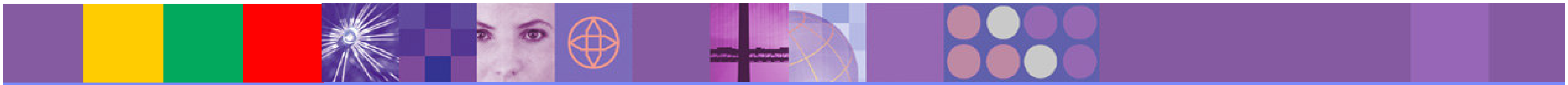
2004 WDI / WBIC Customer Conference

Global Business Transformation

WebSphere Data Interchange

WMQ Integration

David Shannon and Lee Whitaker



Welcome and Introductions

- David Shannon
  - Joined the WDI Team 2001
  - Started with IBM at the AIX Support Center
  - Introduced the WDIServer Program

Welcome and Introductions

- Lee Whitaker
  - Joined WebSphere Data Interchange in 2001
  - IBM certified WMQ Specialist

Critical Definitions

Relative Humidity

Palmetto Bug

Hurricane Season

Objectives

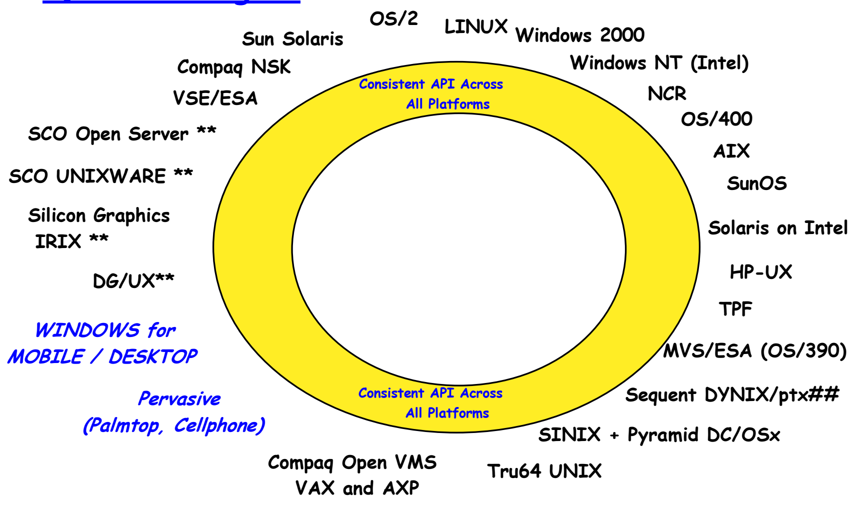
- Become familiar with the following topics
  - WMQ Setup
  - WMQ Related WDI Profiles
  - Using WDI to Read and Write to WMQ
  - Command Chaining
  - WDIAdapter and WDIServer Programs

MQSeries Setup

- Define WebSphere MQ
- Define WebSphere MQ Objects
- Demonstrate WebSphere MQ Tools

WebSphere MQ: A Platform Integrator

Queue Managers



Consistent API Across All Platforms

OS/2
LINUX
Windows 2000
Windows NT (Intel)
NCR
OS/400
AIX
SunOS
Solaris on Intel
HP-UX
TPF
MVS/ESA (OS/390)
Sequent DYNIX/ptx##
SINIX + Pyramid DC/OSx
Tru64 UNIX
Compaq Open VMS
VAX and AXP

Pervasive
(Palmtop, Cellphone)

WINDOWS for
MOBILE / DESKTOP

DG/UX**
Silicon Graphics
IRIX **
SCO UNIXWARE **
SCO Open Server **
VSE/ESA
Compaq NSK
Sun Solaris

Consistent API Across All Platforms

** Product of Willow Technology)
++ Product of Unisys
Product of Sequent
*** Product of CommerceQuest

Clients

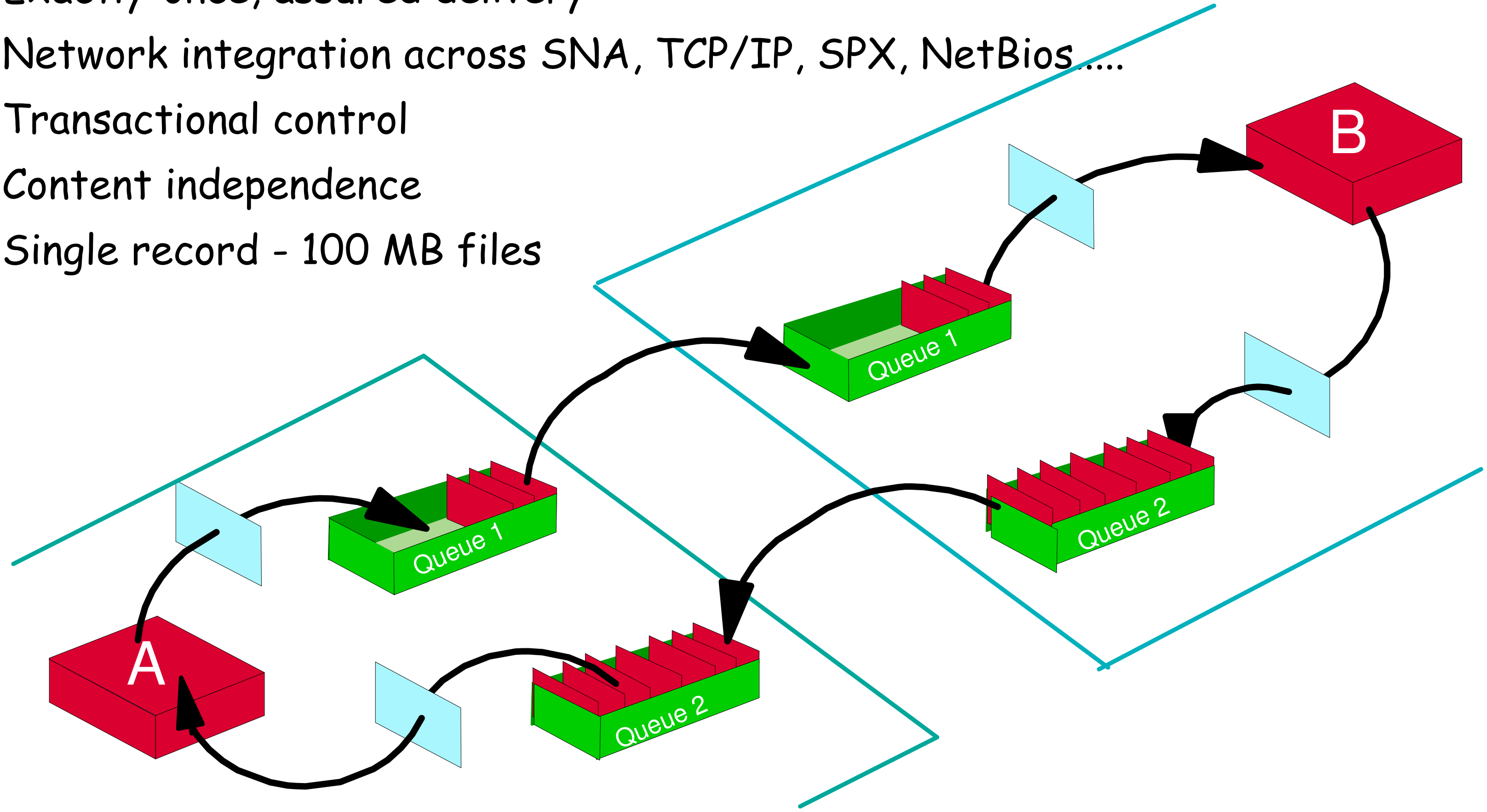
Tru64 UNIX
DEC OpenVMS
SUN OS
SUN Solaris
LINUX
SINIX/DC/OSx

Windows (Intel)

DOS
OS/2
TPF
Pyramid DC/SOx
HP/UX
NCR
SCO Unixware
SCO Open Server
AIX/6000
Sequent DYNIX/ptx
Java
Stratus VOS
Apple MacOS **
VM
IBM 4690 ***
HP 3000 MPE/iX
UNIXWARE
DG/UX for Intel **
Silicon Graphics IRIX **
Unisys A-Series ++

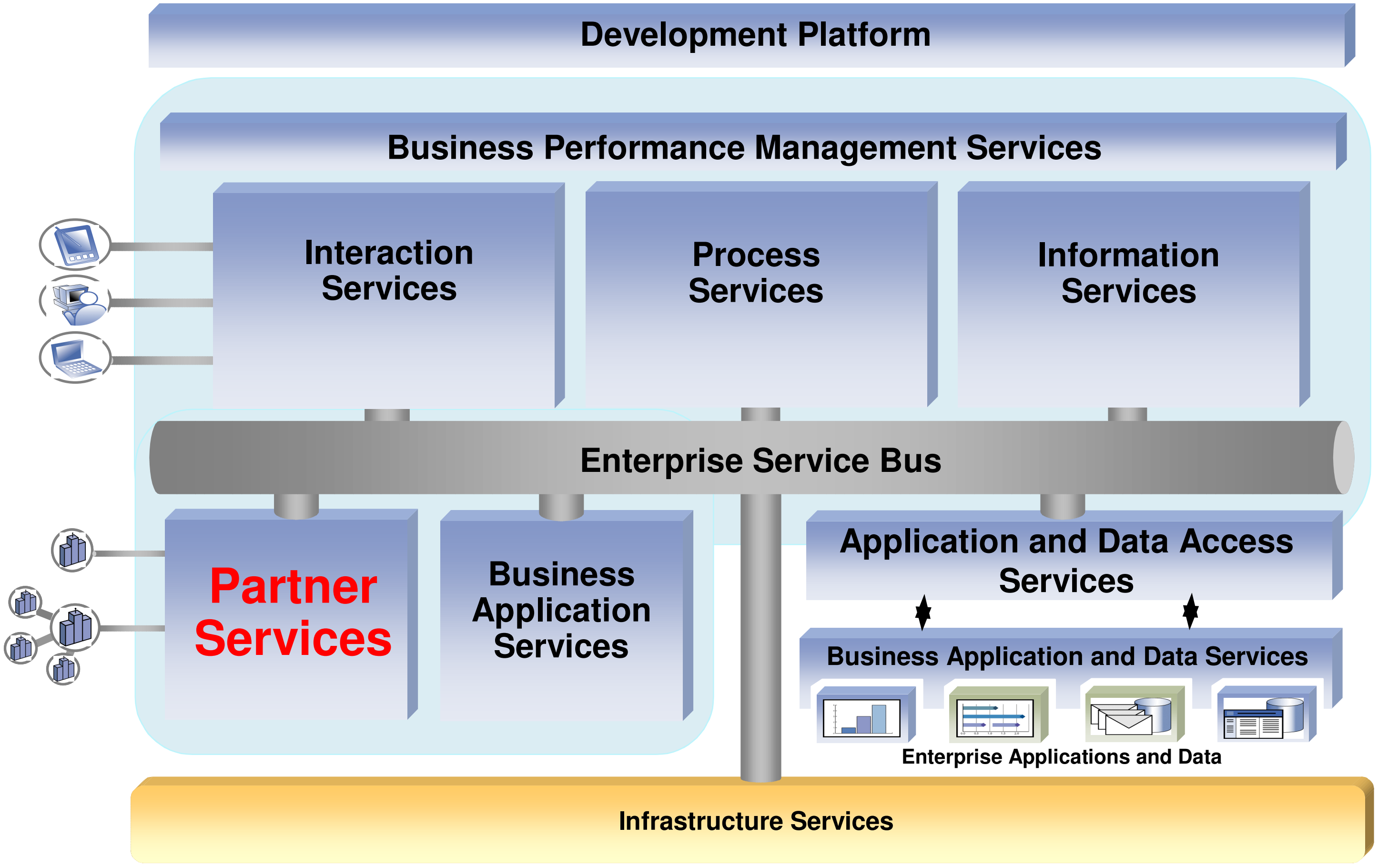
WebSphere MQ: An Application Integrator

- Loosely coupling between applications
- A single, multi-platform API (> 40 platforms)
- Exactly-once, assured delivery
- Network integration across SNA, TCP/IP, SPX, NetBios.....
- Transactional control
- Content independence
- Single record - 100 MB files



Business Integration Reference Architecture

IBM Software Offerings


Development Platform
Business Performance Management Services
Interaction Services
Process Services
Information Services
Enterprise Service Bus
Partner Services
Business Application Services
Application and Data Access Services
Business Application and Data Services
Enterprise Applications and Data
Infrastructure Services

WebSphere MQ Objects and Definitions

- Message
- Queue Manager
- Queues
  - Local Queue
  - Remote Queue
  - Transmission Queue
- Channel
- Process Definition
- Trigger Monitors

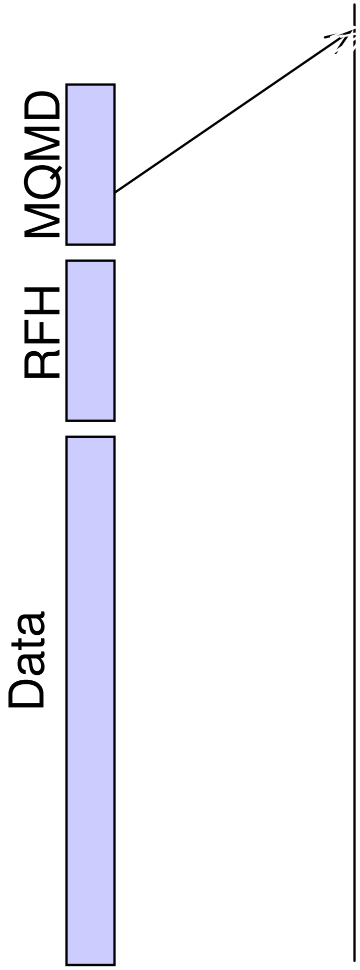
Message

- In message queuing applications, a communication sent between programs. (WMQ Programmer Guide)
- In EDIFACT: an ordered series of characters intended to convey information. [ISO 2382/16] UN/EDIFACT: a set of segments in the order specified in a message directory starting with the message header and ending with the message trailer. [ISO 9735] Equivalent to a transaction set. (http://www.unece.org/trade/untdid/texts/d300_d.htm)

Message: MQ

- A message descriptor (MQMD)
- Optional WMQ headers
  - RFH2 – the Rules and Format Header
- Optional data payload

Message Anatomy



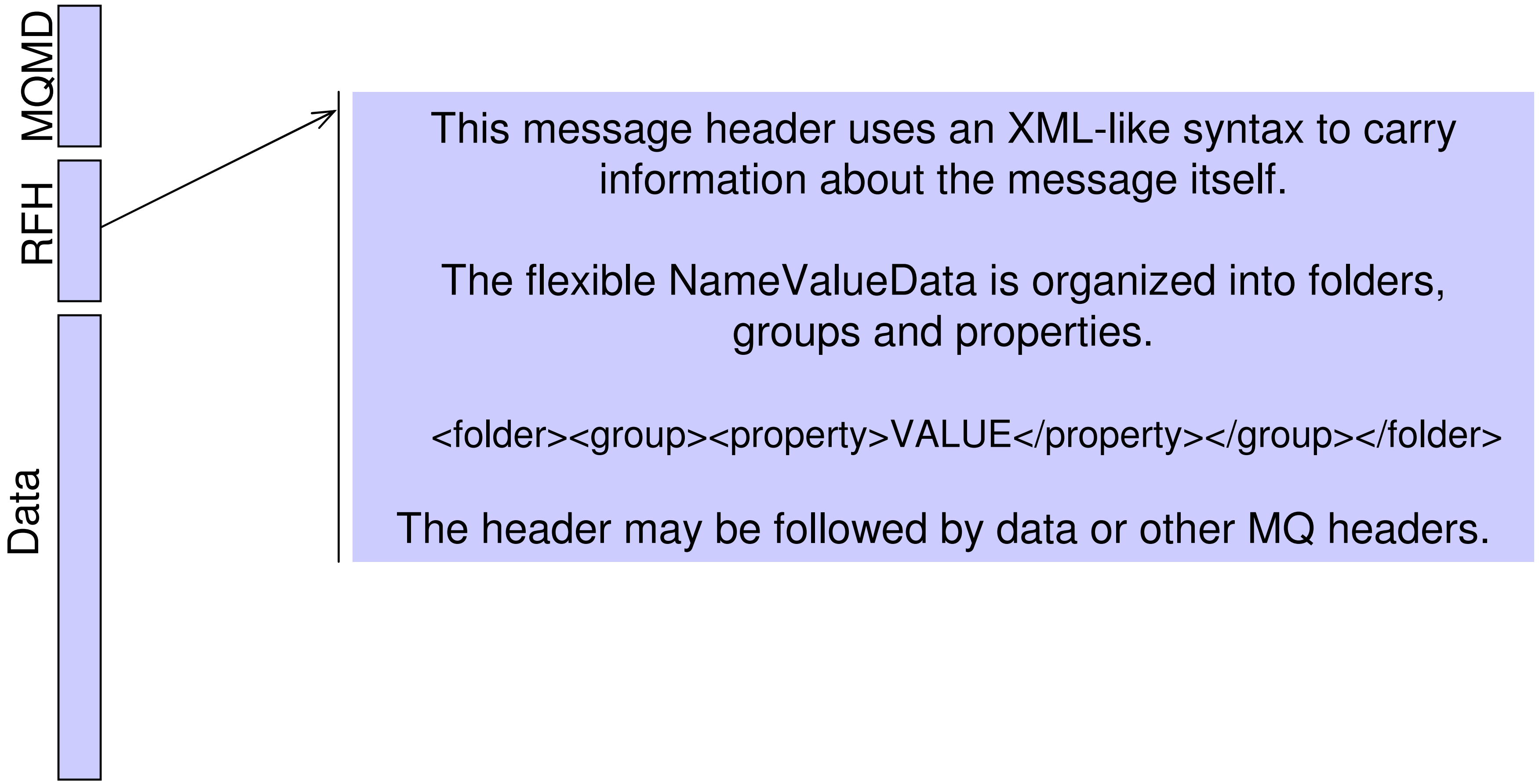
coded character set identifier (CCSID). The name of coded set of characters and their code point assignments.

in ../conv/tables/readme.ccs

037 – EBCDIC US English

437 – PC ASCII US

Message Anatomy



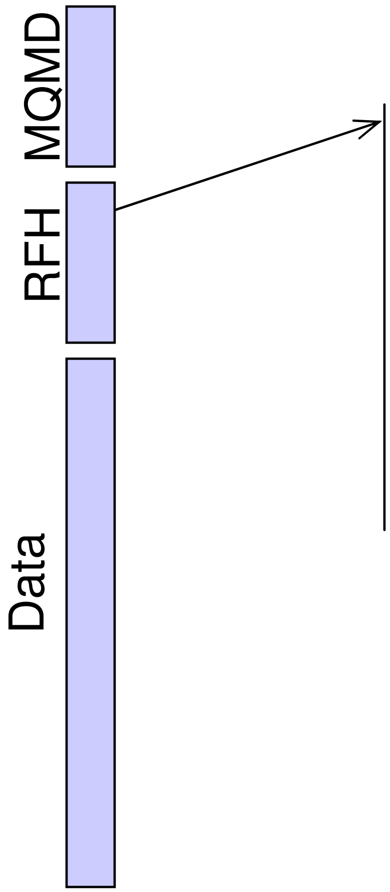
This message header uses an XML-like syntax to carry information about the message itself.

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.

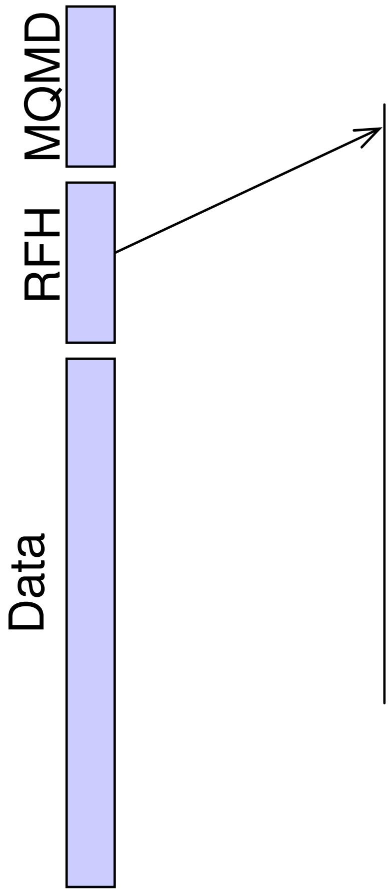
Message Anatomy



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

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

Message Anatomy



Use in WBI-MB

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

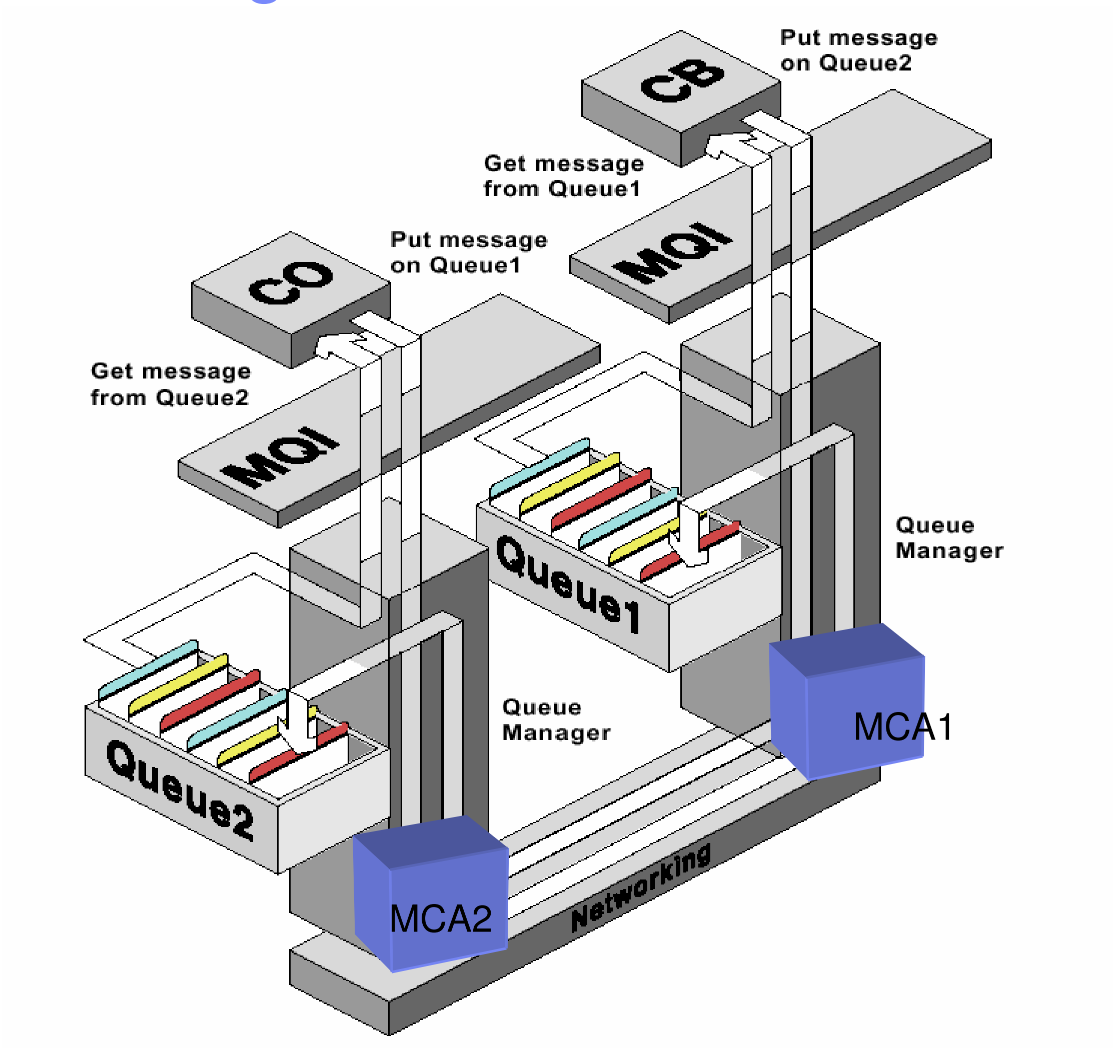
"Native" use in WDI

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

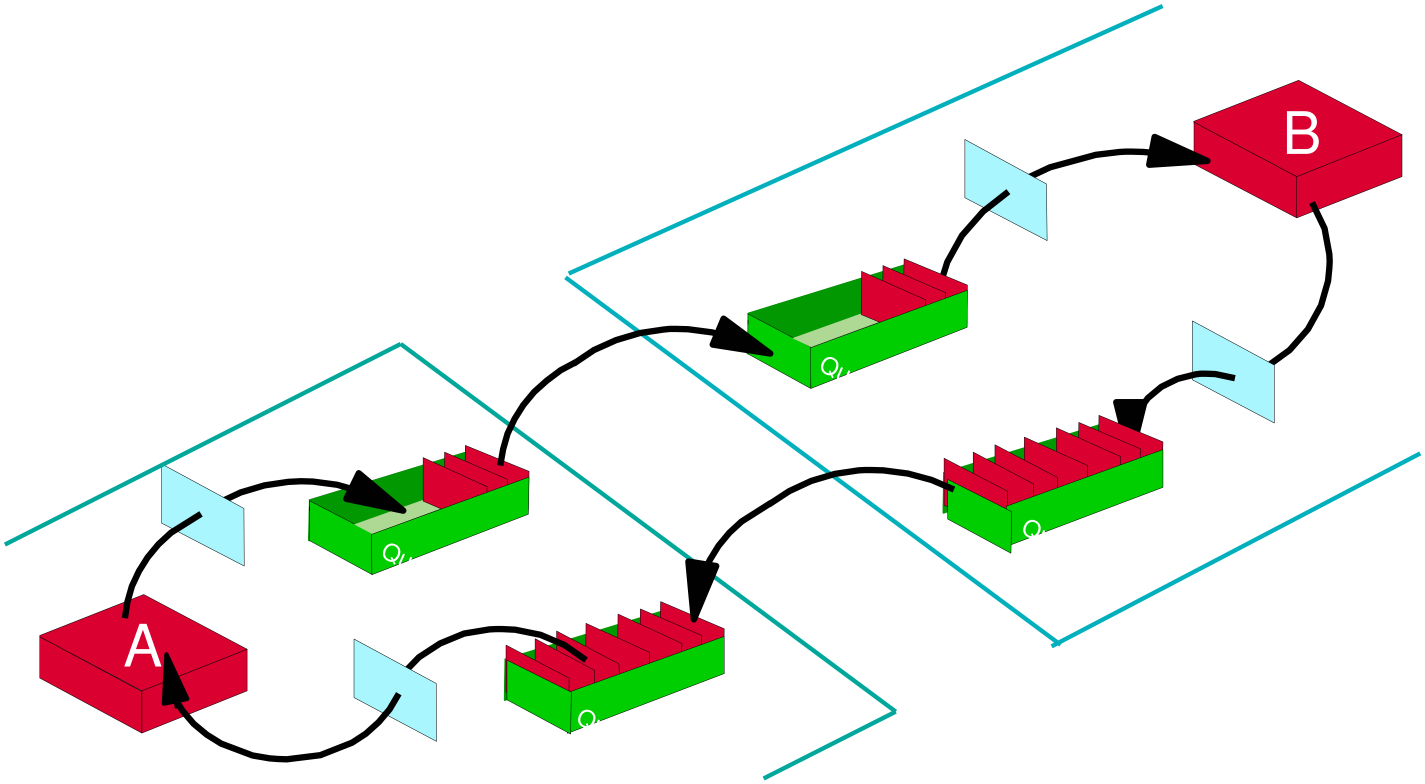
The Queue Manager

- Application Program Interface
  - Message Queuing Interface - MQI (C/C++)
  - JMS (Java)
- Logical Queues in Physical Storage
- Message Channel Agents
- Administrative Tools

The Queue Manager



What are Queues?



A Queue Defines a Place to Put a Message.

- Queues can be defined as local or remote.
  - Local queues are defined on a local queue manager. Programs can put messages onto or get messages off of a local queue.
  - Remote queues point to queues not on the local queue manager. Programs can only put messages onto a remote queue.

Transmission Queues

- Transmission queues are used by channels
  - Can store messages when a remote connection is not available.
  - Must be defined prior to creating a channel.
  - Can be used to trigger the start of a channel

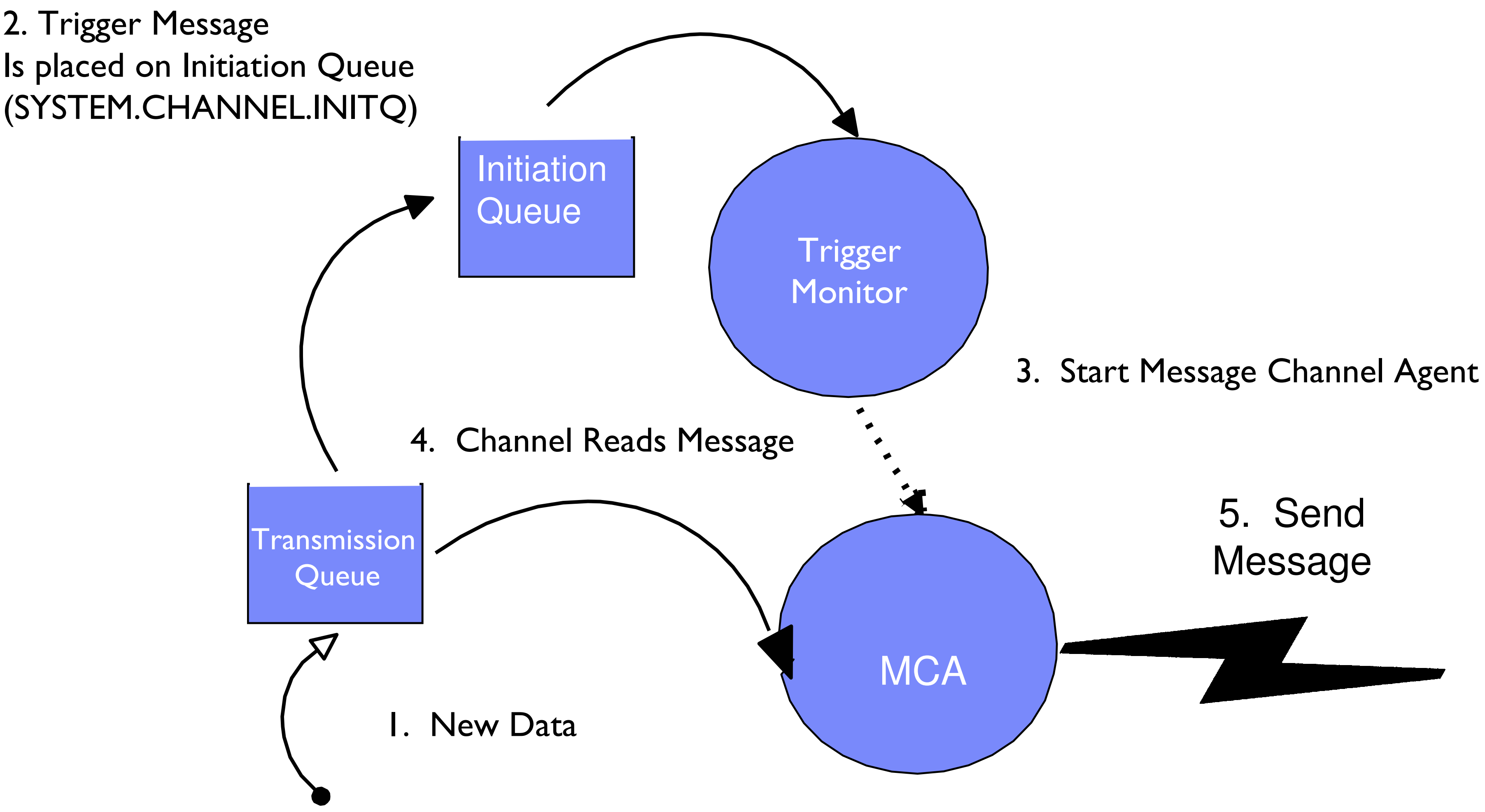
Defining a Channel

- **Channel:** a mechanism for moving messages from one queue manager to another.
  - Receiver Message Channel Agent
  - Sender Message Channel Agent
  - Communication Link (TCP/IP, SNA, etc.)
  - Transmission Queue

Channels and Triggering

- **Triggering**
  - A way to cause the queue manager to start an application automatically when predetermined conditions on a queue are satisfied.
- **Trigger Event**
  - An event (such as a message arriving on a queue) that causes a queue manager to create a trigger message on an initiation queue.
- **Trigger Message**
  - A message containing information about the program that a trigger monitor is to start.

Channel: Triggering the Message Channel Agent (MCA)



Process definition

- A WebSphere MQ object that contains the definition of a WebSphere MQ application.
  - Application Name
  - Arguments for that application

Trigger Monitor

- Application designed to monitor a local queue for trigger messages and take some type of action based upon that trigger message

Initiation Queue

- The local queue that a trigger monitor will monitor. This is where trigger messages will be placed when the queue monitor detects a trigger event on a queue.

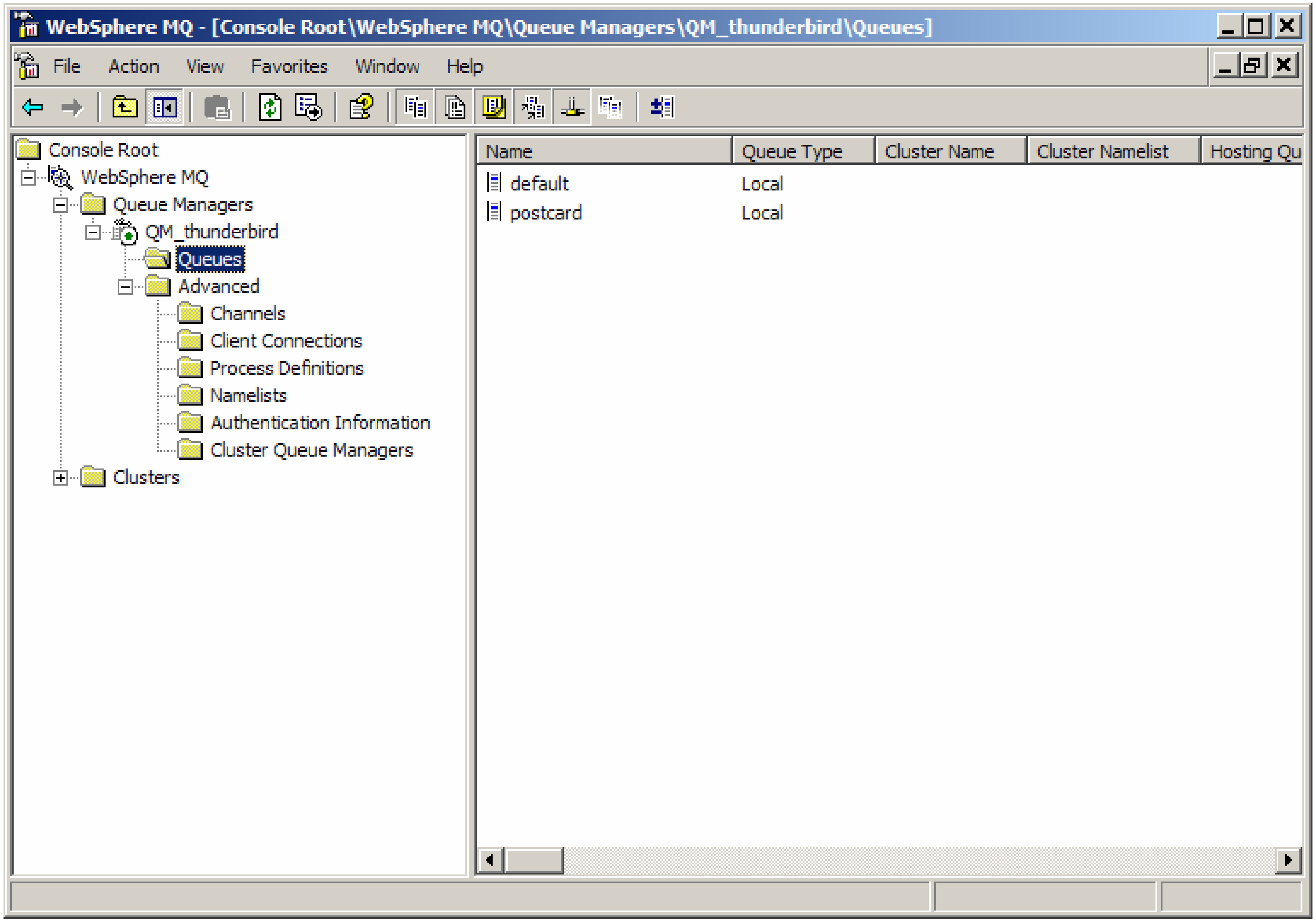
WebSphere MQ Administrative Tools

- WebSphere MQ Explorer
- WebSphere MQ Command Line
  - *runmqsc* for Open Systems
  - *csqutil* for z/OS
- *z/OS Administration*

WebSphere MQ Explorer

- Provides graphical interface to queue manager
- Can administer local or remote queue managers
- Manages WMQ Objects
  - Queues
  - Channels
  - Process Definitions
  - Client Connections
  - Name Lists
  - Authentication Information
  - Cluster Queue Managers

WebSphere MQ Explorer Main Window

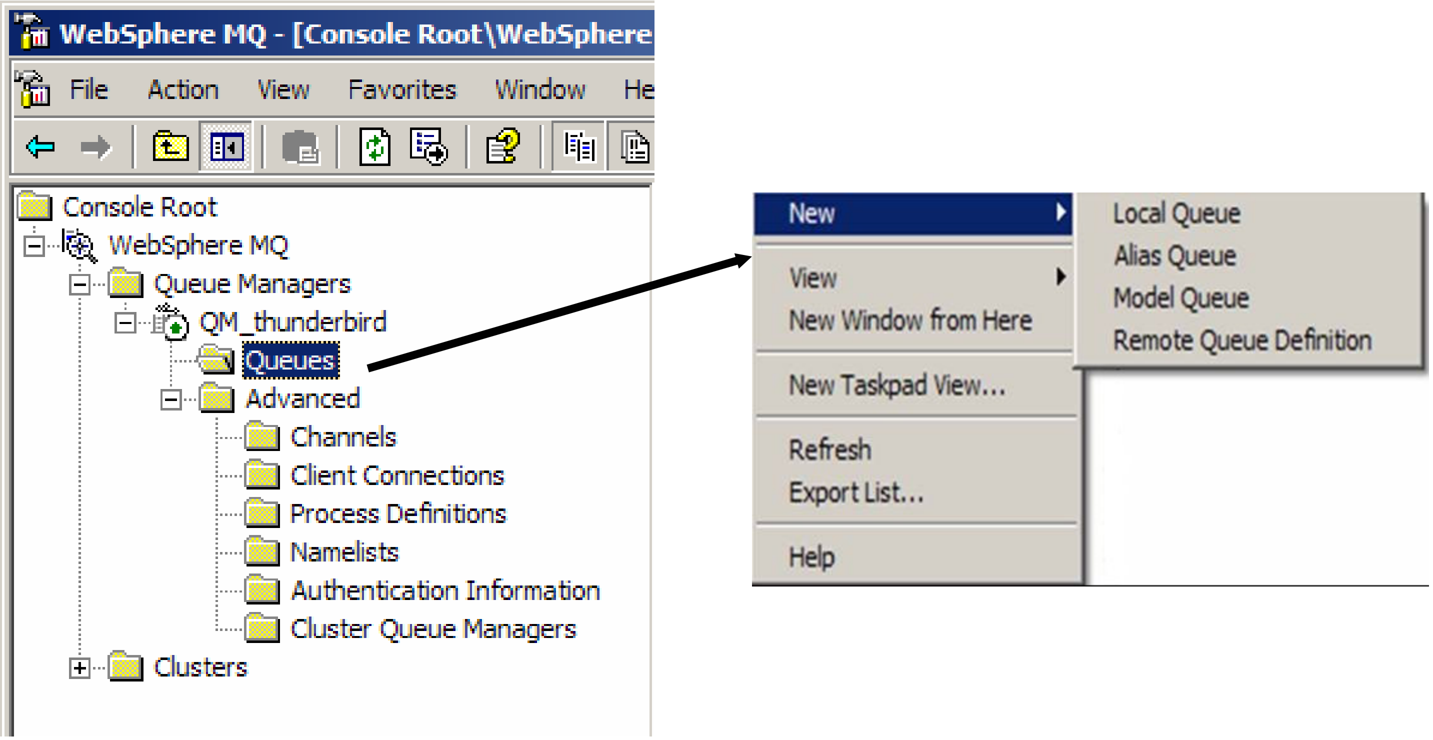


WebSphere MQ - [Console Root\WebSphere MQ\Queue Managers\QM_thunderbird\Queues]

- Console Root
  - WebSphere MQ
    - Queue Managers
      - QM_thunderbird
        - Queues
        - Advanced
          - Channels
          - Client Connections
          - Process Definitions
          - Namelists
          - Authentication Information
          - Cluster Queue Managers
    - Clusters

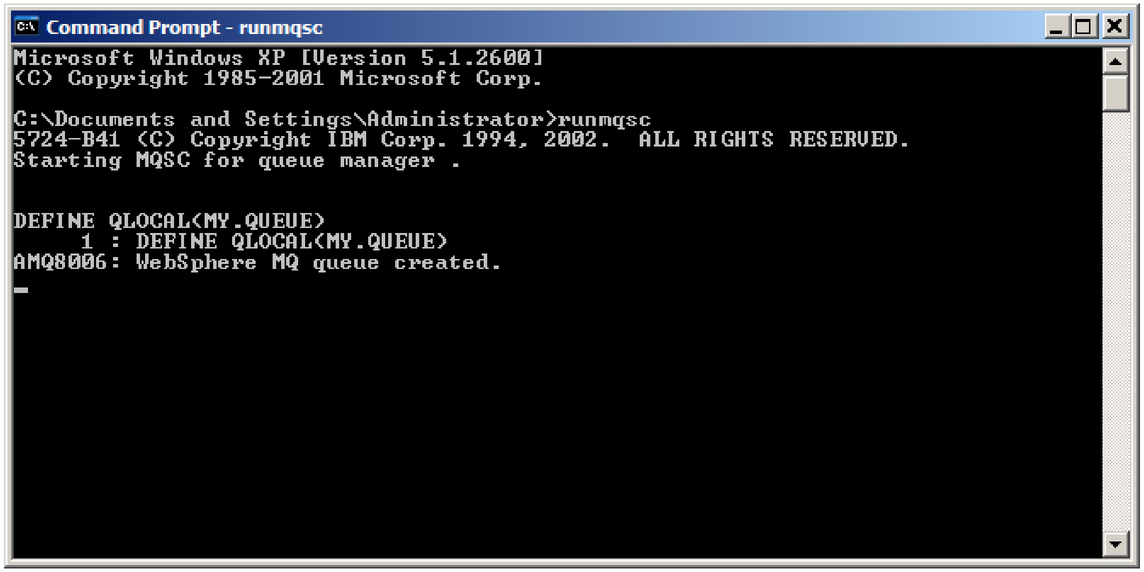
Name	Queue Type	Cluster Name	Cluster Namelist	Hosting Qu
default	Local			
postcard	Local			

WebSphere MQ Explorer Navigation Panel



WebSphere MQ Command Line

- Command line interface is provided by the runmqsc command.
  - ➢Can be used interactively
  - ➢Can be used in batch mode



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>runmqsc
5724-B41 (C) Copyright IBM Corp. 1994, 2002.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager .


DEFINE QLOCAL(MY.QUEUE)
     1 : DEFINE QLOCAL(MY.QUEUE)
AMQ8006: WebSphere MQ queue created.
_
```

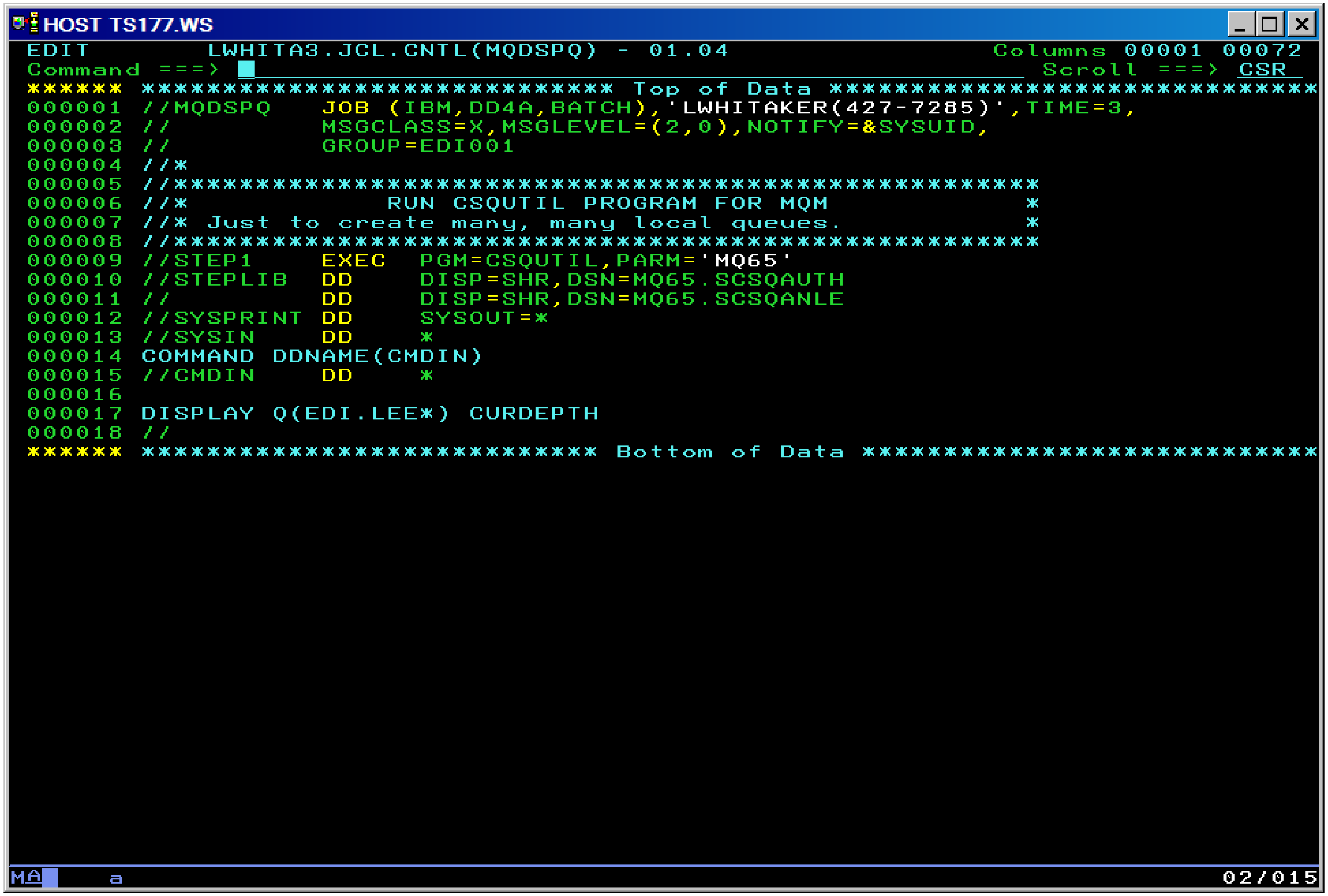
WebSphere Data Interchange Sample Script

```
********************************************************************/
* Program name: AdvAdapterMQ.txt                                   */
*                                                                  */
* Description: builds MQ queues for the WDI Default Installation   */
*                                                                  */
* Use:  Use this file as standard input for  'runmqsc'.            */
*                                                                  */
*            runmqsc QMgr < AdvAdapterMQ.txt                       */
*                                                                  */
* If the default queue manager is used the QMgr parameter is not   */
* needed.                                                          */
* To change the home directory for the WDIAdapter, alter the       */
*     USERDATA in the PROCESS definition.                          */
*                                                                  */
********************************************************************/
DEFINE PROCESS('WDI.TRANSLATOR.PROC') REPLACE +
        DESCR('WebSphere Data Interchange Adapter') +
        APPLTYPE(DEF)   +
        ENVRDATA('NumThreads(1) Timeout(10000)')

********************************************************************/
*   Create the input and output queues as local.                   */
*                                                                  */
*   Input queues will cause a trigger message.                     */
********************************************************************/
DEFINE QLOCAL('WDIAdapterCmd') REPLACE +
                    DESCR('Command queue WDIServer') +
                    MAXMSGL(4194304) +
                    DEFPSIST(YES)  +
                    BOTHRESH(0) +
                    SHARE +
                    GET(ENABLED) +
                    PUT(ENABLED)


DEFINE QLOCAL('WDI.PRTFILE.Q') REPLACE +
                    DESCR('Printfile data from WDIServer program') +
                    MAXMSGL(4194304) +
                    DEFPSIST(YES)  +
                    BOTHRESH(0) +
                    SHARE +
                    GET(ENABLED) +
                    PUT(ENABLED)
```

WebSphere MQ on z/OS

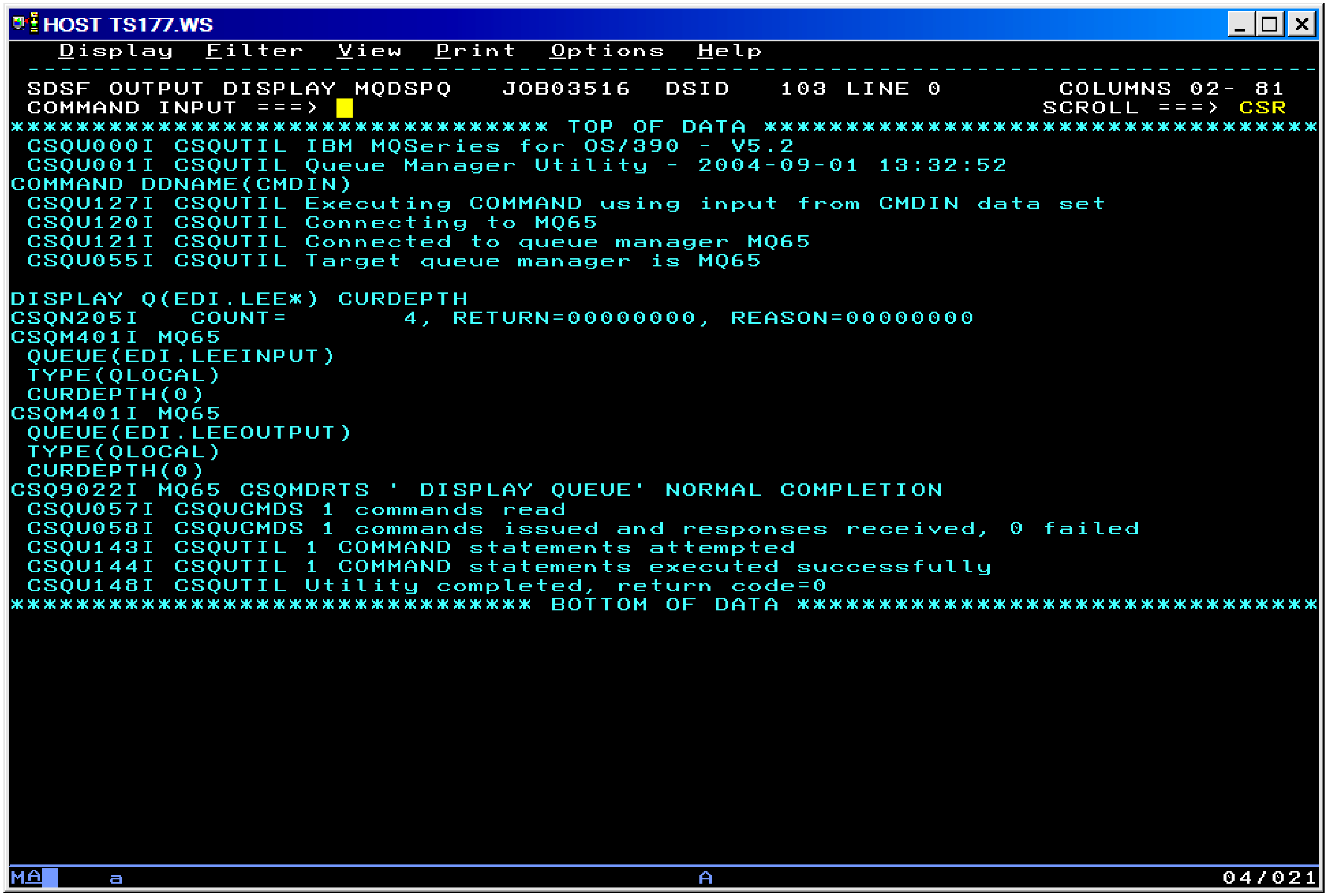
- Support for MQSC
- Operations Console

CSQUTIL with MQSC



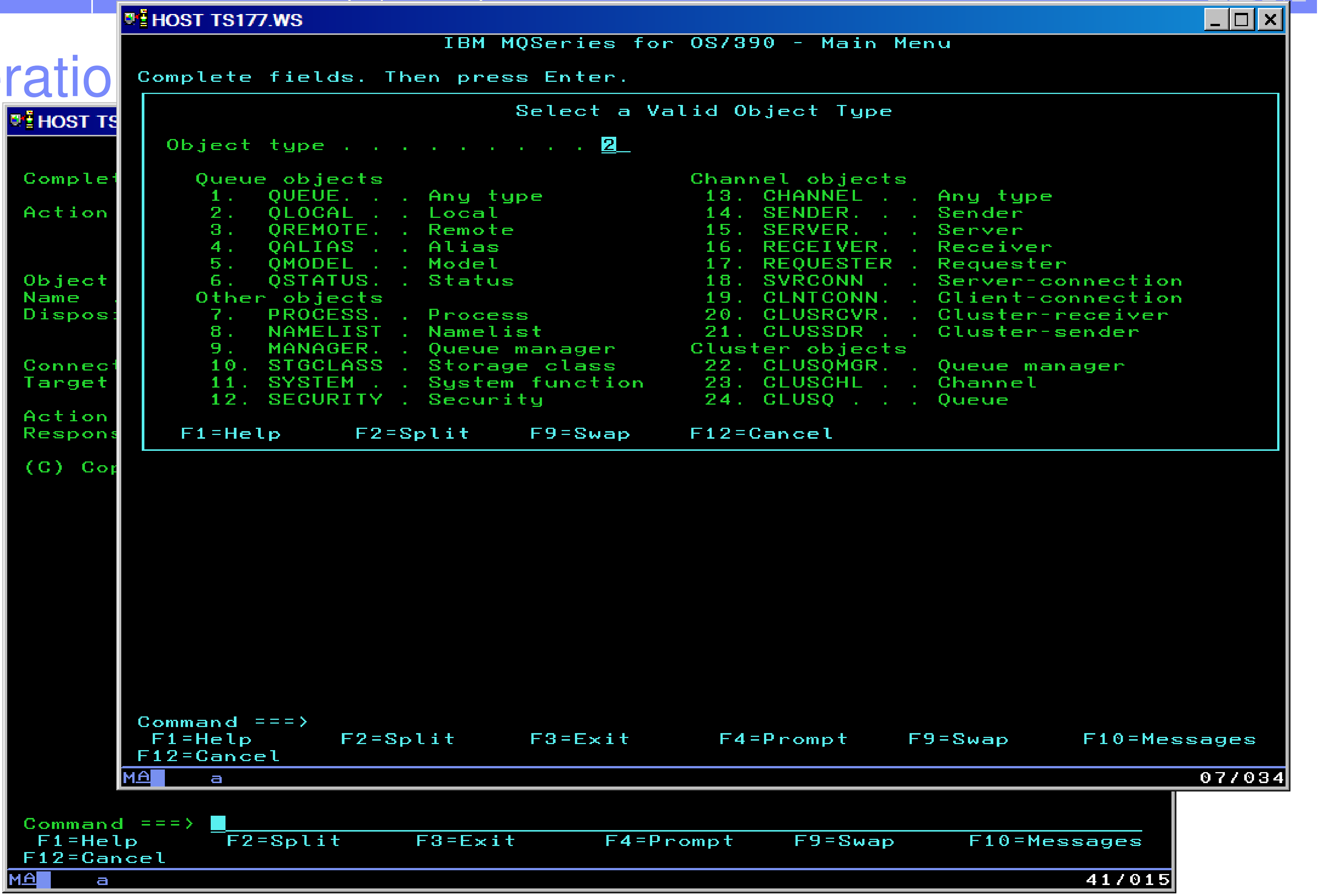
```
EDIT       LWHITA3.JCL.CNTL(MQDSPQ) - 01.04                Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
//MQDSPQ   JOB (IBM,DD4A,BATCH),'LWHITAKER(427-7285)',TIME=3,
//         MSGCLASS=X,MSGLEVEL=(2,0),NOTIFY=&SYSUID,
//         GROUP=EDI001
//*
//******************************************************
//*            RUN CSQUTIL PROGRAM FOR MQM             *
//* Just to create many, many local queues.            *
//******************************************************
//STEP1    EXEC  PGM=CSQUTIL,PARM='MQ65'
//STEPLIB  DD    DISP=SHR,DSN=MQ65.SCSQAUTH
//         DD    DISP=SHR,DSN=MQ65.SCSQANLE
//SYSPRINT DD    SYSOUT=*
//SYSIN    DD     *
COMMAND DDNAME(CMDIN)
//CMDIN    DD     *

DISPLAY Q(EDI.LEE*) CURDEPTH
//
****** **************************** Bottom of Data ****************************
```

CSQUTIL with MQSC



```
 SDSF OUTPUT DISPLAY MQDSPQ    JOB03516   DSID    103 LINE 0        COLUMNS 02- 81
 COMMAND INPUT ===>                                                SCROLL ===> CSR
********************************* TOP OF DATA **********************************
 CSQU000I CSQUTIL IBM MQSeries for OS/390 - V5.2
 CSQU001I CSQUTIL Queue Manager Utility - 2004-09-01 13:32:52
COMMAND DDNAME(CMDIN)
 CSQU127I CSQUTIL Executing COMMAND using input from CMDIN data set
 CSQU120I CSQUTIL Connecting to MQ65
 CSQU121I CSQUTIL Connected to queue manager MQ65
 CSQU055I CSQUTIL Target queue manager is MQ65

DISPLAY Q(EDI.LEE*) CURDEPTH
CSQN205I   COUNT=       4, RETURN=00000000, REASON=00000000
CSQM401I MQ65
 QUEUE(EDI.LEEINPUT)
 TYPE(QLOCAL)
 CURDEPTH(0)
CSQM401I MQ65
 QUEUE(EDI.LEEOUTPUT)
 TYPE(QLOCAL)
 CURDEPTH(0)
CSQ9022I MQ65 CSQMDRTS ' DISPLAY QUEUE' NORMAL COMPLETION
 CSQU057I CSQUCMDS 1 commands read
 CSQU058I CSQUCMDS 1 commands issued and responses received, 0 failed
 CSQU143I CSQUTIL 1 COMMAND statements attempted
 CSQU144I CSQUTIL 1 COMMAND statements executed successfully
 CSQU148I CSQUTIL Utility completed, return code=0
******************************** BOTTOM OF DATA ********************************
```

Operatio



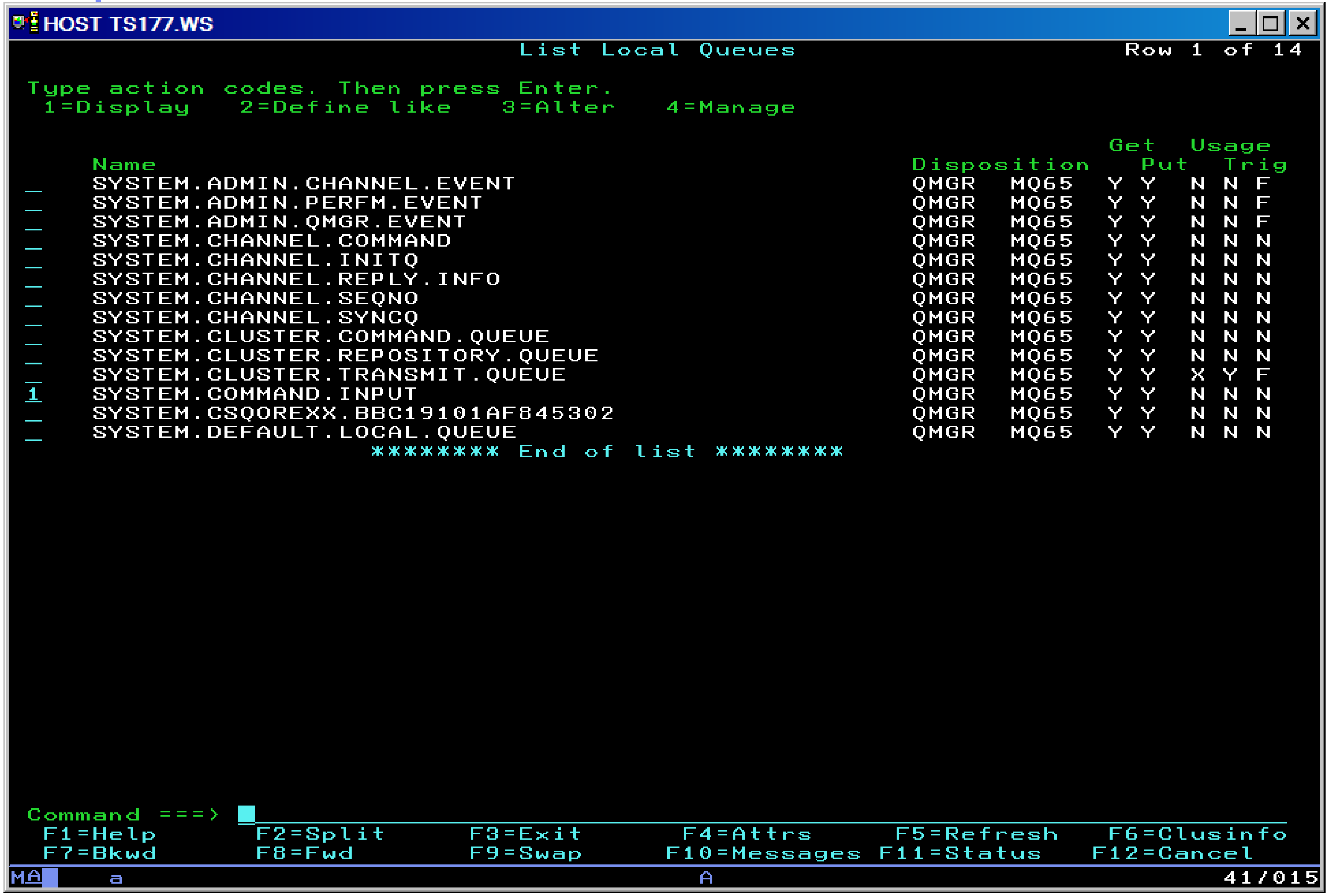
HOST TS177.WS

```
                    IBM MQSeries for OS/390 - Main Menu

 Complete fields. Then press Enter.
 ┌──────────────────────────────────────────────────────────────────────────┐
 │                        Select a Valid Object Type                        │
 │                                                                          │
 │ Object type . . . . . . . . . 2_                                         │
 │                                                                          │
 │   Queue objects                      Channel objects                     │
 │    1.  QUEUE. . . Any type            13. CHANNEL . . Any type           │
 │    2.  QLOCAL . . Local               14. SENDER. . . Sender             │
 │    3.  QREMOTE. . Remote              15. SERVER. . . Server             │
 │    4.  QALIAS . . Alias               16. RECEIVER. . Receiver           │
 │    5.  QMODEL . . Model               17. REQUESTER . Requester          │
 │    6.  QSTATUS. . Status              18. SVRCONN . . Server-connection  │
 │   Other objects                       19. CLNTCONN. . Client-connection  │
 │    7.  PROCESS. . Process             20. CLUSRCVR. . Cluster-receiver   │
 │    8.  NAMELIST . Namelist            21. CLUSSDR . . Cluster-sender     │
 │    9.  MANAGER. . Queue manager      Cluster objects                     │
 │    10. STGCLASS . Storage class       22. CLUSQMGR. . Queue manager      │
 │    11. SYSTEM . . System function     23. CLUSCHL . . Channel            │
 │    12. SECURITY . Security            24. CLUSQ . . . Queue              │
 │                                                                          │
 │  F1=Help     F2=Split     F9=Swap     F12=Cancel                         │
 └──────────────────────────────────────────────────────────────────────────┘









 Command ===>
  F1=Help      F2=Split     F3=Exit      F4=Prompt    F9=Swap      F10=Messages
 F12=Cancel
MA        a                                                              07/034
```

HOST TS

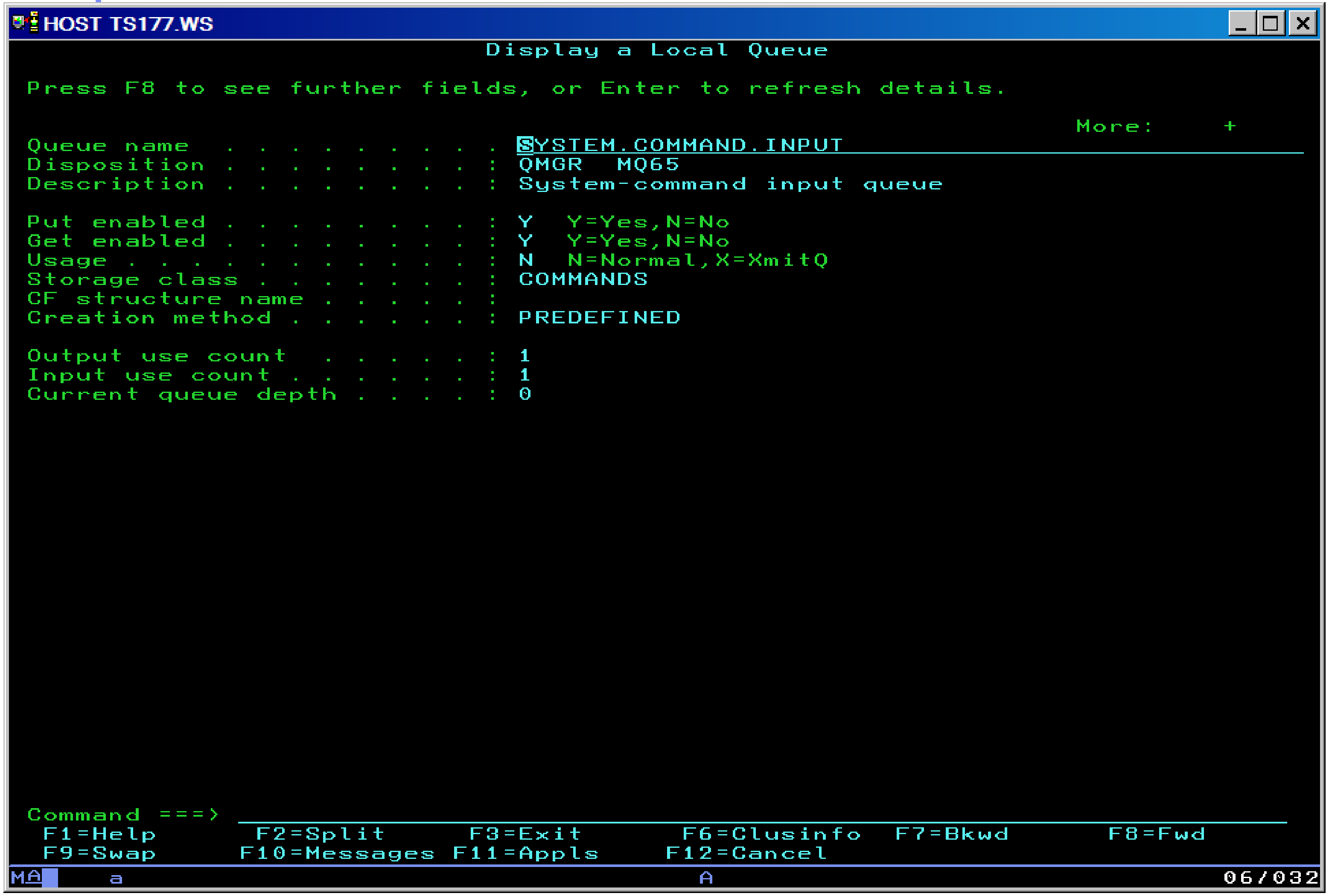
```
 Complet

 Action



 Object
 Name   .
 Dispos


 Connect
 Target

 Action
 Respons

 (C) Cop




 Command ===> _______________________________________________________________
  F1=Help      F2=Split     F3=Exit      F4=Prompt    F9=Swap      F10=Messages
 F12=Cancel
MA        a                                                              41/015
```

z/OS Operations Panel



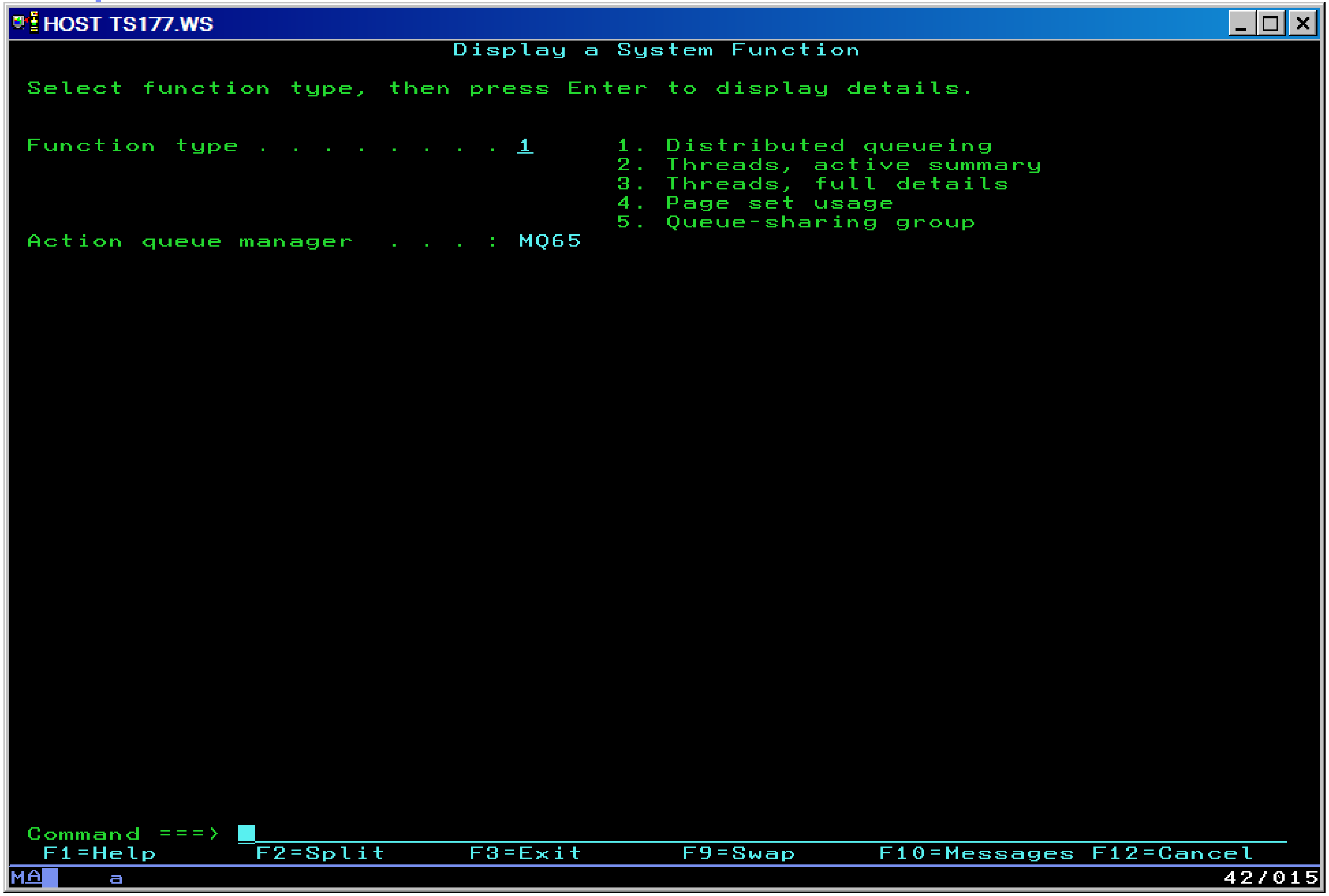
```
                               List Local Queues                    Row 1 of 14

 Type action codes. Then press Enter.
  1=Display   2=Define like   3=Alter   4=Manage

                                                                   Get  Usage
     Name                                              Disposition   Put  Trig
 _   SYSTEM.ADMIN.CHANNEL.EVENT                        QMGR  MQ65  Y Y  N N F
 _   SYSTEM.ADMIN.PERFM.EVENT                          QMGR  MQ65  Y Y  N N F
 _   SYSTEM.ADMIN.QMGR.EVENT                           QMGR  MQ65  Y Y  N N F
 _   SYSTEM.CHANNEL.COMMAND                            QMGR  MQ65  Y Y  N N N
 _   SYSTEM.CHANNEL.INITQ                              QMGR  MQ65  Y Y  N N N
 _   SYSTEM.CHANNEL.REPLY.INFO                         QMGR  MQ65  Y Y  N N N
 _   SYSTEM.CHANNEL.SEQNO                              QMGR  MQ65  Y Y  N N N
 _   SYSTEM.CHANNEL.SYNCQ                              QMGR  MQ65  Y Y  N N N
 _   SYSTEM.CLUSTER.COMMAND.QUEUE                      QMGR  MQ65  Y Y  N N N
 _   SYSTEM.CLUSTER.REPOSITORY.QUEUE                   QMGR  MQ65  Y Y  N N N
 _   SYSTEM.CLUSTER.TRANSMIT.QUEUE                     QMGR  MQ65  Y Y  X Y F
 1   SYSTEM.COMMAND.INPUT                              QMGR  MQ65  Y Y  N N N
 _   SYSTEM.CSQOREXX.BBC19101AF845302                  QMGR  MQ65  Y Y  N N N
 _   SYSTEM.DEFAULT.LOCAL.QUEUE                        QMGR  MQ65  Y Y  N N N
                      ******** End of list ********









 Command ===>
  F1=Help      F2=Split     F3=Exit      F4=Attrs     F5=Refresh   F6=Clusinfo
  F7=Bkwd      F8=Fwd       F9=Swap     F10=Messages F11=Status   F12=Cancel
```

z/OS Operations Panel



```
                         Display a Local Queue

 Press F8 to see further fields, or Enter to refresh details.

                                                                 More:     +
 Queue name . . . . . . . . . : SYSTEM.COMMAND.INPUT
 Disposition . . . . . . . . : QMGR   MQ65
 Description . . . . . . . . : System-command input queue

 Put enabled . . . . . . . . : Y   Y=Yes,N=No
 Get enabled . . . . . . . . : Y   Y=Yes,N=No
 Usage . . . . . . . . . . . : N   N=Normal,X=XmitQ
 Storage class . . . . . . . : COMMANDS
 CF structure name . . . . . :
 Creation method . . . . . . : PREDEFINED

 Output use count  . . . . . : 1
 Input use count . . . . . . : 1
 Current queue depth . . . . : 0










 Command ===> ________________________________________________________________
  F1=Help      F2=Split     F3=Exit      F6=Clusinfo  F7=Bkwd      F8=Fwd
  F9=Swap     F10=Messages F11=Appls    F12=Cancel
MA     a                                    A                             06/032
```

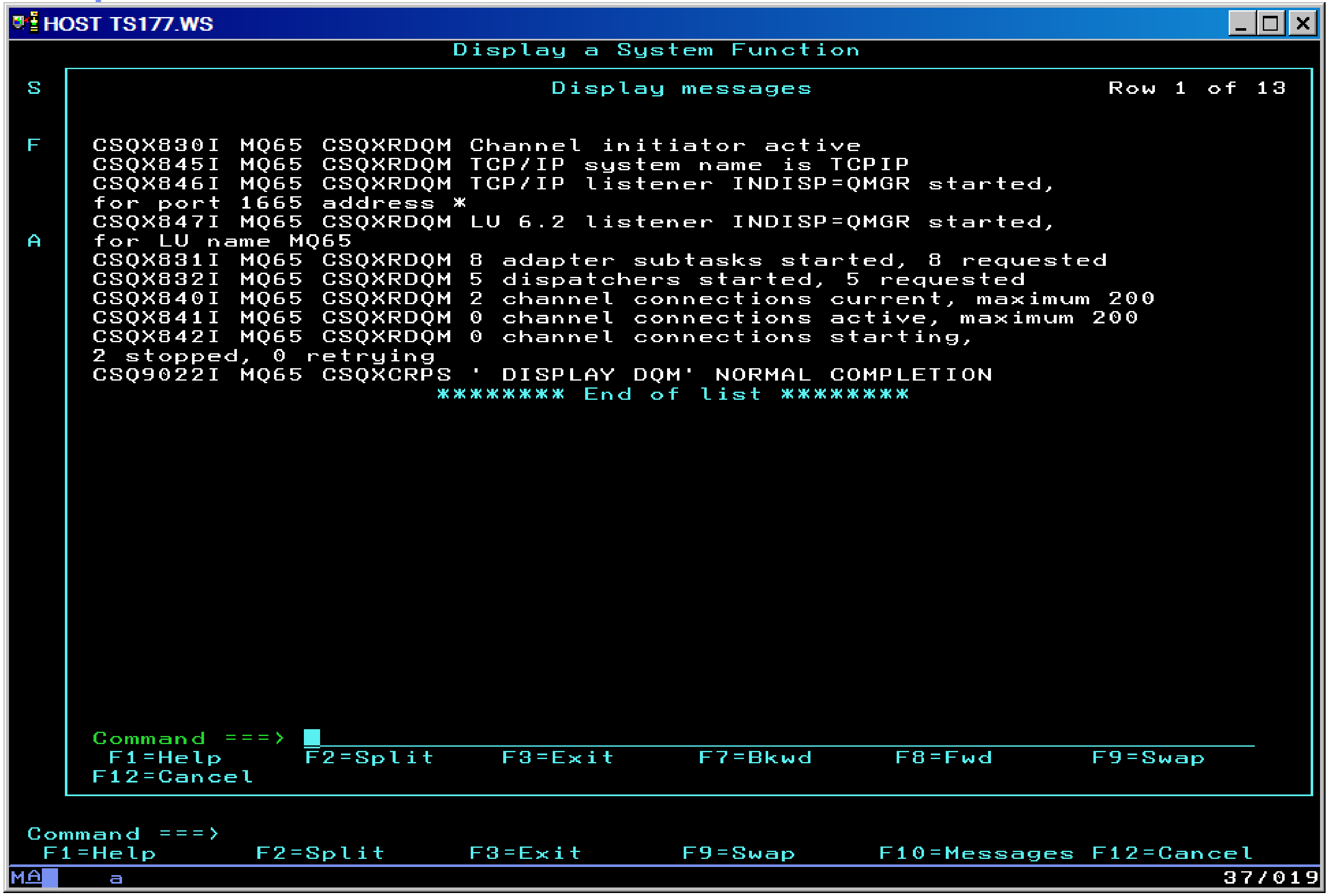
z/OS Operations Panel



```
                           Display a System Function

Select function type, then press Enter to display details.

Function type . . . . . . . . . 1      1. Distributed queueing
                                       2. Threads, active summary
                                       3. Threads, full details
                                       4. Page set usage
                                       5. Queue-sharing group
Action queue manager . . . . : MQ65
```

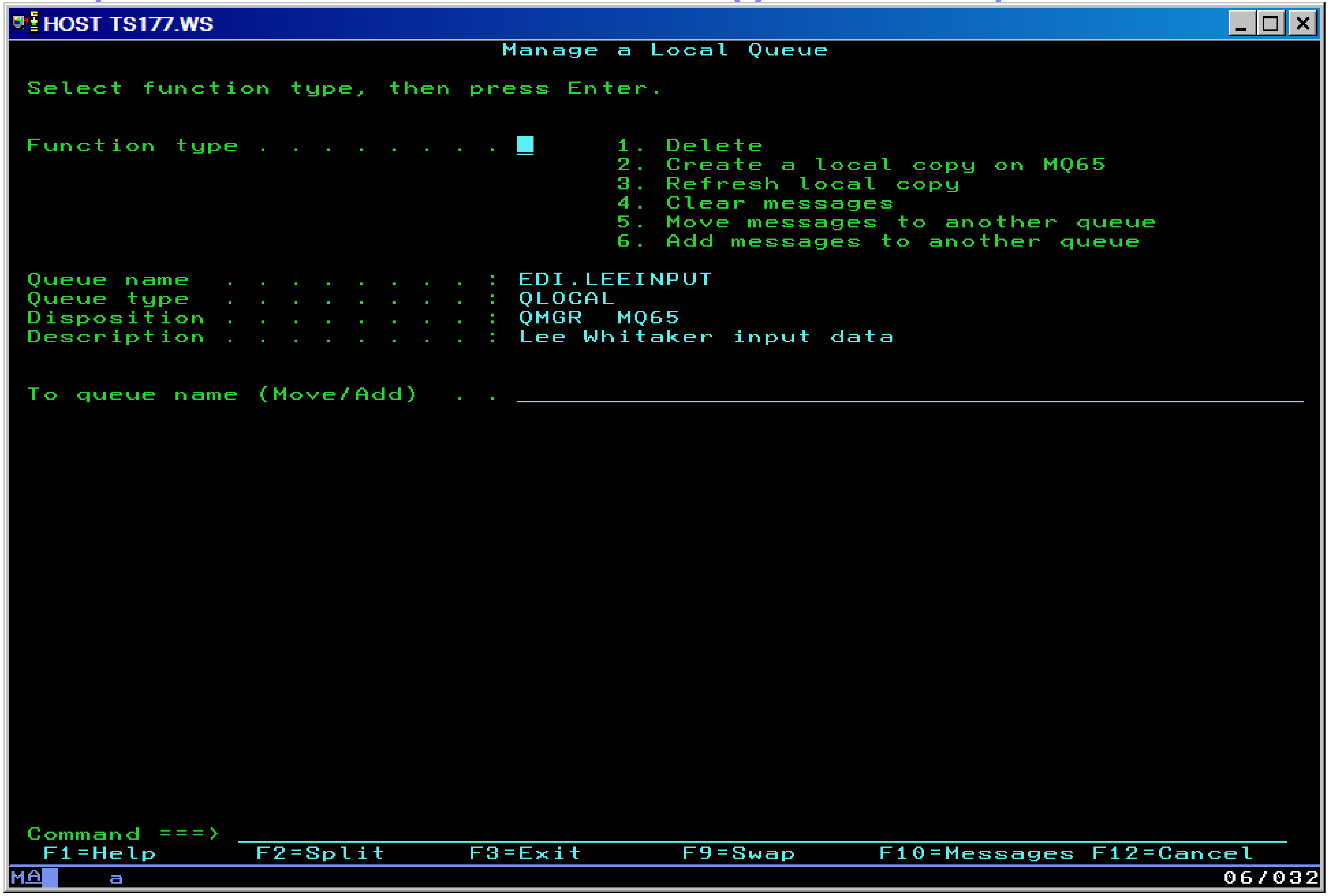
```
Command ===>
 F1=Help       F2=Split      F3=Exit       F9=Swap     F10=Messages F12=Cancel
MA     a                                                                42/015
```

z/OS Operations Panel



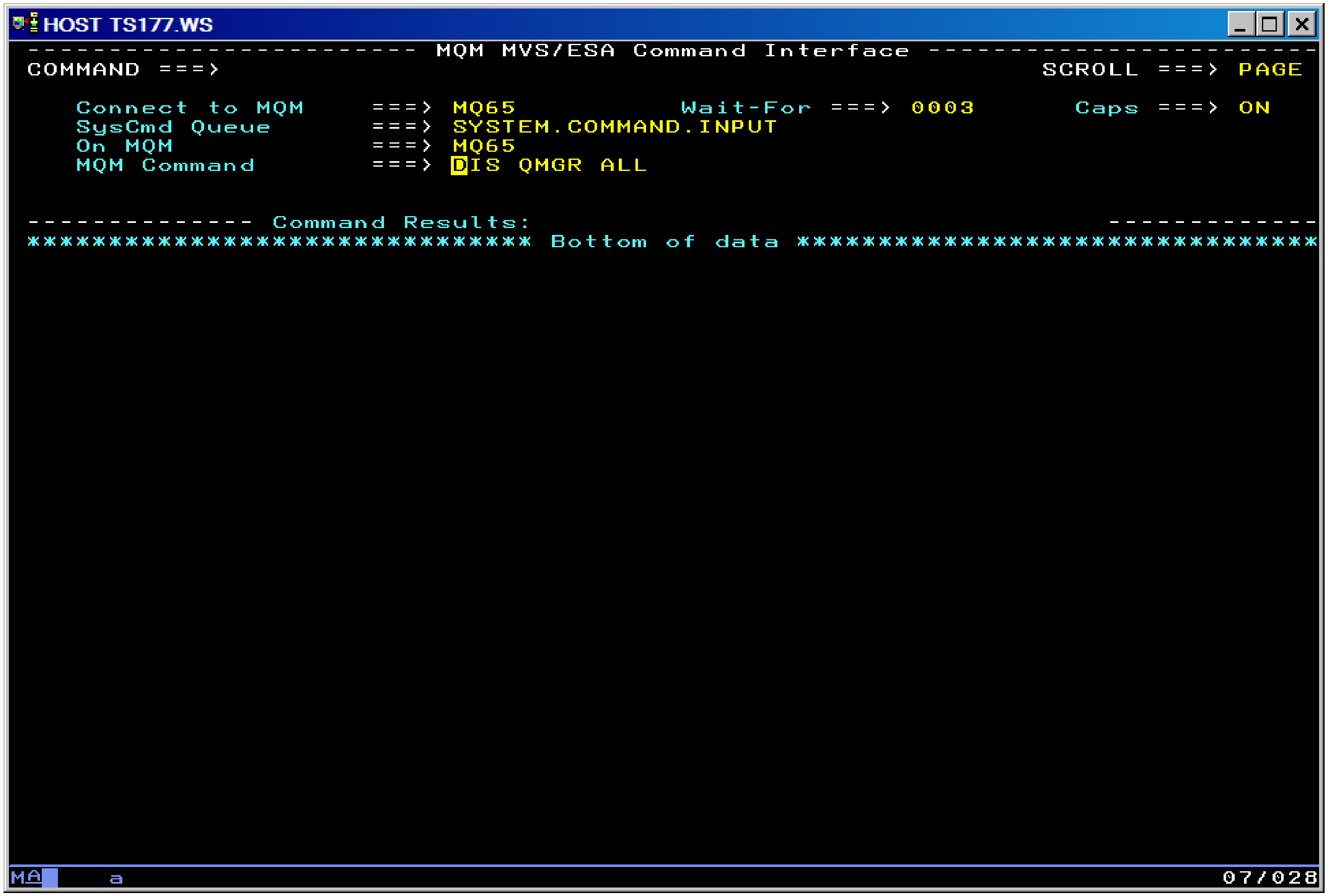
```
                         Display a System Function
 S                             Display messages                    Row 1 of 13

 F    CSQX830I MQ65 CSQXRDQM Channel initiator active
      CSQX845I MQ65 CSQXRDQM TCP/IP system name is TCPIP
      CSQX846I MQ65 CSQXRDQM TCP/IP listener INDISP=QMGR started,
      for port 1665 address *
      CSQX847I MQ65 CSQXRDQM LU 6.2 listener INDISP=QMGR started,
 A    for LU name MQ65
      CSQX831I MQ65 CSQXRDQM 8 adapter subtasks started, 8 requested
      CSQX832I MQ65 CSQXRDQM 5 dispatchers started, 5 requested
      CSQX840I MQ65 CSQXRDQM 2 channel connections current, maximum 200
      CSQX841I MQ65 CSQXRDQM 0 channel connections active, maximum 200
      CSQX842I MQ65 CSQXRDQM 0 channel connections starting,
      2 stopped, 0 retrying
      CSQ9022I MQ65 CSQXCRPS ' DISPLAY DQM' NORMAL COMPLETION
                          ******** End of list ********

      Command ===>
       F1=Help      F2=Split     F3=Exit      F7=Bkwd      F8=Fwd       F9=Swap
      F12=Cancel

 Command ===>
  F1=Help      F2=Split     F3=Exit      F9=Swap      F10=Messages F12=Cancel
MA    a                                                                 37/019
```

z/OS Operations Panel: manage local queue



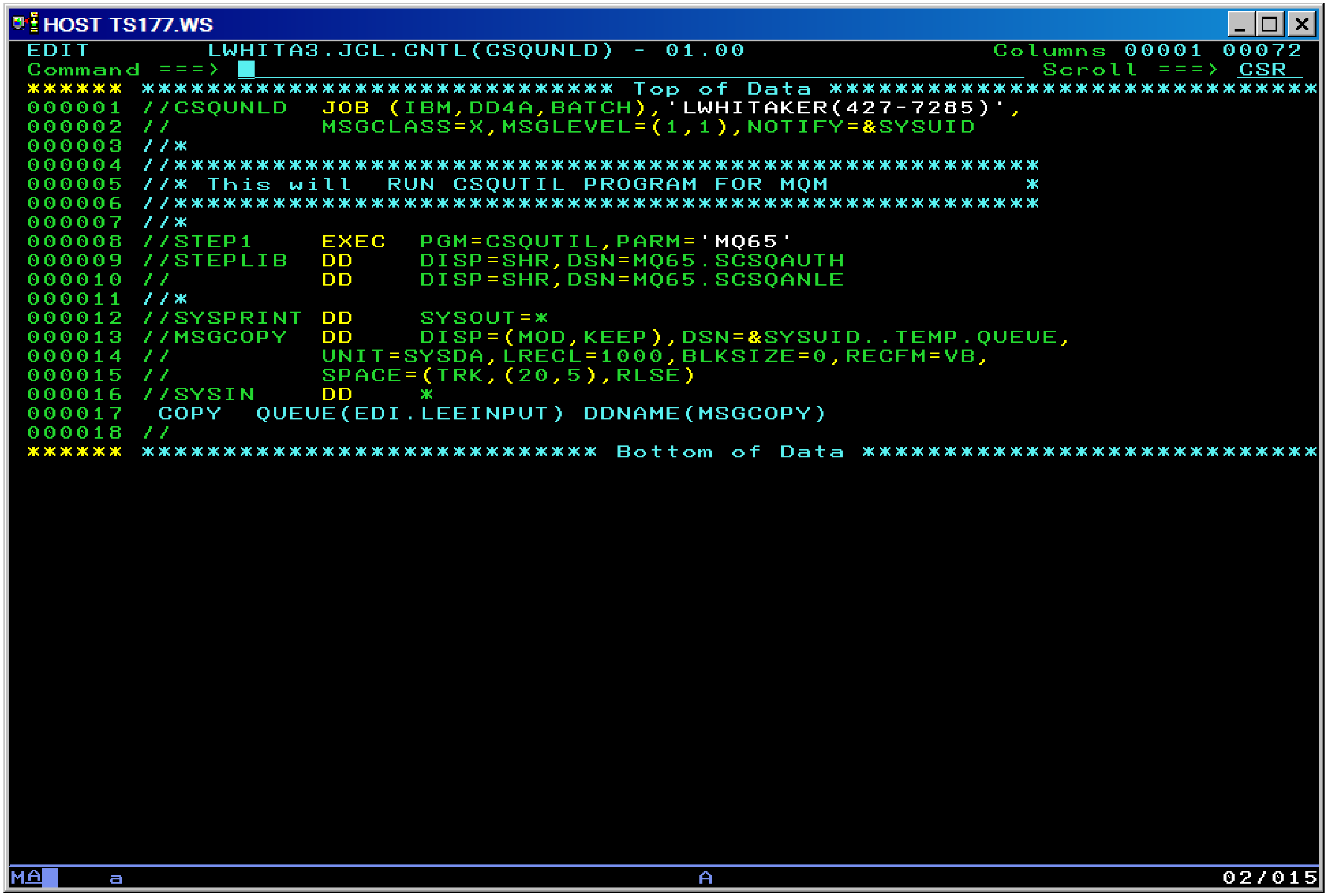
```
                              Manage a Local Queue

 Select function type, then press Enter.

 Function type . . . . . . . . _      1. Delete
                                      2. Create a local copy on MQ65
                                      3. Refresh local copy
                                      4. Clear messages
                                      5. Move messages to another queue
                                      6. Add messages to another queue

 Queue name  . . . . . . . . . : EDI.LEEINPUT
 Queue type  . . . . . . . . . : QLOCAL
 Disposition . . . . . . . . . : QMGR   MQ65
 Description . . . . . . . . . : Lee Whitaker input data

 To queue name (Move/Add)  . . ________________________________________________

 Command ===> ________________________________________________________________
  F1=Help       F2=Split      F3=Exit       F9=Swap     F10=Messages F12=Cancel
MA       a                                                                06/032
```

Command Interface Panel



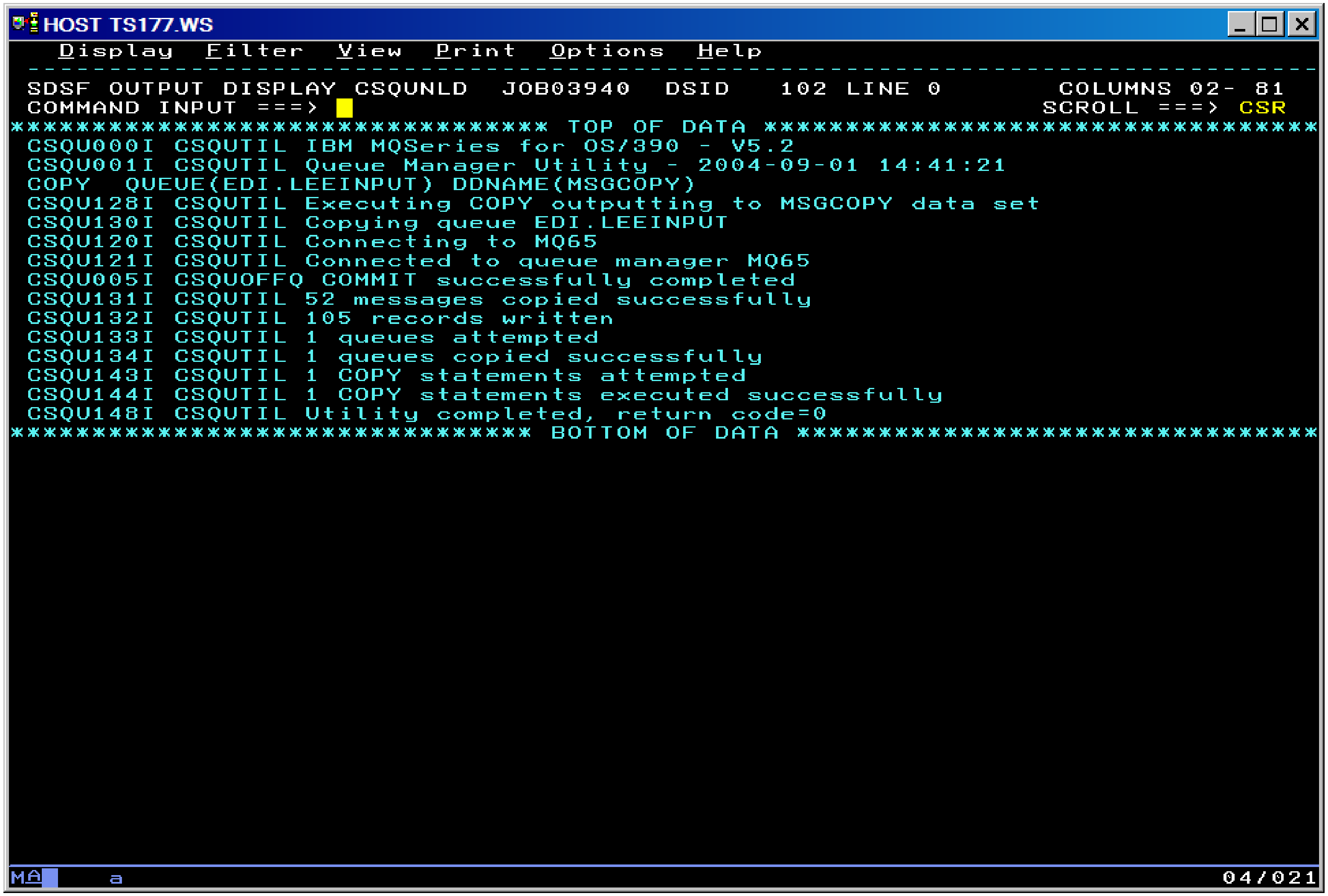
```
 ------------------------ MQM MVS/ESA Command Interface ------------------------
 COMMAND ===>                                                   SCROLL ===> PAGE

    Connect to MQM     ===> MQ65          Wait-For ===> 0003       Caps ===> ON
    SysCmd Queue       ===> SYSTEM.COMMAND.INPUT
    On MQM             ===> MQ65
    MQM Command        ===> DIS QMGR ALL


 -------------- Command Results:                                  -------------
 ******************************* Bottom of data ********************************
```

CSQUTIL with COPY



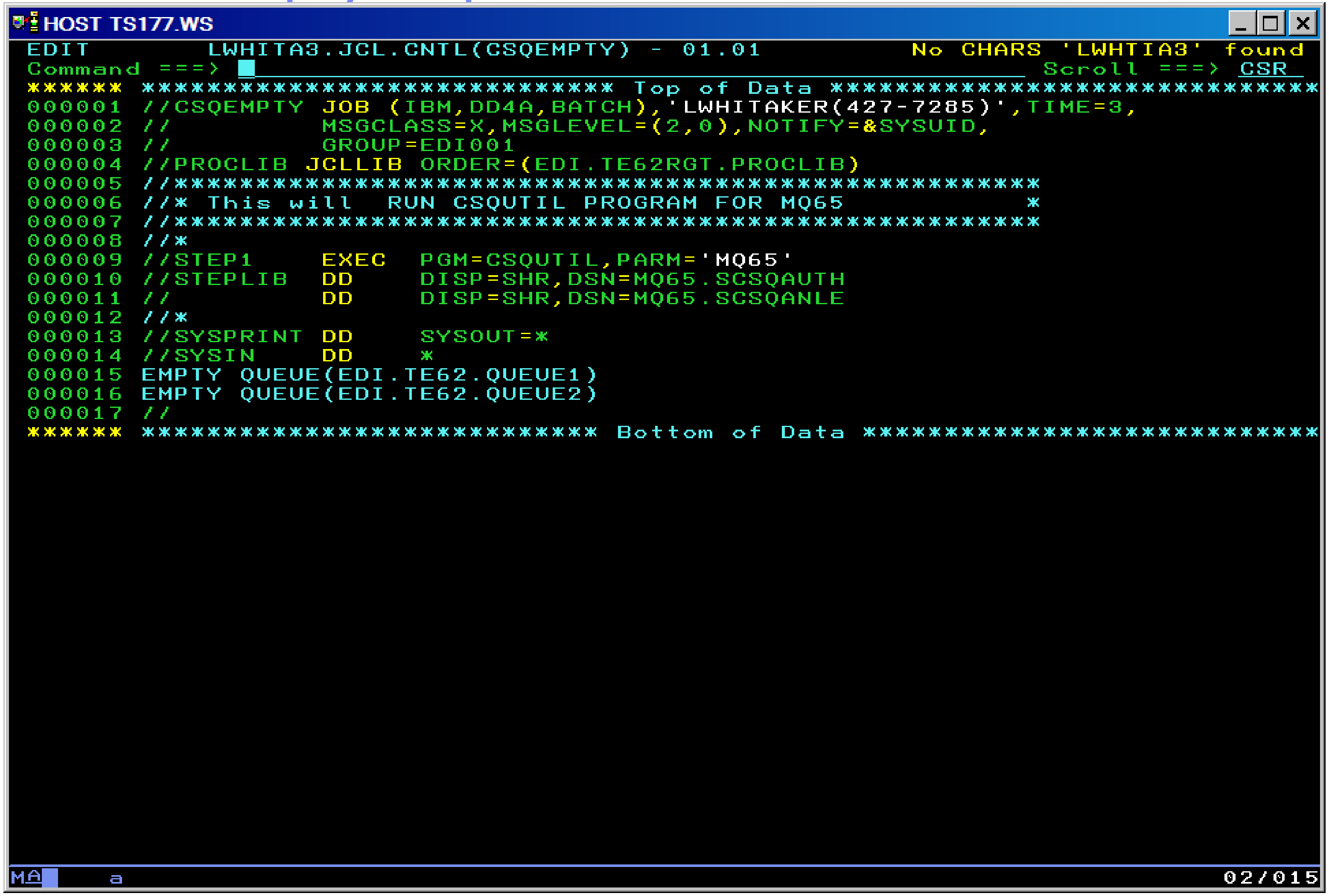
```
EDIT       LWHITA3.JCL.CNTL(CSQUNLD) - 01.00                Columns 00001 00072
Command ===>                                                   Scroll ===> CSR
****** ***************************** Top of Data ******************************
//CSQUNLD  JOB (IBM,DD4A,BATCH),'LWHITAKER(427-7285)',
//           MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*
//*****************************************************
//* This will  RUN CSQUTIL PROGRAM FOR MQM            *
//*****************************************************
//*
//STEP1      EXEC  PGM=CSQUTIL,PARM='MQ65'
//STEPLIB    DD    DISP=SHR,DSN=MQ65.SCSQAUTH
//           DD    DISP=SHR,DSN=MQ65.SCSQANLE
//*
//SYSPRINT   DD    SYSOUT=*
//MSGCOPY    DD    DISP=(MOD,KEEP),DSN=&SYSUID..TEMP.QUEUE,
//           UNIT=SYSDA,LRECL=1000,BLKSIZE=0,RECFM=VB,
//           SPACE=(TRK,(20,5),RLSE)
//SYSIN      DD    *
 COPY  QUEUE(EDI.LEEINPUT) DDNAME(MSGCOPY)
//
****** **************************** Bottom of Data ****************************

MA    a                                  A                               02/015
```

CSQUTIL with COPY



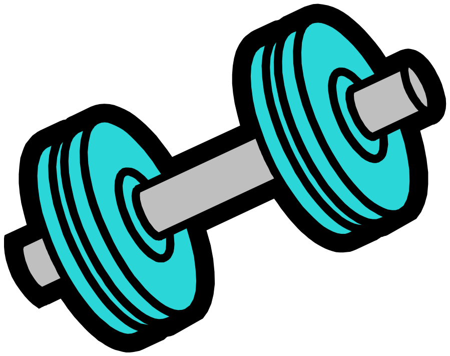
```
 SDSF OUTPUT DISPLAY CSQUNLD  JOB03940  DSID    102 LINE 0       COLUMNS 02- 81
 COMMAND INPUT ===>                                             SCROLL ===> CSR
******************************** TOP OF DATA **********************************
 CSQU000I CSQUTIL IBM MQSeries for OS/390 - V5.2
 CSQU001I CSQUTIL Queue Manager Utility - 2004-09-01 14:41:21
 COPY  QUEUE(EDI.LEEINPUT) DDNAME(MSGCOPY)
 CSQU128I CSQUTIL Executing COPY outputting to MSGCOPY data set
 CSQU130I CSQUTIL Copying queue EDI.LEEINPUT
 CSQU120I CSQUTIL Connecting to MQ65
 CSQU121I CSQUTIL Connected to queue manager MQ65
 CSQU005I CSQUOFFQ COMMIT successfully completed
 CSQU131I CSQUTIL 52 messages copied successfully
 CSQU132I CSQUTIL 105 records written
 CSQU133I CSQUTIL 1 queues attempted
 CSQU134I CSQUTIL 1 queues copied successfully
 CSQU143I CSQUTIL 1 COPY statements attempted
 CSQU144I CSQUTIL 1 COPY statements executed successfully
 CSQU148I CSQUTIL Utility completed, return code=0
******************************** BOTTOM OF DATA *******************************
```

CSQUTIL: empty a queue



```
EDIT       LWHITA3.JCL.CNTL(CSQEMPTY) - 01.01             No CHARS 'LWHTIA3' found
Command ===>                                                      Scroll ===> CSR
****** ***************************** Top of Data ******************************
//CSQEMPTY JOB (IBM,DD4A,BATCH),'LWHITAKER(427-7285)',TIME=3,
//         MSGCLASS=X,MSGLEVEL=(2,0),NOTIFY=&SYSUID,
//         GROUP=EDI001
//PROCLIB JCLLIB ORDER=(EDI.TE62RGT.PROCLIB)
//*****************************************************
//* This will  RUN CSQUTIL PROGRAM FOR MQ65           *
//*****************************************************
//*
//STEP1    EXEC  PGM=CSQUTIL,PARM='MQ65'
//STEPLIB  DD    DISP=SHR,DSN=MQ65.SCSQAUTH
//         DD    DISP=SHR,DSN=MQ65.SCSQANLE
//*
//SYSPRINT DD    SYSOUT=*
//SYSIN    DD    *
EMPTY QUEUE(EDI.TE62.QUEUE1)
EMPTY QUEUE(EDI.TE62.QUEUE2)
//
****** **************************** Bottom of Data ****************************

MA     a                                                                  02/015
```

WebSphere MQ Exercise

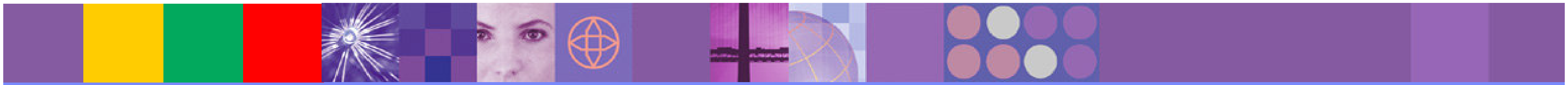


- What you will be able to do:
  - Create a remote queue definition
  - Create a transmission queue
  - Create a channel
  - Send a test message to the remote queue

IBM Software Group

WDI Server

WebSphere. software

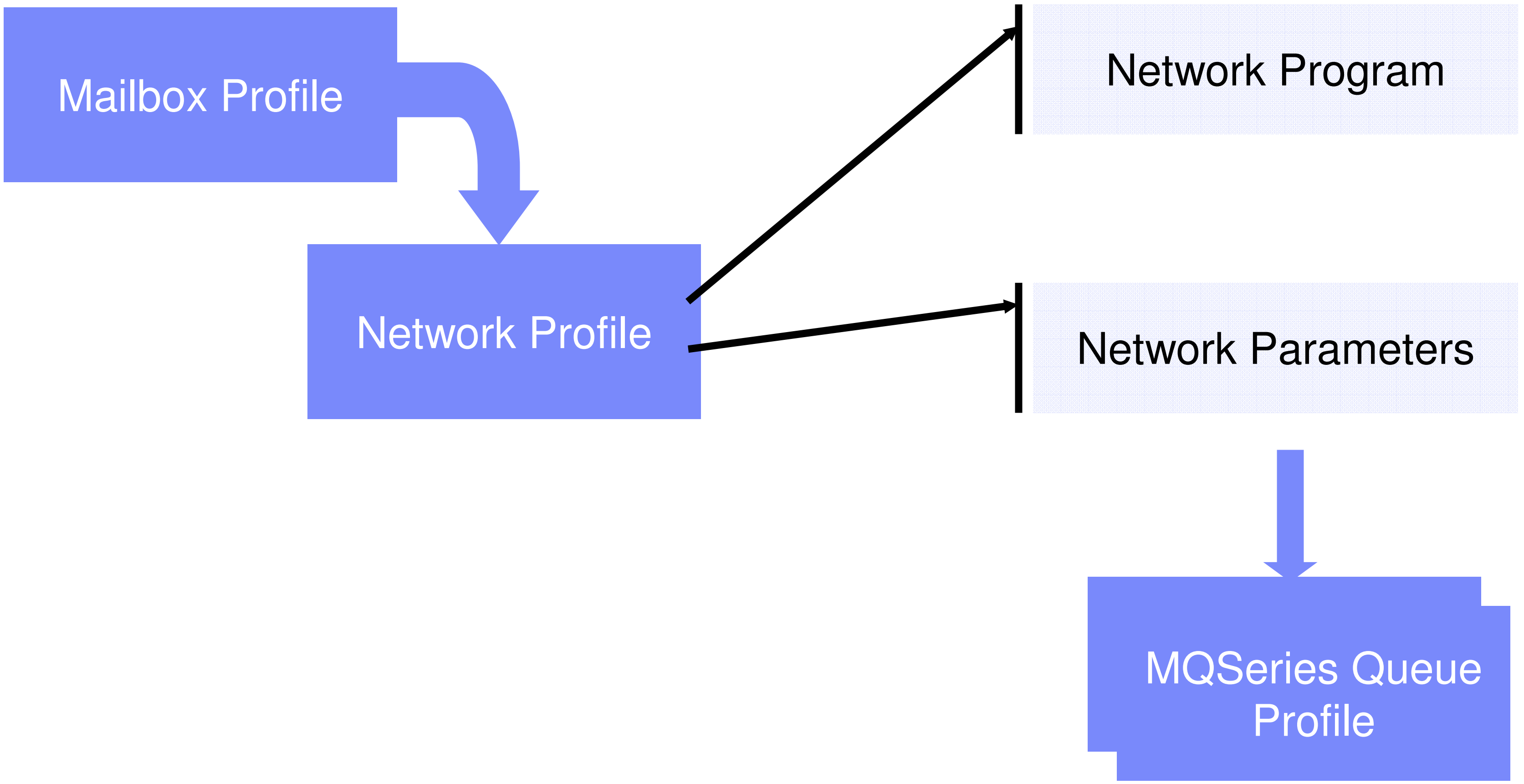


e business software

WDI Profiles for WMQ Integration

- Mailbox Profile
- Network Profile
- MQSeries Queue Profile

WDI Profiles Overview



Mailbox Profile

- **Mailbox Profile** describes the individual users or groups who request network services for sending or receiving documents, EDI transactions, messages, or files on a WebSphere Data Interchange System.

Network Profile

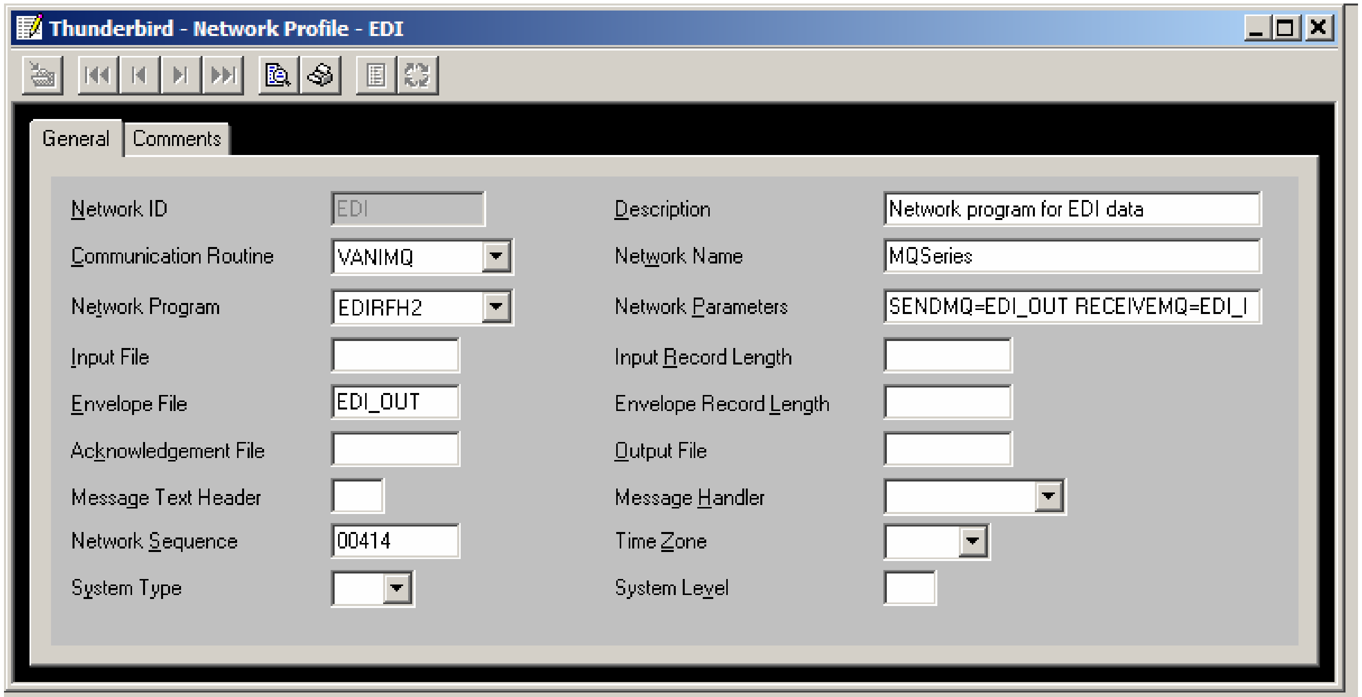
- **Network Profile** defines the characteristics of a network you use for communications with trading partners. It provides options and parameters used when communicating with the network. A Network profile may also be used to identify the MQSeries Queue profiles used when sending or receiving documents from a trading partner.

Network Programs

- EDIMQS
  - Simple MQ interface
- EDIRFH2
  - Includes RFH header support
- EDICYCL
  - Sets special properties required by Cyclone

Network Parameters

- SENDMQ=PROF_NAME
- RECEIVEMQ=PROF_NAME



Network ID	EDI	Description	Network program for EDI data
Communication Routine	VANIMQ	Network Name	MQSeries
Network Program	EDIRFH2	Network Parameters	SENDMQ=EDI_OUT RECEIVEMQ=EDI_I
Input File		Input Record Length	
Envelope File	EDI_OUT	Envelope Record Length	
Acknowledgement File		Output File	
Message Text Header		Message Handler	
Network Sequence	00414	Time Zone	
System Type		System Level	

MQSeries Queue Profile

- **MQSeries Queue Profile** associates a logical name with a physical WebSphere MQ message queue. Message queue processing options are also identified in the MQSeries Queue profile. There will at least one MQSeries Queue profile for each WebSphere MQ message queue used by WebSphere Data Interchange. Additional MQSeries Queue profiles can be defined for a single message queue when different processing options will be used for the same message queue.

Profiles Summary

- Defined Mailbox Profiles
- Defined Network Profiles
- Defined three WMQ related Network Programs
- Defined purpose of Network Parameters
- Defined the MQSeries Queue Profile

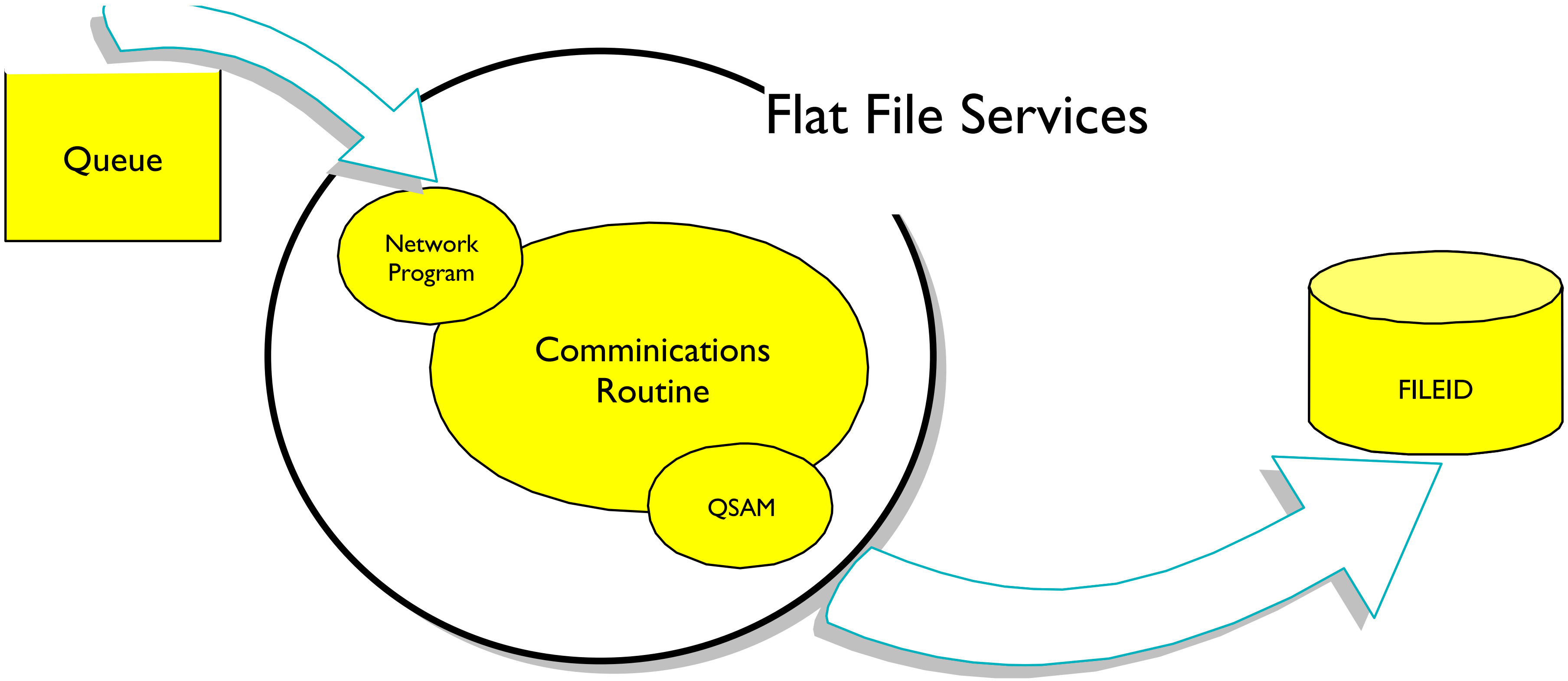
WDI Reading from and Writing to WMQ Queues

- SEND/RECEIVE COMMANDS
- PERFORM TRANSFORM
- Using MQ File Type
- Exercise

Perform Commands

- SEND and RECEIVE
- SENDFILE and RECVFILE
- RECEIVE AND SEND

Perform Commands: Receive (or Send)



PERFORM RECEIVE WHERE REQID()

Perform Commands – Send/Receive

- REQID – valid mail box profile
- FILEID – Logical name of file to send
- CLEARFILE – should the file be cleared before the Receive or after the Send
- DELFILE – remove ADF input after a TRANSLATE

PERFORM Commands– Reading From WMQ

- PERFORM TRANSFORM WHERE
  - INFILE(Q_PROF) INTYPE(MQ)
  - OUTFILE(Q_PROF) OUTTYPE(MQ)

PERFORM Commands – OUTTYPE(MQ)

- Things to keep in mind with OUTTYPE(MQ)
  - Physical filename needs to be specified
  - OUTFILE must identify a **Mailbox Profile**

```
FF0007 Data was written to EDI_OUT. Message control number or document id was 000000022.
FF0585 The PERFORM TRANSFORM command completed successfully.
FF0110 Unable to retrieve mailbox or requestor profile EDI_OUT, return code = 8 extended return code = 301.
```

Using File Type MQ – Usages

- Document Destination
  - Name specifies the Mailbox Profile
  - Type drop down must specify “MQ”

INTYPE(MQ) Example

- Two XML to EDI scenarios
  - PERFORM TRANSFORM INTYPE(MQ)
  - PERFORM TRANSFORM using File Type MQ on Rule

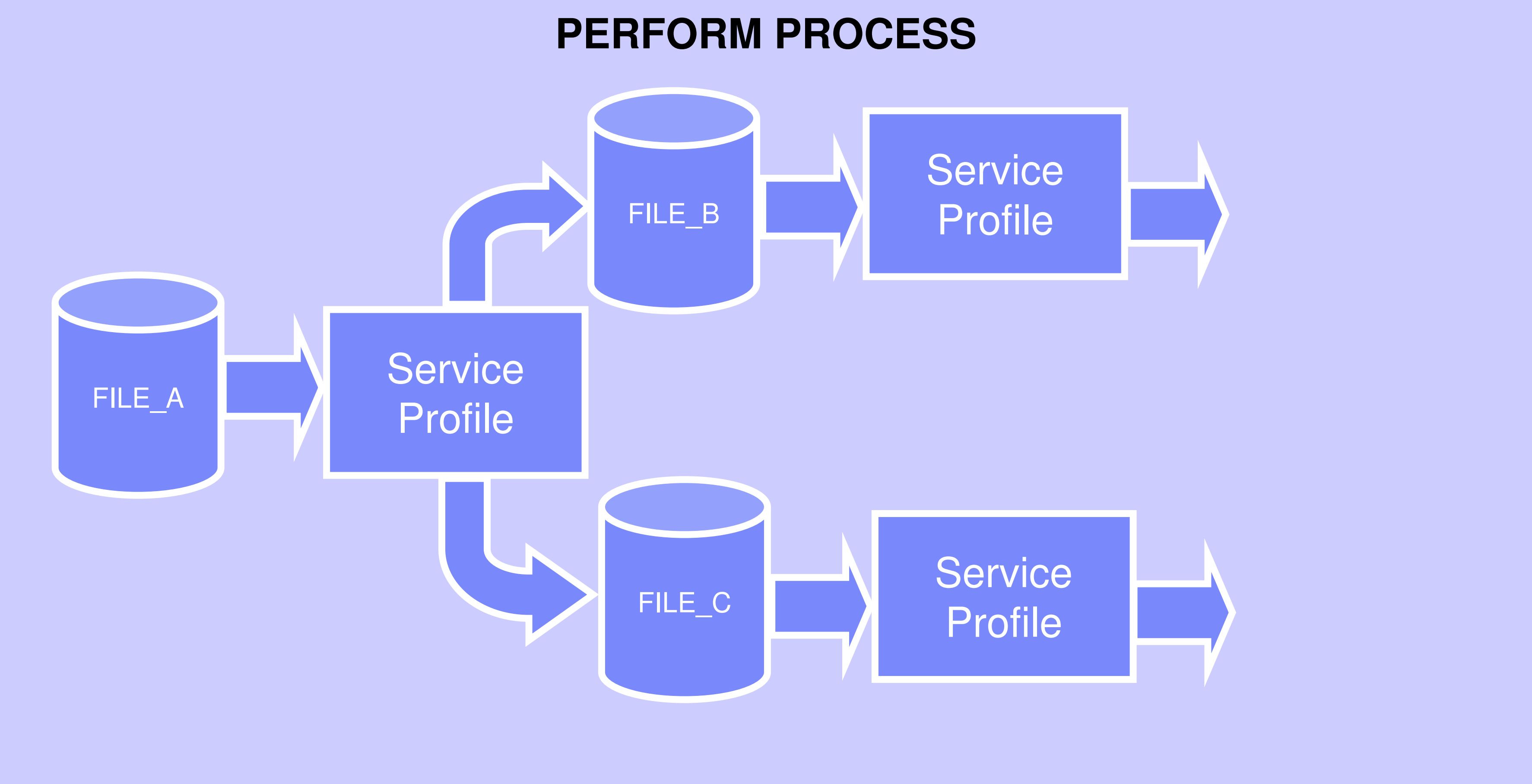
Command Chaining

- We want to be able to understand the following:
  - What is command chaining?
  - What functionality does it provide?
  - What is a Service Profile?
  - What determines which Service Profile is used?
  - How does command chaining work?

What is Command Chaining?

- Introduced with the PROCESS keyword
- Uses PERFORM command templates
- Allows dynamic command substitutions
- Invokes commands when logical files are available
- Used by WDIAdapter and WDIServer

Command Chaining Overview



What functionality does it provide?

- New level of flexibility for solutions.
  - Appropriate action can be taken for each input source.
  - The output of one command becomes input for additional commands.
- Provides the foundation upon which the WDIAdapter and WDIServer programs are built.

What is a Service Profile?

- Service Profiles provide the link between a data source and the specific PERFORM command that will be used to process data found on that data source. The service profile also defines all the files required by the PERFORM command.

Elements of a Service Profile

- Service Profile name
- PERFORM command
- Criteria for continuing the chain
- Input and output file names

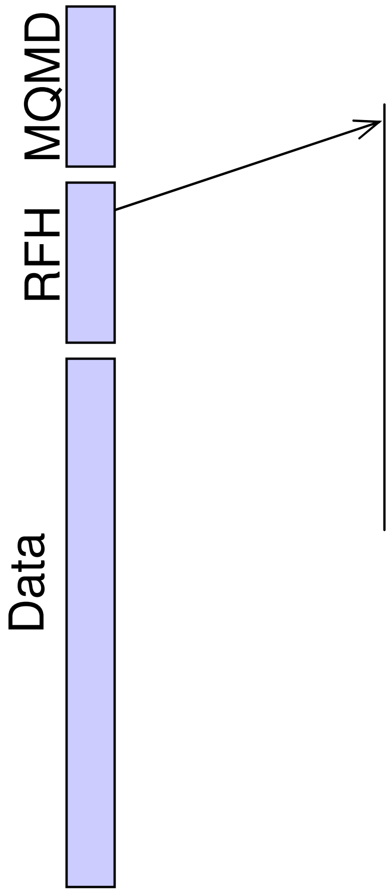
What determines which Service Profile is used?

- Process to select a service profile:
  - ➢ Logical file has been closed
  - ➢ Logical file name matches a Service Profile
  - ➢ The current error level matches Service profile criteria

Command Substitution

- Information from the RFH2 header can be used to specify parameters in a PERFORM command in a Service Profile.
  - Perform command arguments are put into the <usr> folder of the RFH2 header.
  - &TagName is used on the PERFORM COMMAND to substitute the value from the RFH2 header.

Command Substitution



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

<mcd>	Message content descriptor
<psc>	Publish/subscribe command
<pscr>	Publish/subscribe command response
<usr>	**<fmtid>POHD1</fmtid>**

Command Substitution

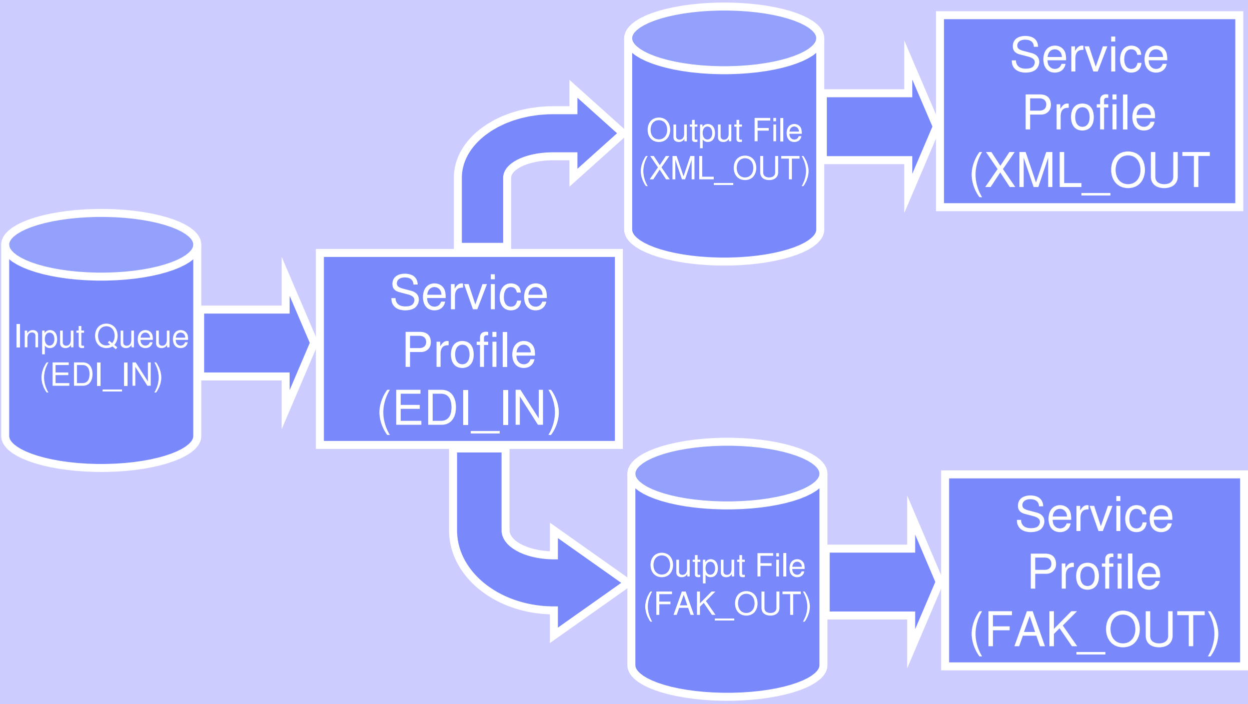
- PERFORM TRANSFORM WHERE DOCID(&fmtid)
  - &fmtid will be replaced with the value from the element in the user folder

How does command chaining work?

- PERFORM RECEIVE AND PROCESS
  - ➢ REQID – Names a Mailbox Profile
  - ➢ FILEID – Logical filename
- PERFORM PROCESS
  - ➢ FILEID – Logical filename

EDI to XML Example

PERFORM RECEIVE AND PROCESS WHERE REQID(EDI_IN) FILEID(EDI_IN)



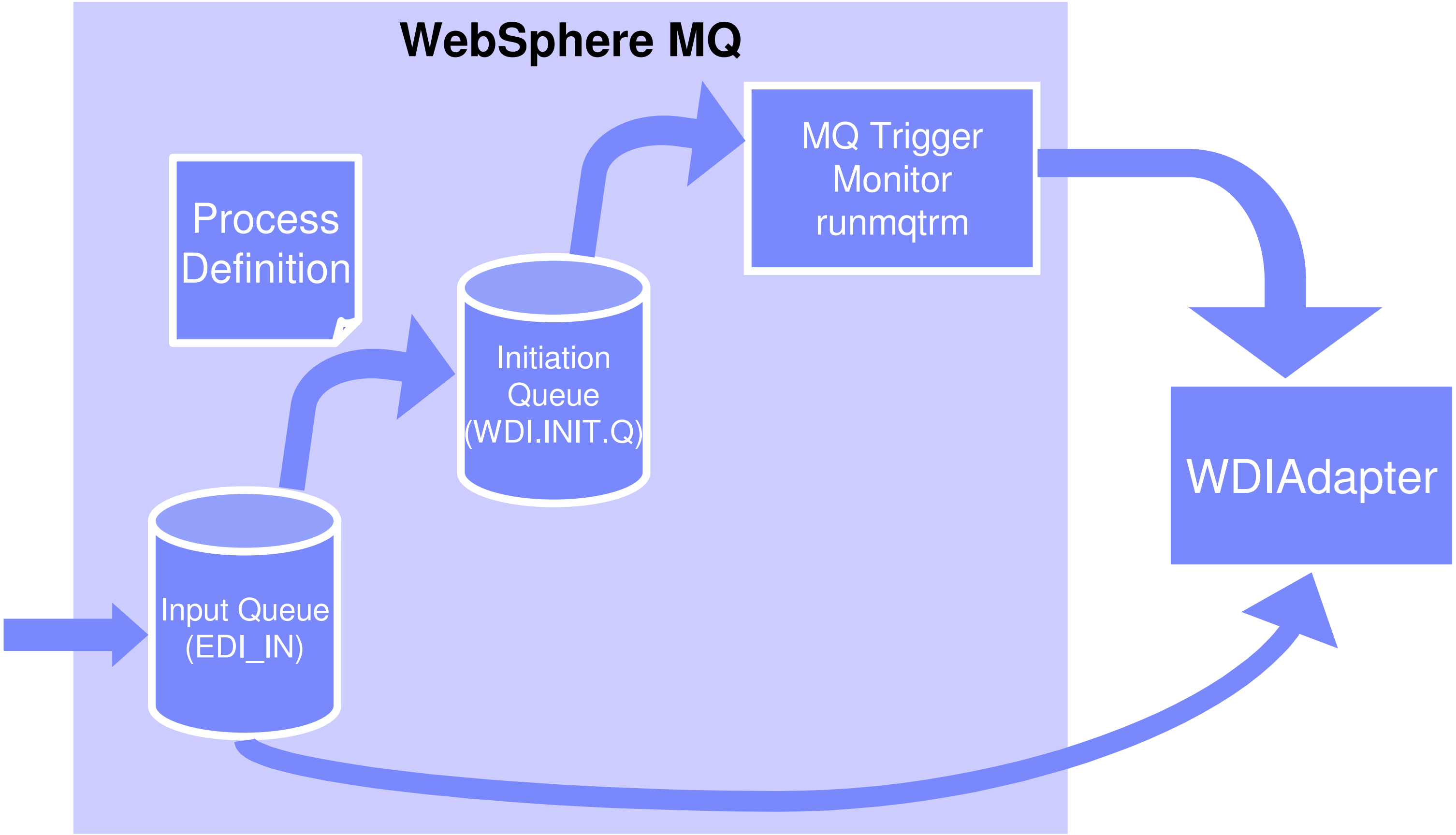
Command Chaining Summary

- Defined command chaining
- Explained Service Profiles
- Explained how Service Profiles are selected
- Discussed ways to start command chaining

WDIAdapter Program

- Overview of WDIAdapter
- Setting up WMQ Objects
- Configuring WDIAdapter Parameters
- Tips and Tricks

WDIAdapter Overview

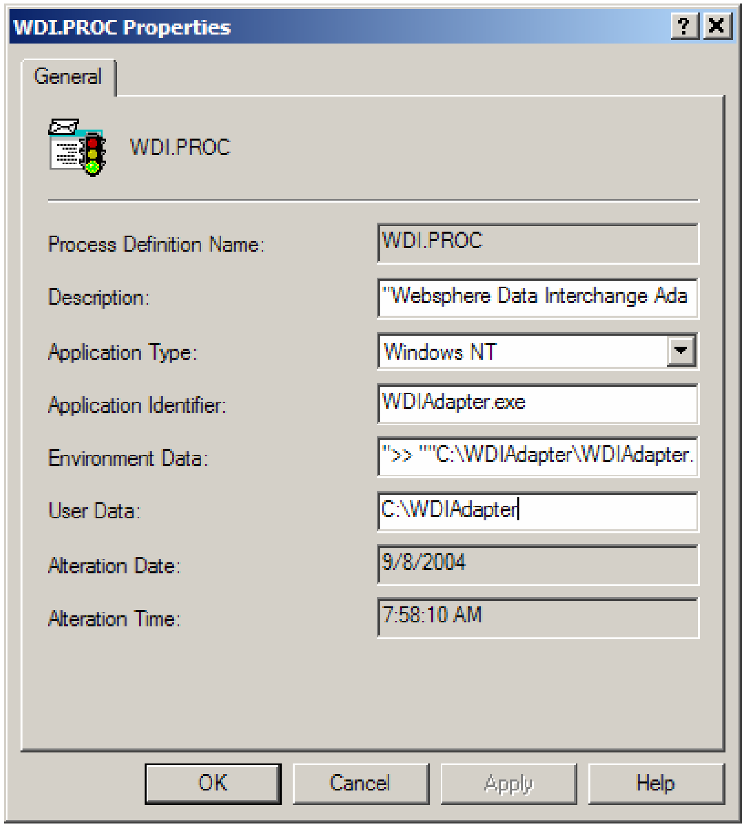


Setting up MQ Objects for WDIAdapter

- Setup Script “wdimqcommands.txt”
  - Process Definition – WDI.PROC
  - Initiation Queue – WDI.INIT.Q
  - Default Input/Output queues
- Execute script with runmqsc command
  - runmqsc < wdimqcommands.txt

Process Definition

- WDI.PROC
  - ➢Environment Data
  - ➢User Data


WDI.PROC Properties

General

WDI.PROC

Process Definition Name:	WDI.PROC
Description:	"Websphere Data Interchange Ada
Application Type:	Windows NT
Application Identifier:	WDIAdapter.exe
Environment Data:	">> ""C:\WDIAdapter\WDIAdapter.
User Data:	C:\WDIAdapter
Alteration Date:	9/8/2004
Alteration Time:	7:58:10 AM

Configuring WDIAdapter with “wdi.properties”

- The “wdi.properties” file specifies:
  - Directories to be used
  - Database Name
  - Language Code
  - Wait Time

Directories to be used

- runtimedirectory – location of WDI executables
- datadirectory – working directory for WDIAdapter
- rcvdirectory – Receive file directory
- prtdirectory – Print file directory
- dtddirectory – directory for DTD files

Database Name, Language Code, Wait Time

- Database Name “EDIEC32E”
  - User id and password can be specified
- Language code
  - Identifies the Language Profile to be used
- Wait Time
  - Indicates how long WDIAdapter will wait for a message

Tips and Tricks

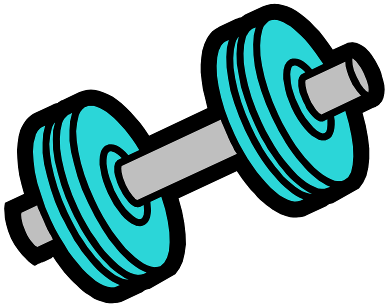
WebSphere. software

e business software

Tips and Tricks

- WDIAdapter will not Trigger
  - Check the open input count on WDI.INIT.Q
  - Ensure triggering is enabled
- WDIAdapter on AIX
  - Check the permissions on the runmqtrm executable
  - Ensure ownership of working directory for WDIAdapter

WDIAdapter Exercise

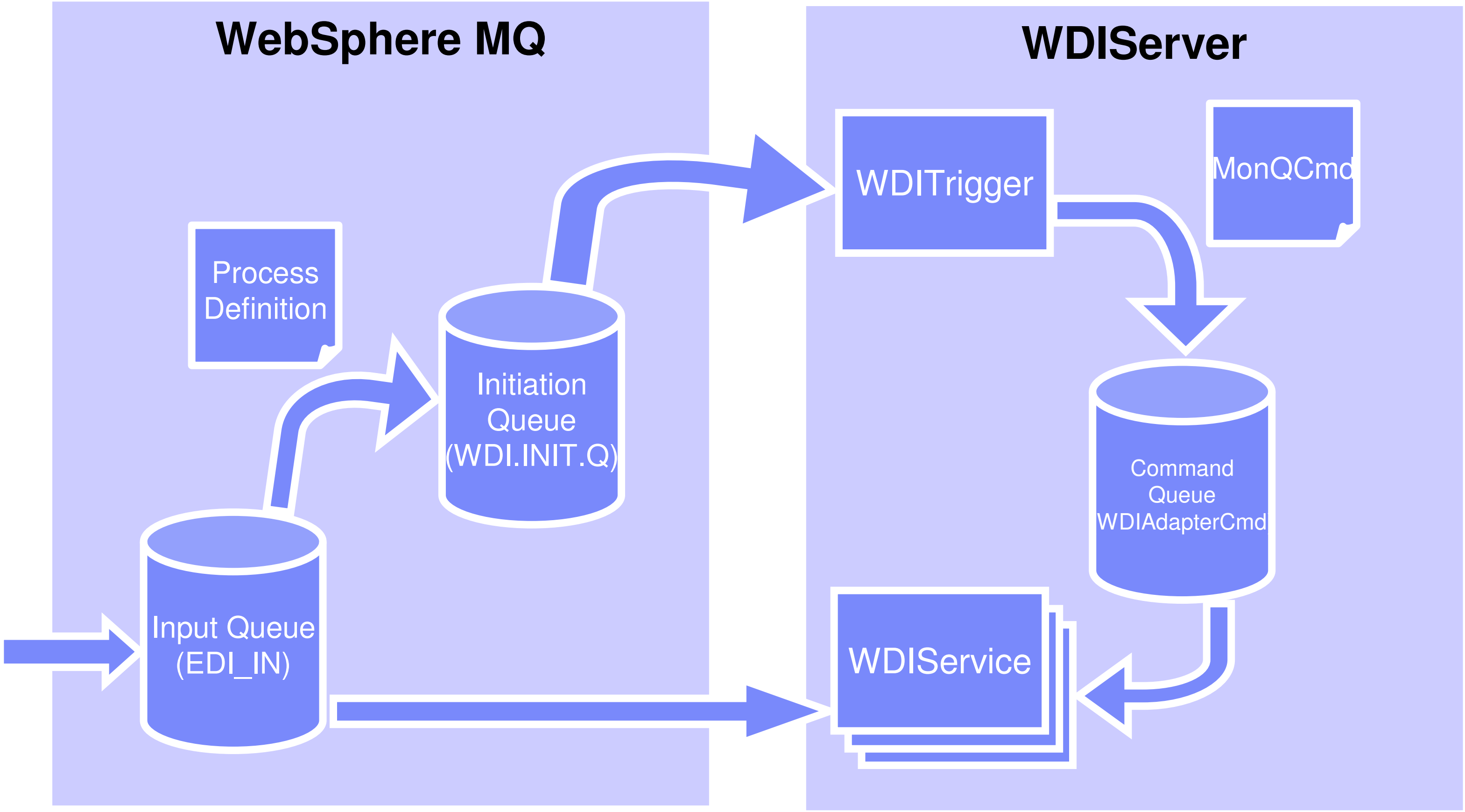


- In this exercise you will:
  - Create an input queue
  - Setup triggering
  - Start the trigger monitor (runmqtrm)
  - Transform an XML document to EDI using WDIAdapter

WDIServer Program

- Overview of WDIServer
- Setting up WMQ Objects
- Configuring WDIServer
- Added Benefits of WDIServer

WDIServer Overview



Setting up MQ Objects

- AdvAdapterMQ.txt
  - Creates WDI.TRANSLATOR.PROC
  - Creates WDIAdapterCmd Queue

WDI.TRANSLATOR.PROC

- New process definition for WDIServer
- Parameters can be passed on “Environment Data”
  - NumThreads – number of translators from pool to assign
  - Timeout – amount of time in microseconds to wait
  - ReqId – value for REQID keyword
  - FileId – value for FILEID keyword

Input Queue - Trigger Data

- Trigger Data can pass parameters
- Environment Data can pass parameters
- Not on both

Large Queue Names

- Long queue names supported with WDIServer
  - Passing ReqId and FileId allows this
  - Default is to use Queue Name for both

Configuring WDIServer

- qmgrname
- initq
- numtranslators
- genprtfile
  - Always/onerror – on error is default
- Keeparchive
  - Always/onerror – never is default

Starting WDIServer

- WDIServer will look for properties when started
  - Current working directory
  - WDISERVER_PROPERITES environment variable
- Initializes translator pool (WDIService processes)
  - Each translator creates a directory
  - WDITransCmdQ_XXXX
- Creates archive directory

Starting WDIServer on AIX

- Extended Shared Memory
  - export EXTSHM=ON

Added Benefits of WDIServer

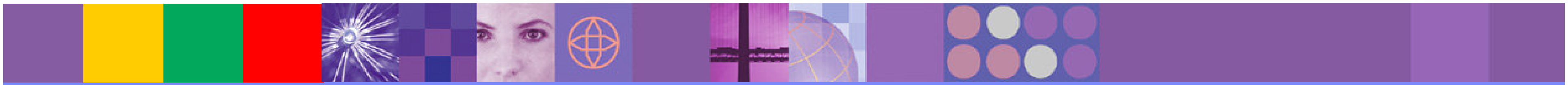
- Multiple Translators per queue
- The ability to use long queue names

WDIServer Exercise

- In this exercise you will:
  - Perform the same transformation from the WDIAdapter exercise using the WDIServer program.

WDI and CICS

WebSphere software

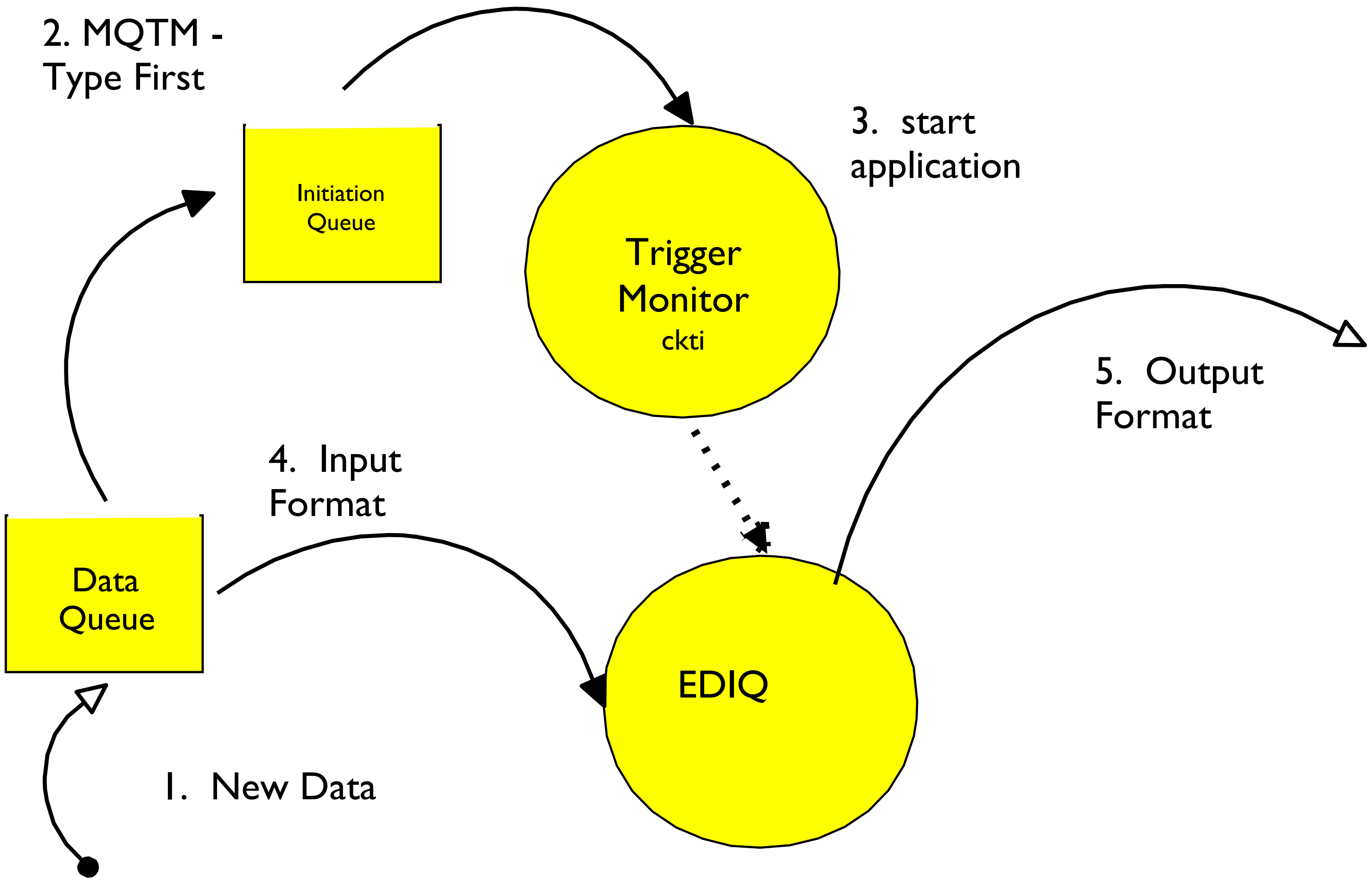


e business software

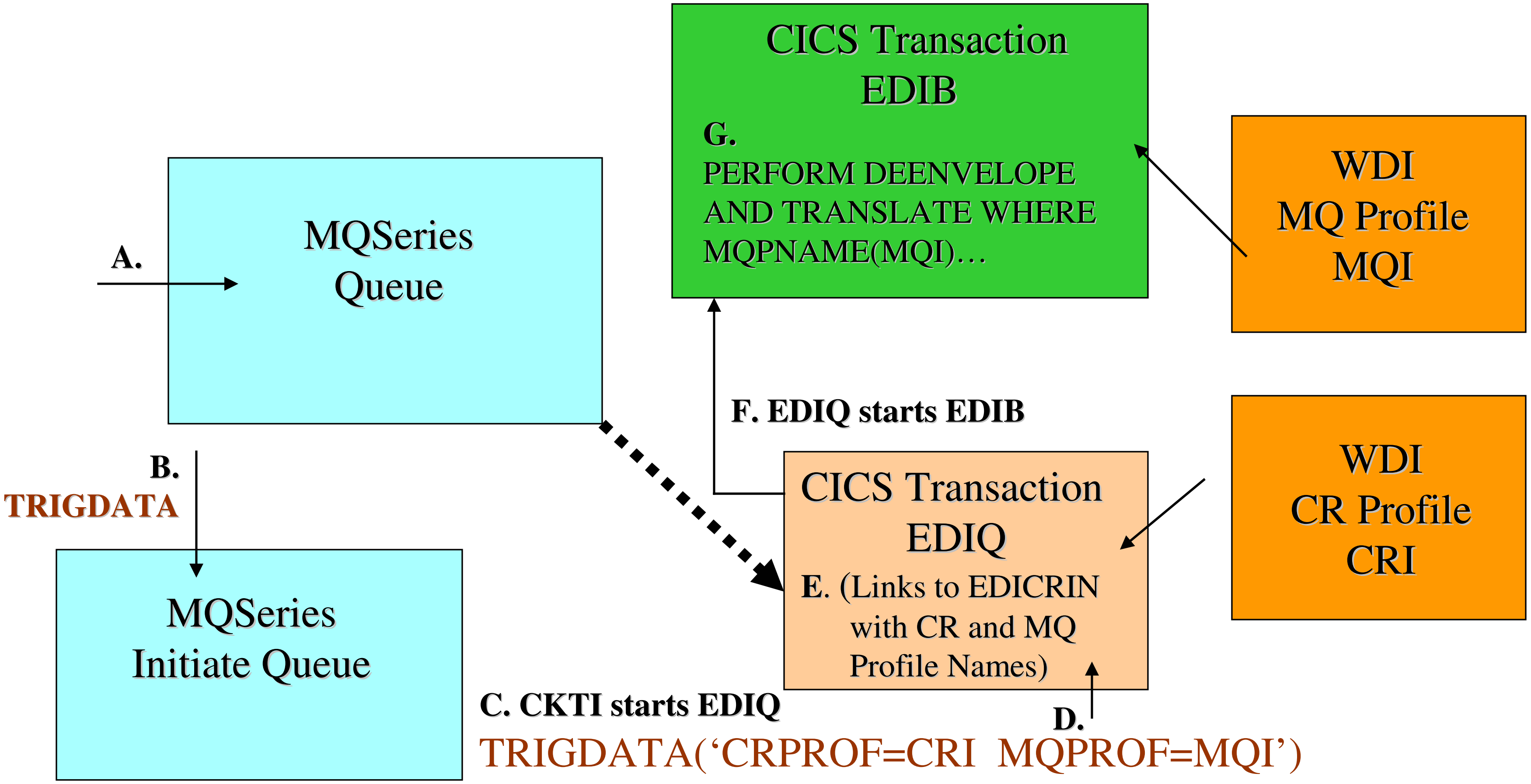
EDIQ

- Provides dynamic behavior in CICS
- Uses triggering
- Based on Continuous Receive

EDIQ


2. MQTM - Type First
Initiation Queue
Trigger Monitor ckti
3. start application
5. Output Format
4. Input Format
Data Queue
EDIQ
1. New Data

Transaction overview



CICS Transaction EDIB

G.
PERFORM DEENVELOPE AND TRANSLATE WHERE MQPNAME(MQI)…

WDI
MQ Profile
MQI

A.

MQSeries
Queue

F. EDIQ starts EDIB

B.
TRIGDATA

MQSeries
Initiate Queue

CICS Transaction
EDIQ

E. (Links to EDICRIN with CR and MQ Profile Names)

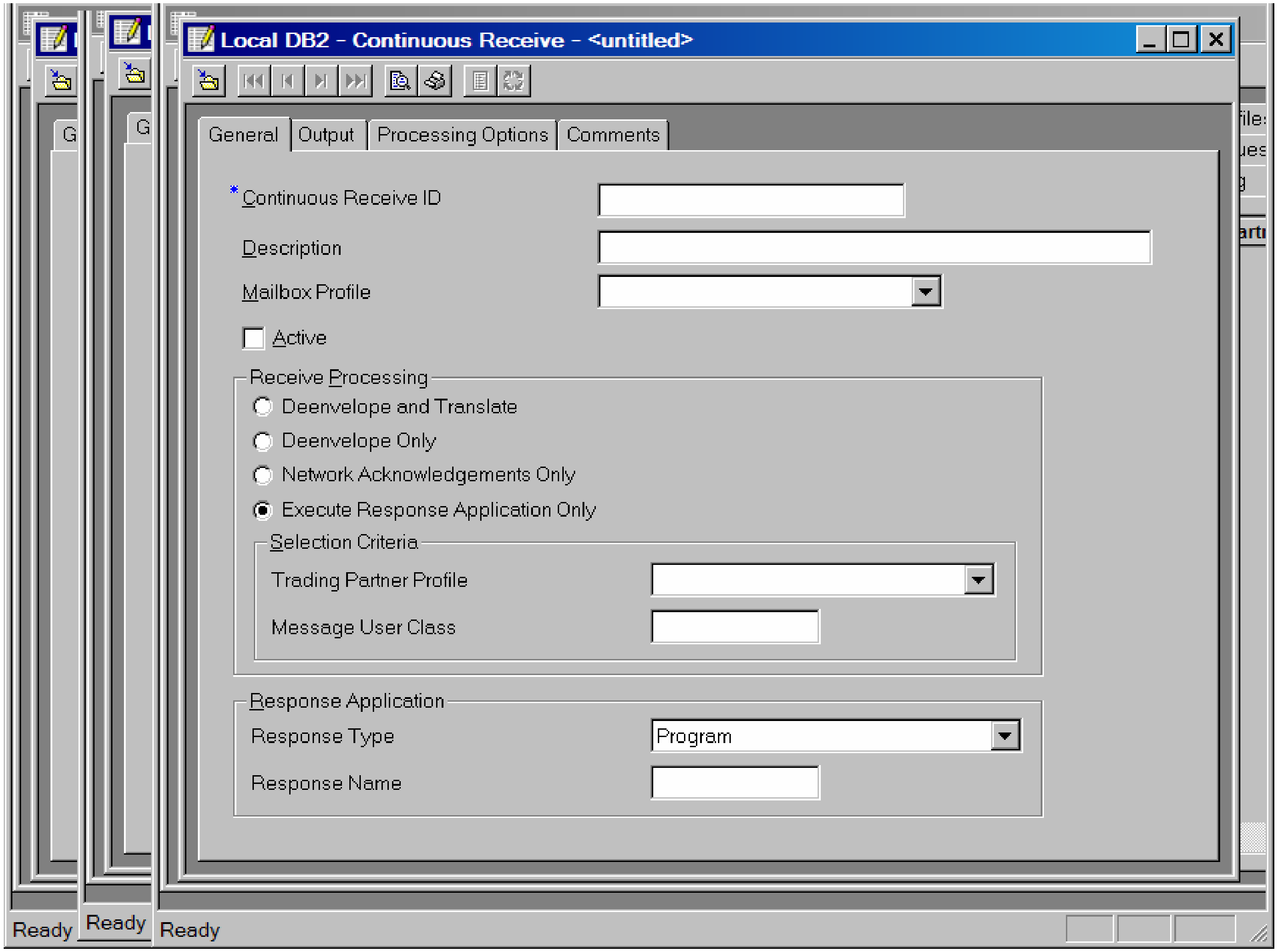
WDI
CR Profile
CRI

C. CKTI starts EDIQ

D.

TRIGDATA(‘CRPROF=CRI MQPROF=MQI’)

CR Profile



Local DB2 - Continuous Receive - <untitled>

*Continuous Receive ID	
Description	
Mailbox Profile	

□ Active

Receive Processing

○ Deenvelope and Translate
○ Deenvelope Only
○ Network Acknowledgements Only
● Execute Response Application Only

Selection Criteria

Trading Partner Profile	
Message User Class	

Response Application

Response Type	Program
Response Name	

Ready

Questions and Discussion