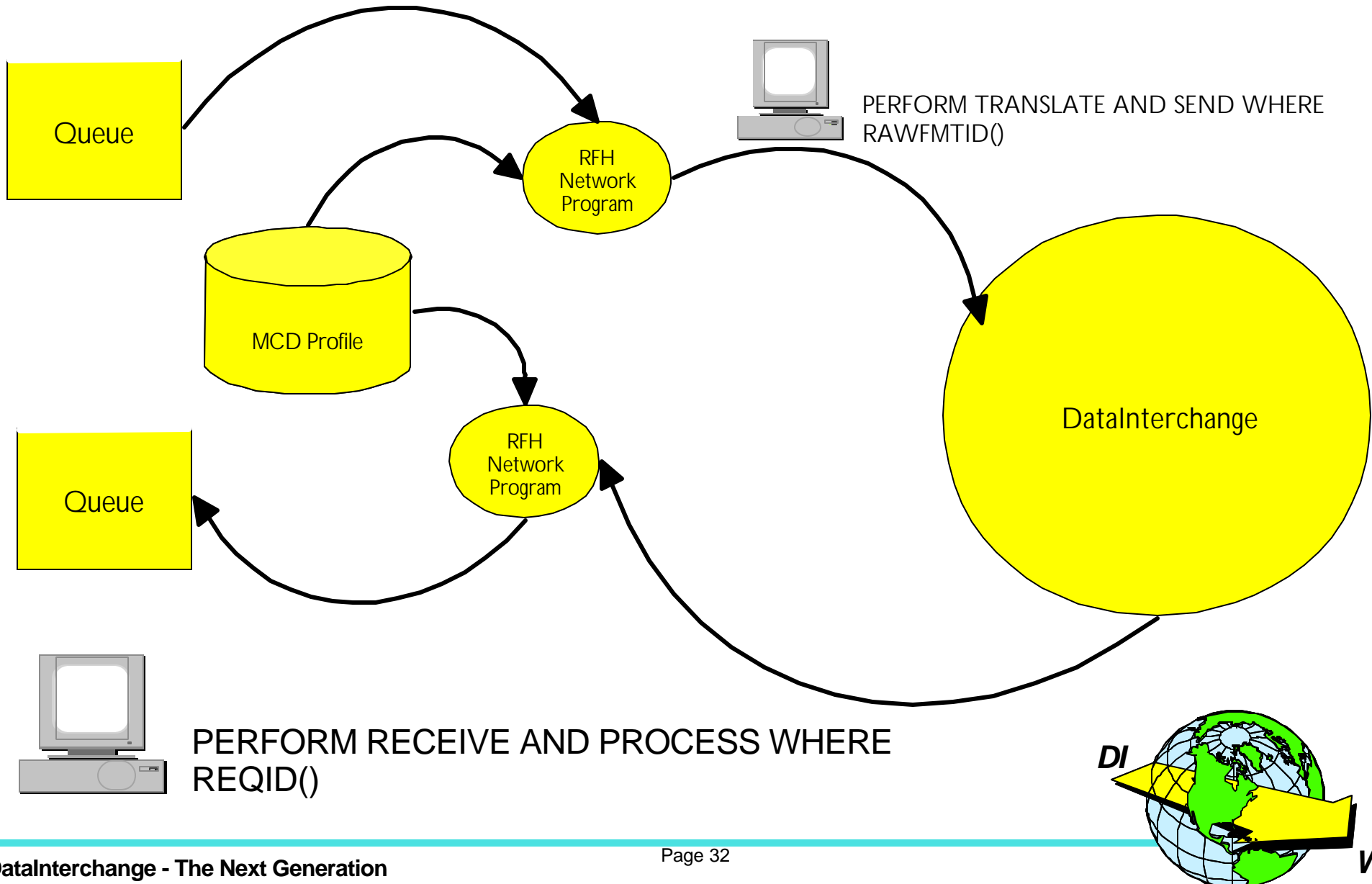


```
<mcd><Msd>wdi</Msd>  
<Set>X12V4R1</Set>  
<Type>855<Type>  
<Fmt>edi</Fmt></mcd>
```

```
<mcd><Msd>mrm</Msd>  
<Set>set001</Set>  
<Type>type001<Type>  
<Fmt>xml</Fmt></mcd>
```



Data Format to EDI using RFH2

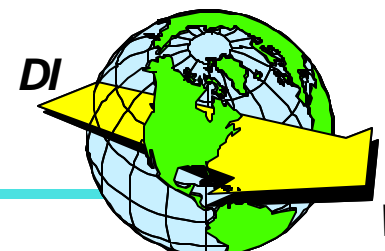


MQ Triggering

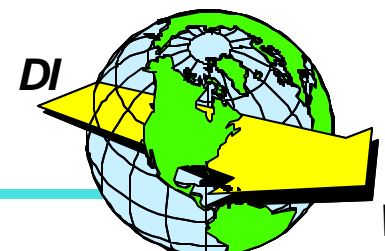
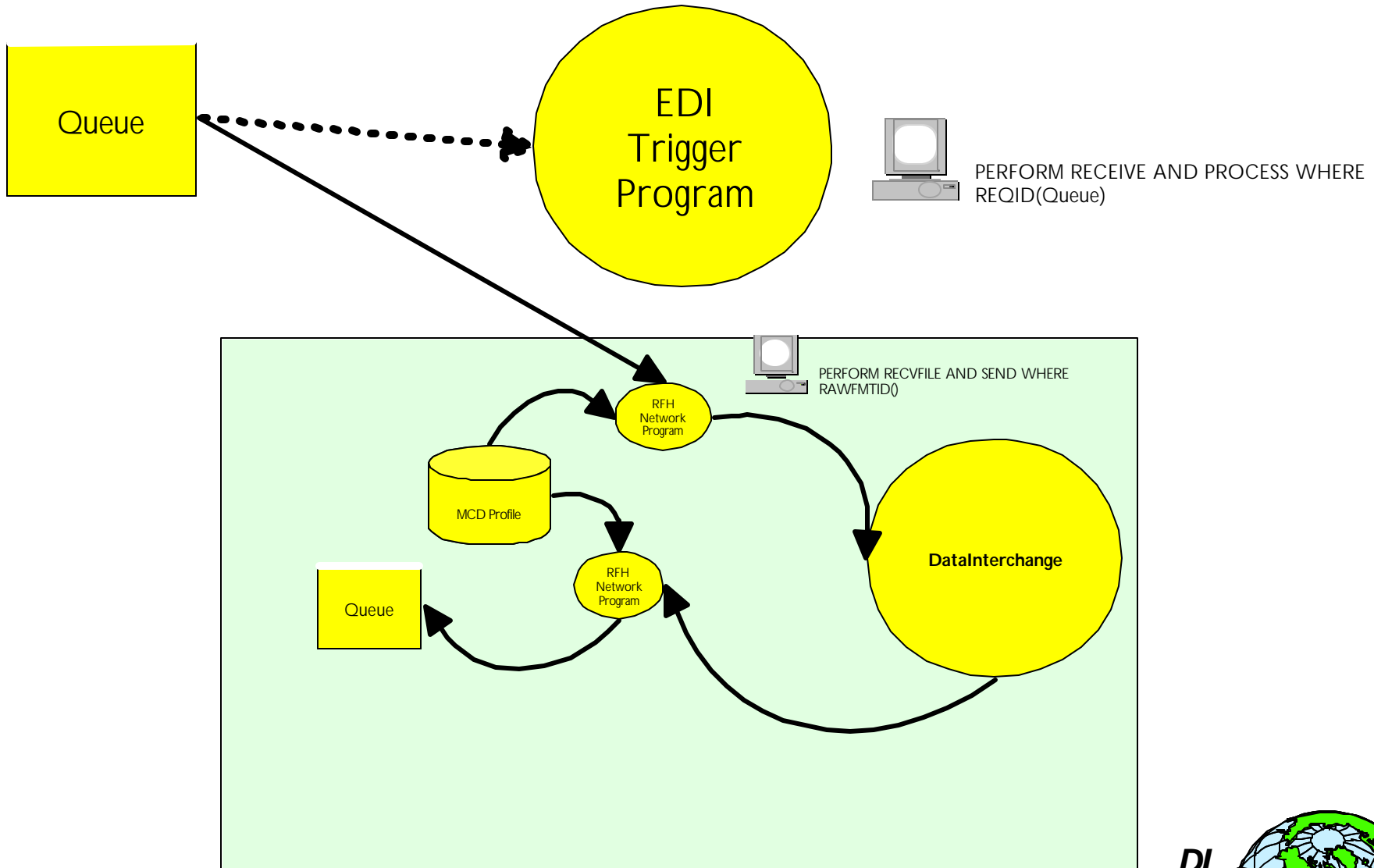
Term

Definition

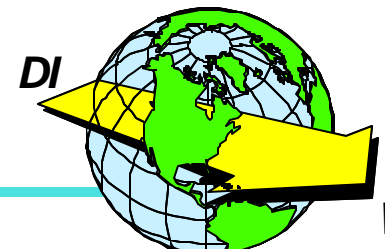
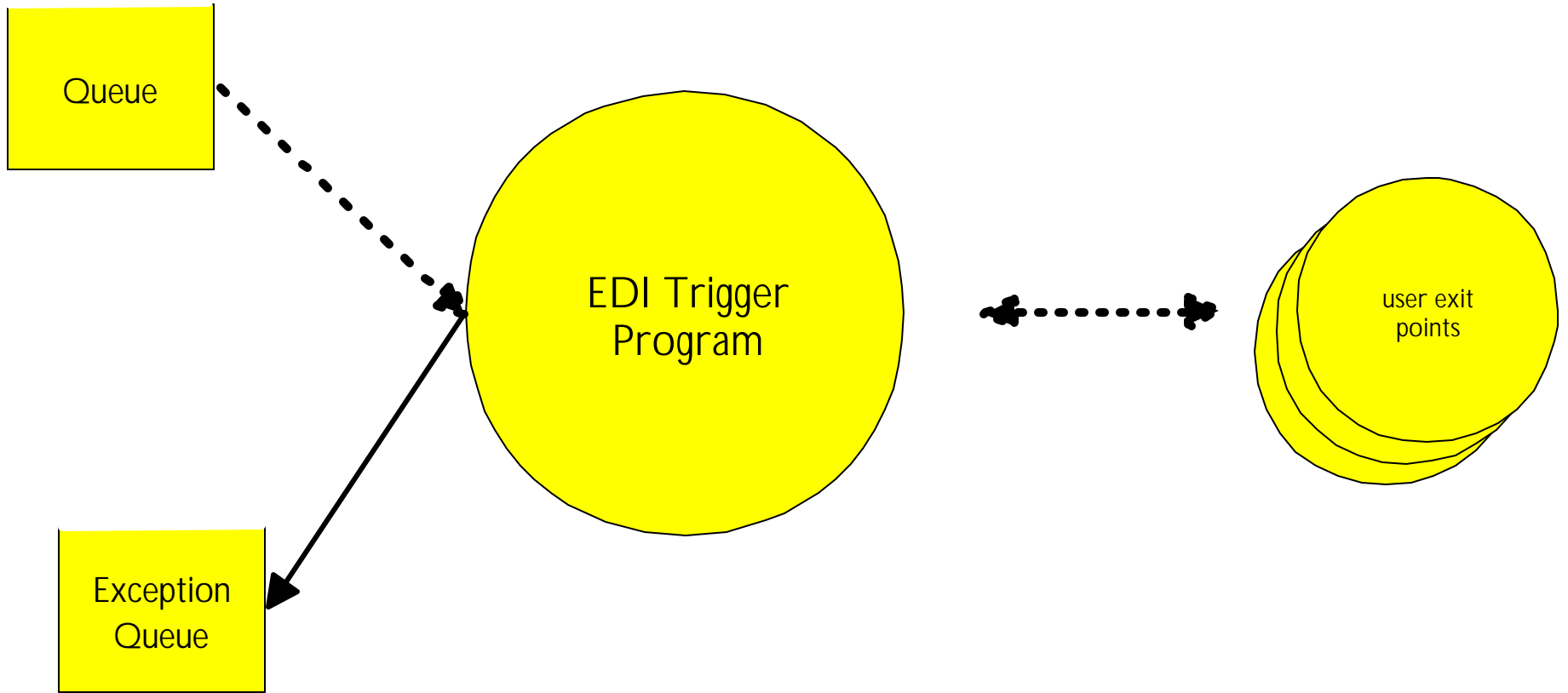
Message	a discrete communication between programs
MQTM	format of the message written to the initiation queue by the queue manager
Initiation Queue	a local queue on which the queue manager puts trigger messages
Trigger Monitor	a continuously-running application serving one or more initiation queues
Trigger Program	an application started by a trigger monitor



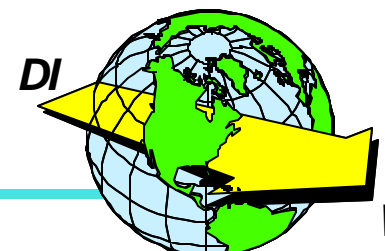
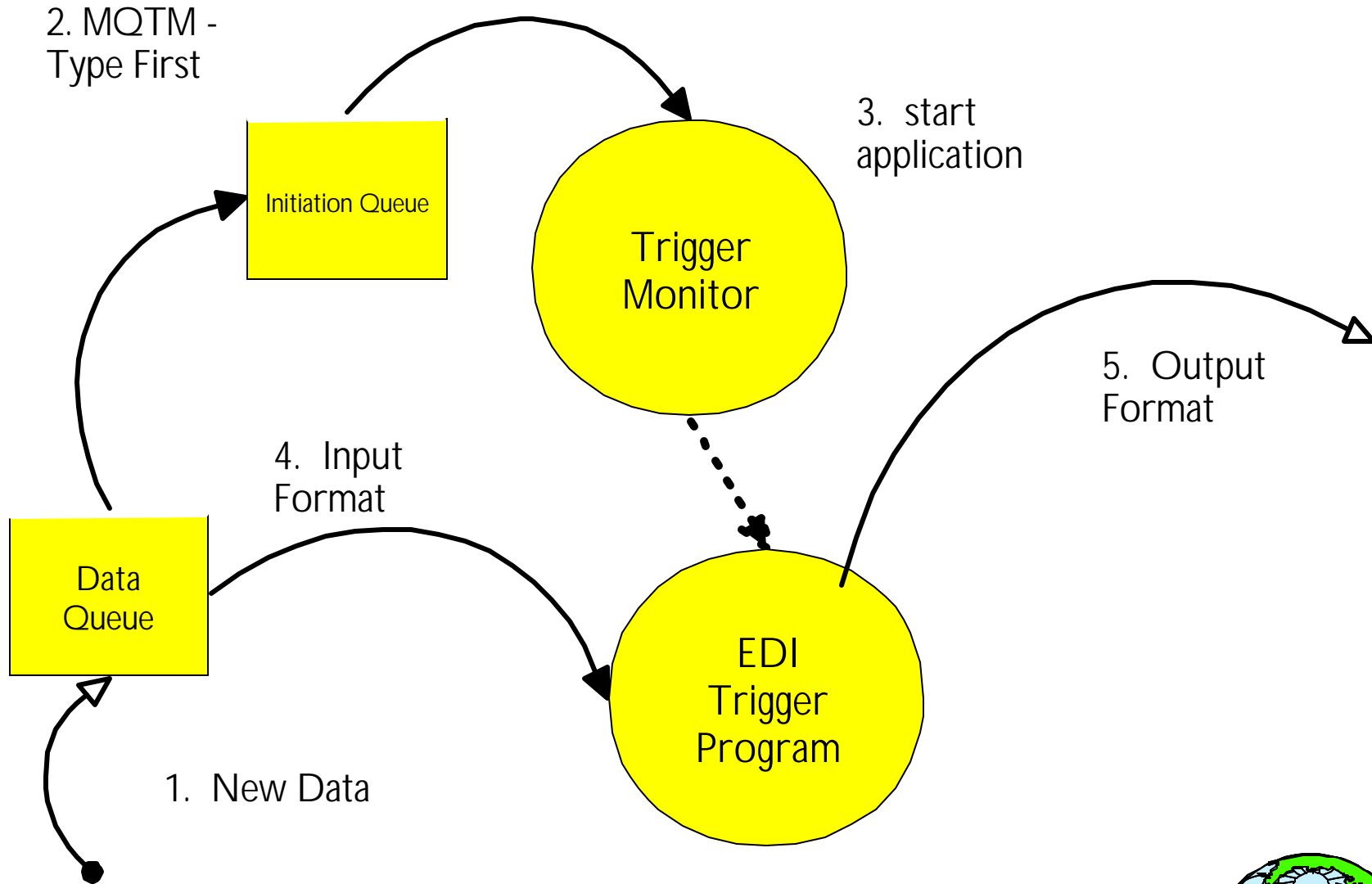
Triggering EDI transformation



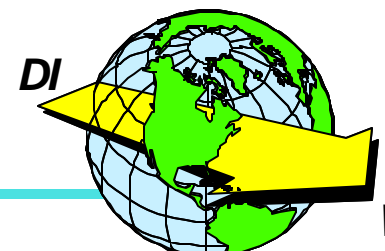
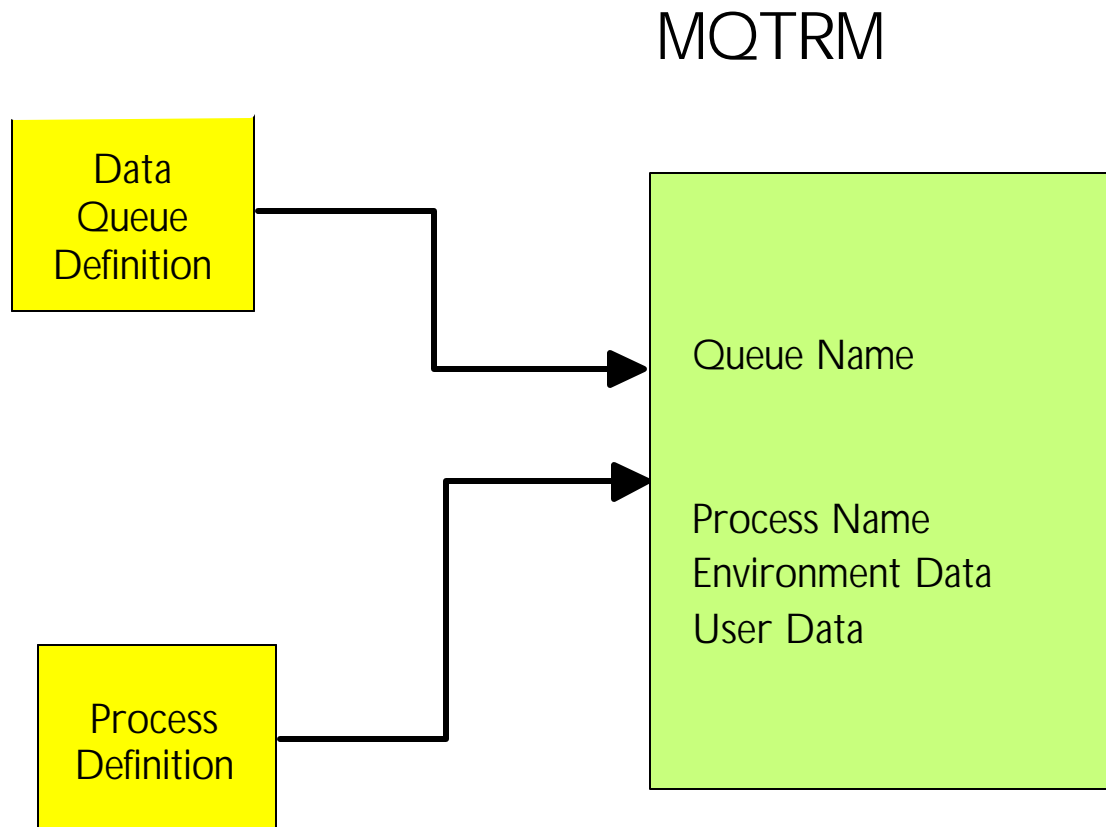
Triggering EDI Transformation



Triggering Sequence of Events



MQ configuration



Installation Queue Definitions

