WebSphere MQ z/OS a-z

z/OS MQ Programming

An IBM Educational Approach

WebSphere MQ z/OS Programming

MQI

What is the MQI?

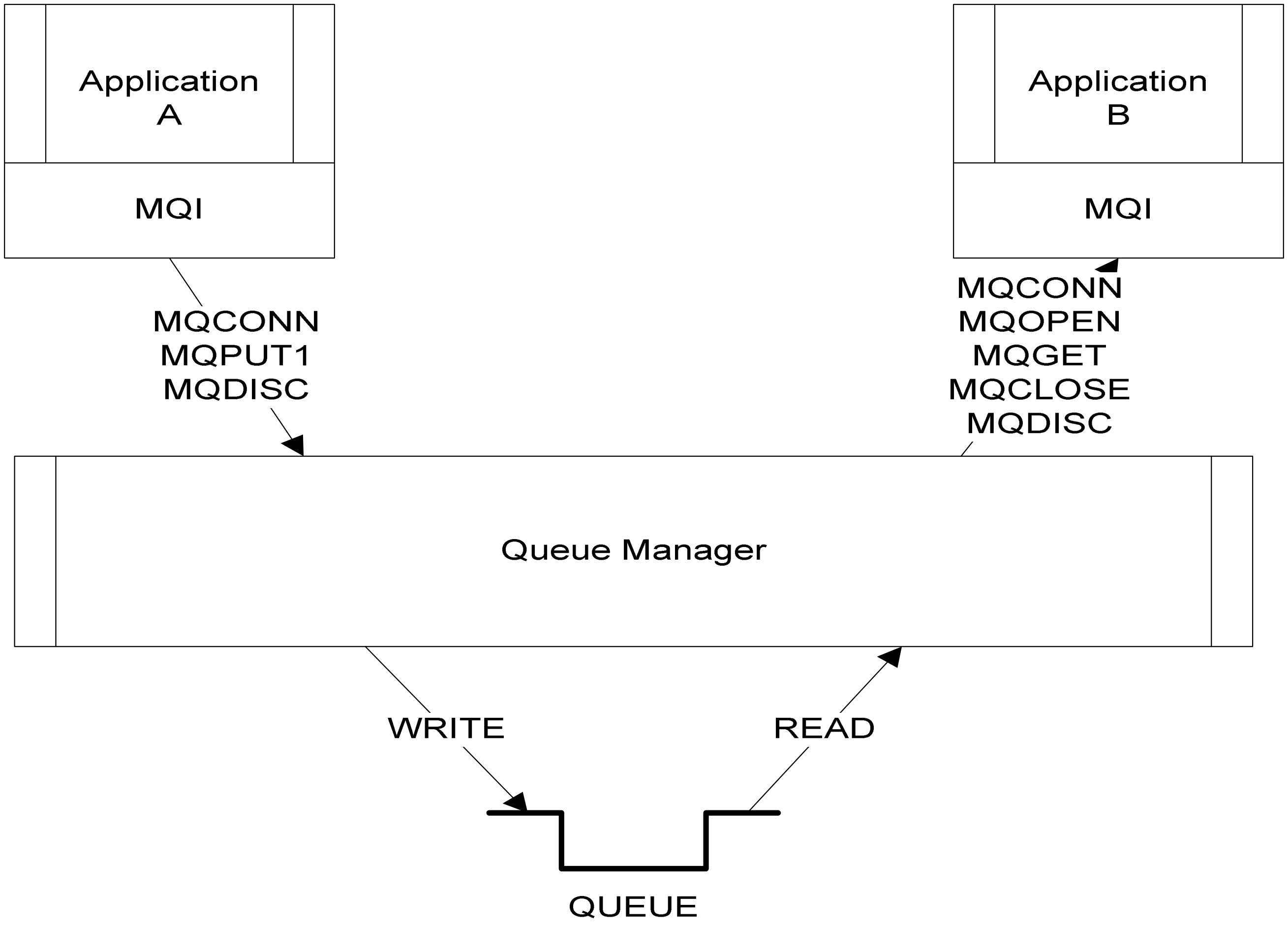
The Message Queuing Interface is a highly portable API for access to MQSeries resources

- MQCONN - Connect to a QMGR
- MQOPEN - Open a queue for processing
- MQPUT - Put one or more messages on a queue
- MQPUT1 - Put one message on a queue
- MQGET - Get a message from a queue
- MQCLOSE - Close a queue to terminate processing
- MQDISC - Disconnect from a QMGR
- Additional MQI calls

WebSphere MQ z/OS Programming

MQI Example

Queuing Example



WebSphere MQ z/OS Programming

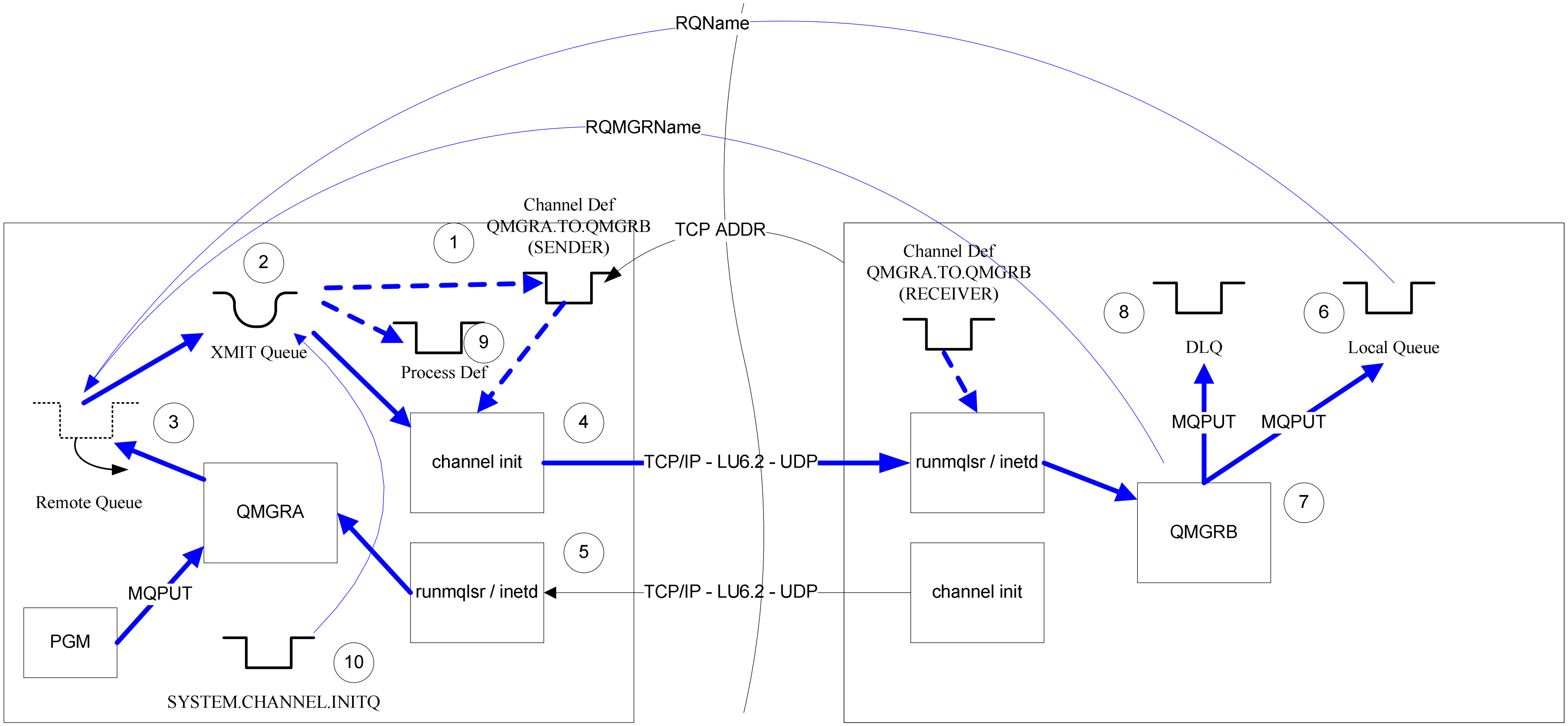
MQI

MQI Supported Languages

- **C**
  - all platforms
- **C++**
  - AIX
  - Sun Solaris
  - Digital OpenVMS
  - Windows
  - HP-UX
  - Windows
- **COBOL**
  - all
  - Windows
- **PL/I**
  - z/OS
  - AIX
  - Windows
- **Assembler**
  - OS/390
- **RPG**
  - OS/400
- **TAL**
  - Tandem
- **Visual Basic**
  - Windows
- **Java**
- **.NET**
- **JMS**

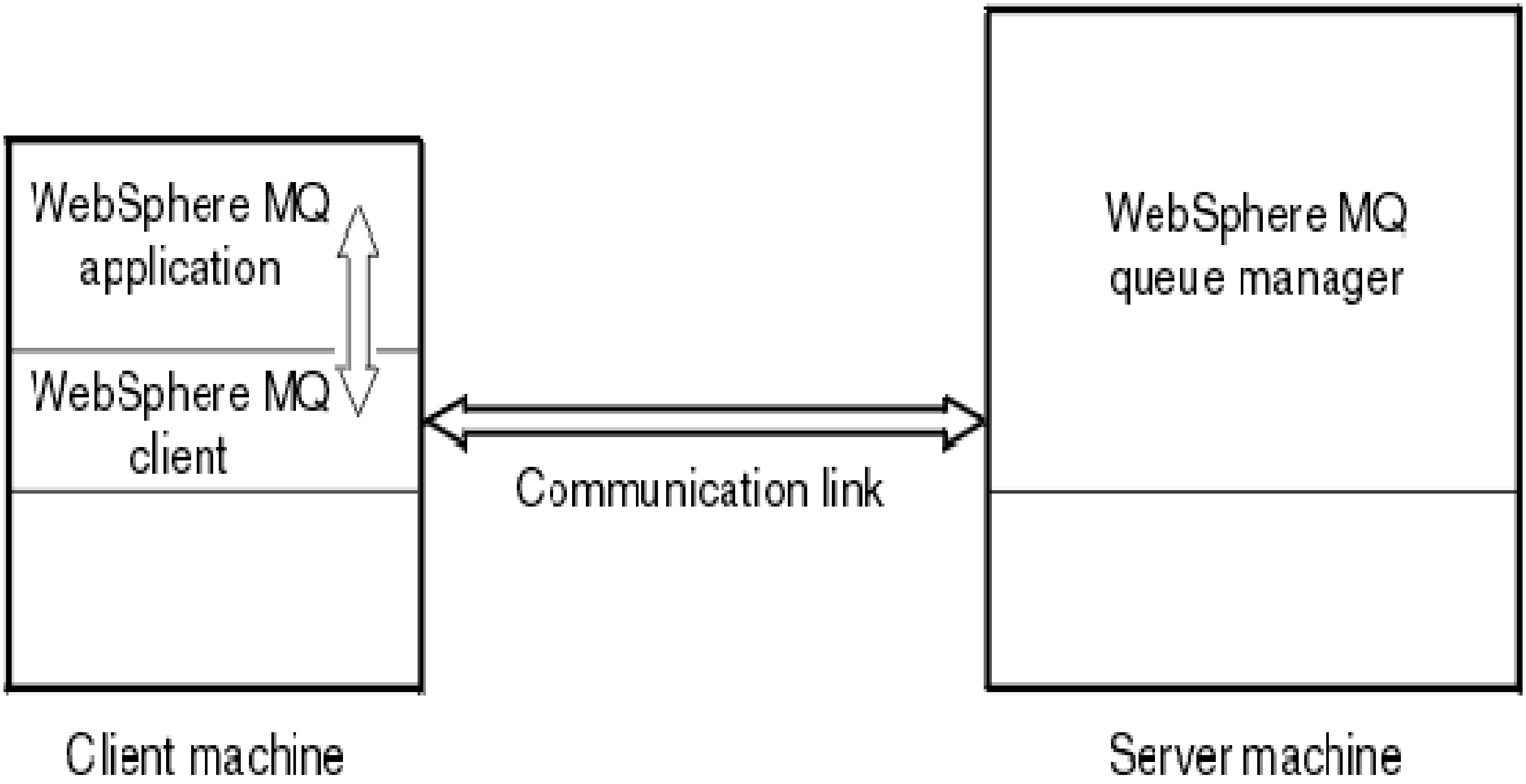
WebSphere MQ z/OS Programming

Distributed Queuing



WebSphere MQ z/OS Programming

MQClient



WebSphere MQ z/OS Programming

MQI Structures

The most commonly used MQI structures:

- MQOD Object Descriptor
  used on MQOPEN and MQPUT1 calls
- MQMD Message Descriptor
  used on MQGET, MQPUT and MQPUT1 calls
- MQPMO Put Message Options
  used on MQPUT and MQPUT1
- MQGMO Get Message Options
  used on MQGET

A number of other special use structures

WebSphere MQ z/OS Programming

COBOL Copybooks

Structures with initial values

- MQOD, MQMD, MQPMO, MQGMO

Structures initial values - working storage

- CMQODV, CMQMDV, CMQPMOV, CMQGMOV

Structures w/o initial values - linkage section

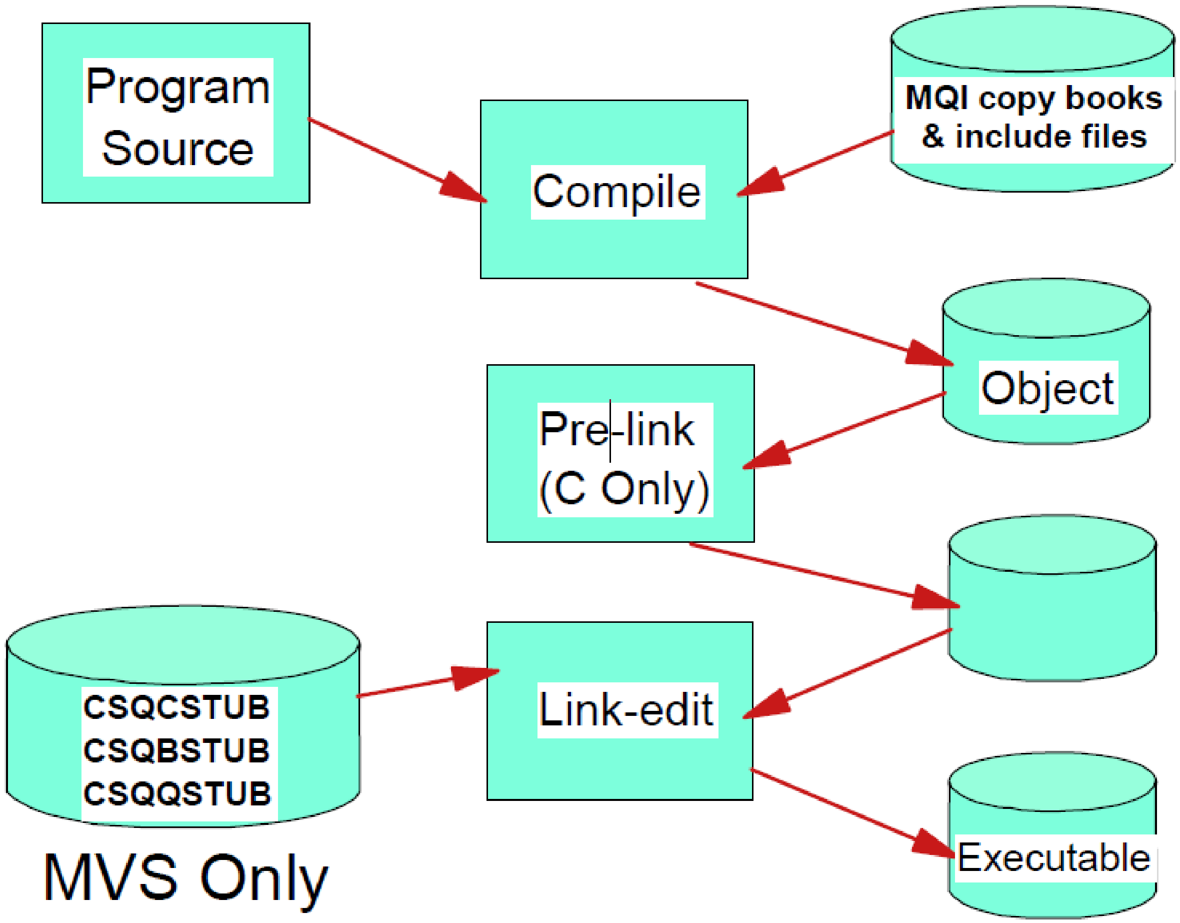
- CMQODL, CMQMDL, CMQPMOL, CMQGMOL

Constants

CMQV - completion codes, reason codes, options, constants

WebSphere MQ z/OS Programming

Program Preperation



WebSphere MQ z/OS Programming

MQOD Fields

Object Descriptor (MQOPEN/MQPUT1)

StrucId	(MQCHAR4: value 'OD ')
Version	(MQLONG)
ObjectType	(MQLONG)
ObjectName	(MQCHAR48)
ObjectQmgrName	(MQCHAR48)
DynamicQName	(MQCHAR48)
AlternateUserId	(MQCHAR12)
RecsPresent	(MQLONG)
KnownDestCount	(MQLONG)
UnknownDestCount	(MQLONG)
InvalidDestCount	(MQLONG)
ObjectRecOffset	(MQLONG)
ObjectRecPtr	(MQPTR)
ResponseRecPtr	(MQPTR)
AlternateSecurityId	(MQCHAR40)
ResolvedQName	(MQCHAR48)
ResolvedQMgrName	(MQCHAR48)

WebSphere MQ z/OS Programming

MQMD Fields

Message Descriptor (MQPUT/MQPUT/MQGET)

StrucId	(MQCHAR4; value 'MD ')
Version	(MQLONG)
Report	(MQLONG)
MsgType	(MQLONG)
Expiry	(MQLONG)
Feedback	(MQLONG)
Encoding	(MQLONG)
CodedCharSetId	(MQLONG)
Format	(MQLONG)
Priority	(MQLONG)
Persistence	(MQLONG)
MsgId	(MQBYTE24)
CorrelId	(MQBYTE24)
BackoutCount	(MQLONG)
ReplyToQ	(MQCHAR48)
ReplyToQMgr	(MQCHAR48)
UserIdentifier	(MQCHAR12)
AccountingToken	(MQBYTE32)
ApplIdentityData	(MQCHAR32)
PutApplType	(MQLONG)

WebSphere MQ z/OS Programming

MQMD Fields (cont)

Message Descriptor (MQPUT/MQPUT/MQGET)

• PutApplName	(MQCHAR28)
• PutDate	(MQCHAR8)
• PutTime	(MQCHAR8)
• ApplOriginData	(MQCHAR4)
• GroupId(1)	(MQBYTE24)
• MsgSeqNumber(1)	(MQLONG)
• Offset(1)	(MQLONG)
• MsgFlags(1)	(MQLONG)
• OriginalLength(1)	(MQLONG)

WebSphere MQ z/OS Programming

MQPMO Fields

Put Message Options (MQPUT/MQPUT1)

StrucId	(MQCHAR4: value 'PMO ')
Version	(MQLONG)
Options	(MQLONG)
Timeout	(MQLONG)
Context	(MQHOBJ)
KnownDestCount	(MQLONG)
UnKnownDestCount	(MQLONG)
InvalidDestCount	(MQLONG)
ResolvedQName	(MQCHAR48)
ResolvedQMgrName	(MQCHAR48)
RecsPresent	(MQLONG)
PutMsgRecFields	(MQLONG)
PutMsgRecOffset	(MQLONG)
PutMsgRecPtr	none
ResponseRecPtr	(MQPTR)

WebSphere MQ z/OS Programming

MQGMO Fields

Get Message Options (MQGET)

StrucId	(MQCHAR4: value 'GMO')
Version	(MQLONG)
Options	(MQLONG)
WaitInterval	(MQLONG)
Signal1	(MQLONG)
Signal2	(MQLONG)
ResolvedQName	(MQCHAR48)
MatchOptions	(MQLONG)
GroupStatus	(MQCHAR)
SegmentStatus	(MQCHAR)
Segmentation	(MQCHAR)
Reserved1	(MQCHAR)
MsgToken	(MQBYTE16)
ReturnedLength	(MQBYTE16)

WebSphere MQ z/OS Programming

Integration Patterns

Request/Reply

1. MQCONN
2. MQOPEN – Request Queue
3. MQPUT
4. MQCLOSE Request Queue
5. MQOPEN Reply Queue
6. MQGET
7. MQCLOSE Reply Queue
8. MQDISC
9. Terminate

Datagram

1. MQCONN
2. MQOPEN – Output Queue
3. <Records to process?> No=> #7
4. MQPUT
5. <Process>
6. Loop back to #3
7. MQCLOSE Reply Queue
8. MQDISC
9. Terminate

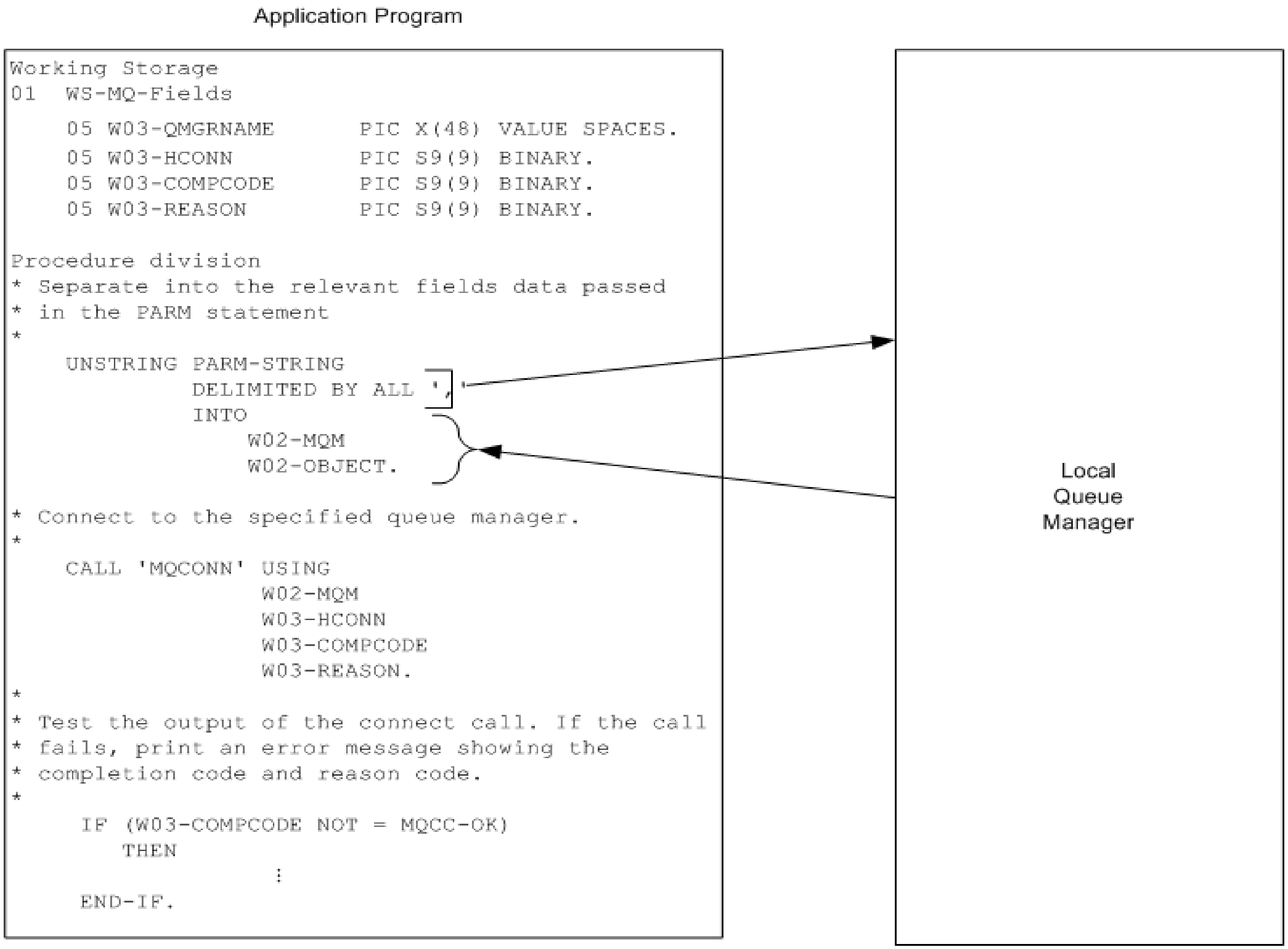
WebSphere MQ z/OS Programming

Interface Calls

- MQCONN Connect to a QMGR
- MQOPEN Open a queue
- MQCLOSE Close a queue
- MQDISC Disconnect from a QMGR

WebSphere MQ z/OS Programming

MQCONN



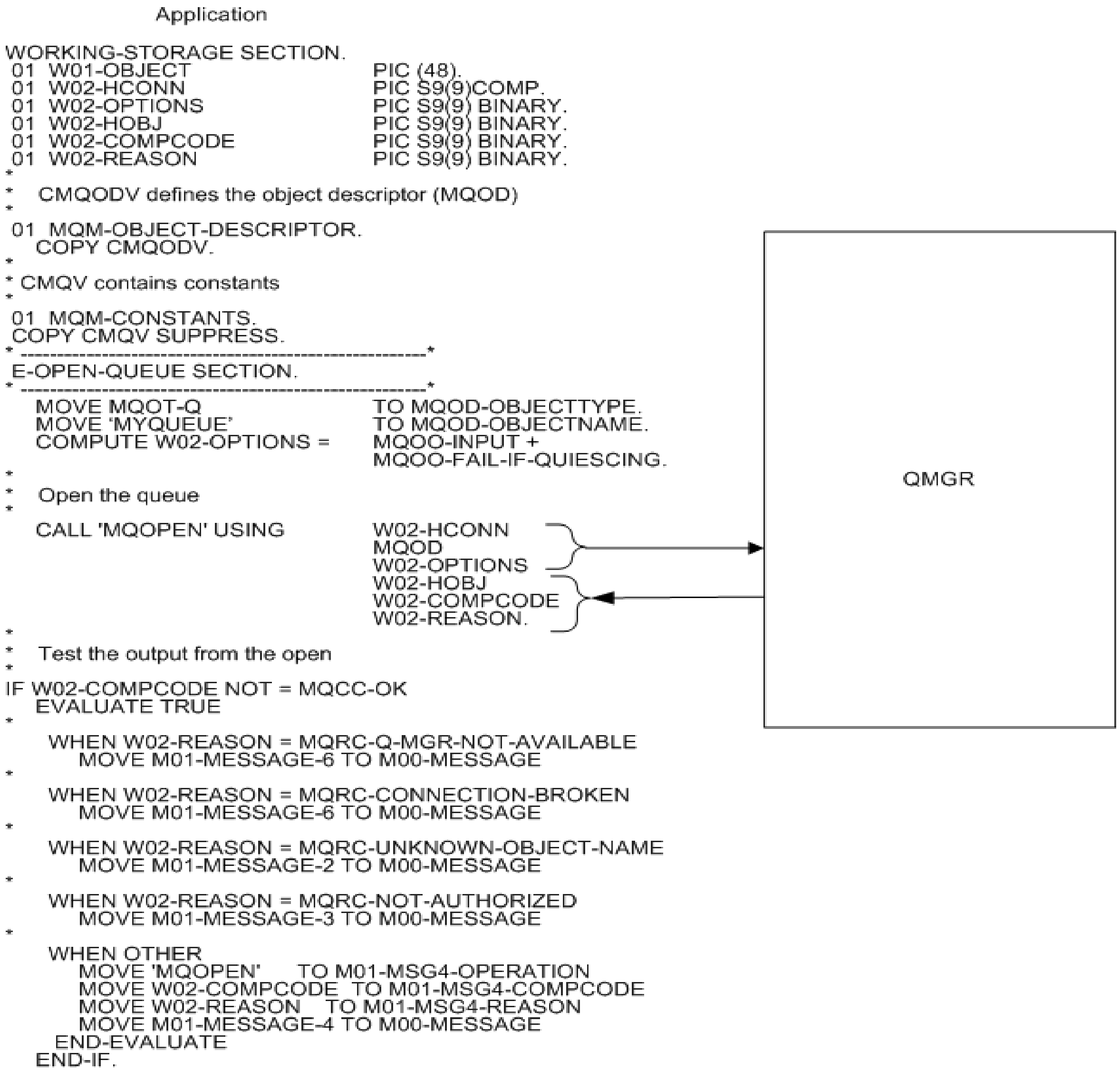
```
Working Storage
01  WS-MQ-Fields
    05 W03-QMGRNAME      PIC X(48) VALUE SPACES.
    05 W03-HCONN         PIC S9(9) BINARY.
    05 W03-COMPCODE      PIC S9(9) BINARY.
    05 W03-REASON        PIC S9(9) BINARY.

Procedure division
* Separate into the relevant fields data passed
* in the PARM statement
*
    UNSTRING PARM-STRING
             DELIMITED BY ALL ','
             INTO
                 W02-MQM
                 W02-OBJECT.

* Connect to the specified queue manager.
*
    CALL 'MQCONN' USING
                  W02-MQM
                  W03-HCONN
                  W03-COMPCODE
                  W03-REASON.
*
* Test the output of the connect call. If the call
* fails, print an error message showing the
* completion code and reason code.
*
     IF (W03-COMPCODE NOT = MQCC-OK)
        THEN
                   :
     END-IF.
```

WebSphere MQ z/OS Programming

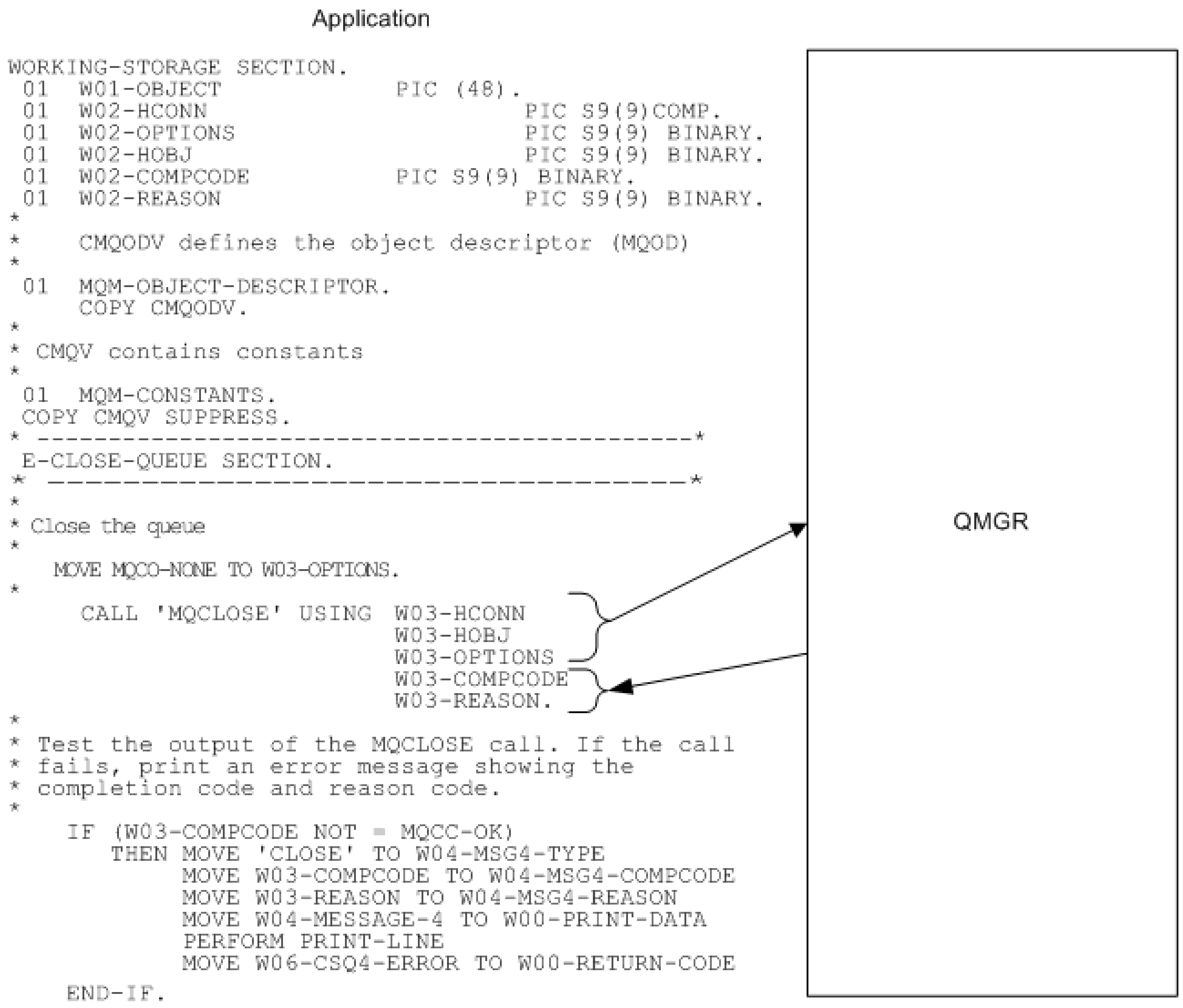
MQOPEN



```
WORKING-STORAGE SECTION.
 01  W01-OBJECT                          PIC (48).
 01  W02-HCONN                           PIC S9(9)COMP.
 01  W02-OPTIONS                         PIC S9(9) BINARY.
 01  W02-HOBJ                            PIC S9(9) BINARY.
 01  W02-COMPCODE                        PIC S9(9) BINARY.
 01  W02-REASON                          PIC S9(9) BINARY.
*
*    CMQODV defines the object descriptor (MQOD)
*
 01  MQM-OBJECT-DESCRIPTOR.
     COPY CMQODV.
*
* CMQV contains constants
*
 01  MQM-CONSTANTS.
 COPY CMQV SUPPRESS.
* ---------------------------------------------------------------*
 E-OPEN-QUEUE SECTION.
* ---------------------------------------------------------------*
     MOVE MQOT-Q                         TO MQOD-OBJECTTYPE.
     MOVE 'MYQUEUE'                      TO MQOD-OBJECTNAME.
     COMPUTE W02-OPTIONS =               MQOO-INPUT +
                                         MQOO-FAIL-IF-QUIESCING.
*
*    Open the queue
*
     CALL 'MQOPEN' USING                 W02-HCONN
                                         MQOD
                                         W02-OPTIONS
                                         W02-HOBJ
                                         W02-COMPCODE
                                         W02-REASON.
*
*    Test the output from the open
*
IF W02-COMPCODE NOT = MQCC-OK
     EVALUATE TRUE
*
       WHEN W02-REASON = MQRC-Q-MGR-NOT-AVAILABLE
          MOVE M01-MESSAGE-6 TO M00-MESSAGE
*
       WHEN W02-REASON = MQRC-CONNECTION-BROKEN
          MOVE M01-MESSAGE-6 TO M00-MESSAGE
*
       WHEN W02-REASON = MQRC-UNKNOWN-OBJECT-NAME
          MOVE M01-MESSAGE-2 TO M00-MESSAGE
*
       WHEN W02-REASON = MQRC-NOT-AUTHORIZED
          MOVE M01-MESSAGE-3 TO M00-MESSAGE
*
       WHEN OTHER
          MOVE 'MQOPEN'      TO M01-MSG4-OPERATION
          MOVE W02-COMPCODE  TO M01-MSG4-COMPCODE
          MOVE W02-REASON    TO M01-MSG4-REASON
          MOVE M01-MESSAGE-4 TO M00-MESSAGE
       END-EVALUATE
     END-IF.
```

WebSphere MQ z/OS Programming

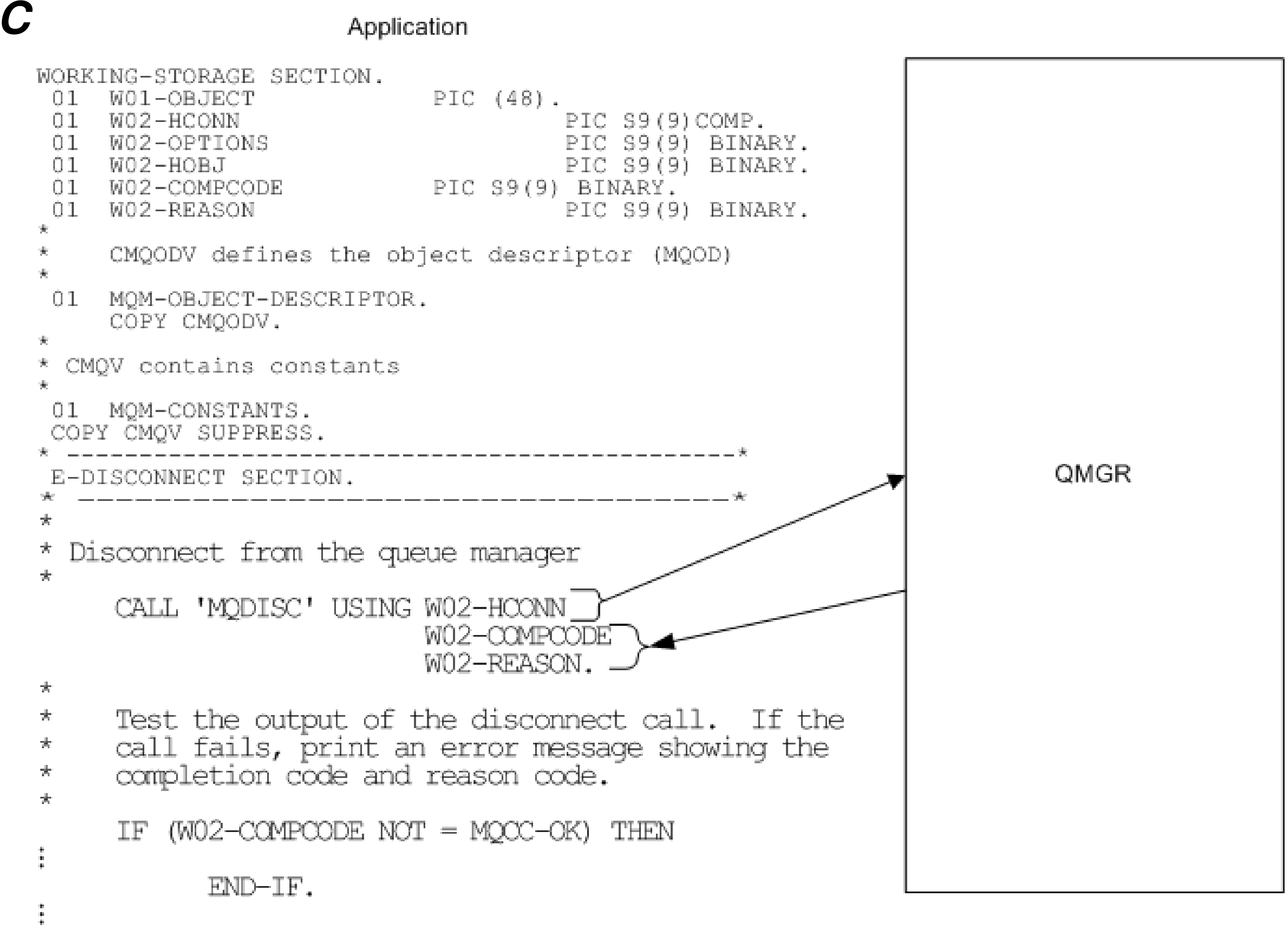
MQCLOSE



```
WORKING-STORAGE SECTION.
 01  W01-OBJECT                PIC (48).
 01  W02-HCONN                           PIC S9(9)COMP.
 01  W02-OPTIONS                         PIC S9(9) BINARY.
 01  W02-HOBJ                            PIC S9(9) BINARY.
 01  W02-COMPCODE              PIC S9(9) BINARY.
 01  W02-REASON                          PIC S9(9) BINARY.
*
*    CMQODV defines the object descriptor (MQOD)
*
 01  MQM-OBJECT-DESCRIPTOR.
     COPY CMQODV.
*
* CMQV contains constants
*
 01  MQM-CONSTANTS.
 COPY CMQV SUPPRESS.
* ---------------------------------------------*
 E-CLOSE-QUEUE SECTION.
* --------------------------------------------*
*
* Close the queue
*
    MOVE MQCO-NONE TO W03-OPTIONS.
*
     CALL 'MQCLOSE' USING  W03-HCONN
                           W03-HOBJ
                           W03-OPTIONS
                           W03-COMPCODE
                           W03-REASON.
*
* Test the output of the MQCLOSE call. If the call
* fails, print an error message showing the
* completion code and reason code.
*
    IF (W03-COMPCODE NOT = MQCC-OK)
       THEN MOVE 'CLOSE' TO W04-MSG4-TYPE
            MOVE W03-COMPCODE TO W04-MSG4-COMPCODE
            MOVE W03-REASON TO W04-MSG4-REASON
            MOVE W04-MESSAGE-4 TO W00-PRINT-DATA
            PERFORM PRINT-LINE
            MOVE W06-CSQ4-ERROR TO W00-RETURN-CODE
    END-IF.
```

WebSphere MQ z/OS Programming

MQDISC



```
WORKING-STORAGE SECTION.
 01  W01-OBJECT             PIC (48).
 01  W02-HCONN                       PIC S9(9)COMP.
 01  W02-OPTIONS                     PIC S9(9) BINARY.
 01  W02-HOBJ                        PIC S9(9) BINARY.
 01  W02-COMPCODE           PIC S9(9) BINARY.
 01  W02-REASON                      PIC S9(9) BINARY.
*
*    CMQODV defines the object descriptor (MQOD)
*
 01  MQM-OBJECT-DESCRIPTOR.
     COPY CMQODV.
*
* CMQV contains constants
*
 01  MQM-CONSTANTS.
 COPY CMQV SUPPRESS.
* ----------------------------------------------*
 E-DISCONNECT SECTION.
*  -------------------------------------------*
*
* Disconnect from the queue manager
*
     CALL 'MQDISC' USING W02-HCONN
                         W02-COMPCODE
                         W02-REASON.
*
*    Test the output of the disconnect call.  If the
*    call fails, print an error message showing the
*    completion code and reason code.
*
     IF (W02-COMPCODE NOT = MQCC-OK) THEN
:
         END-IF.
:
```

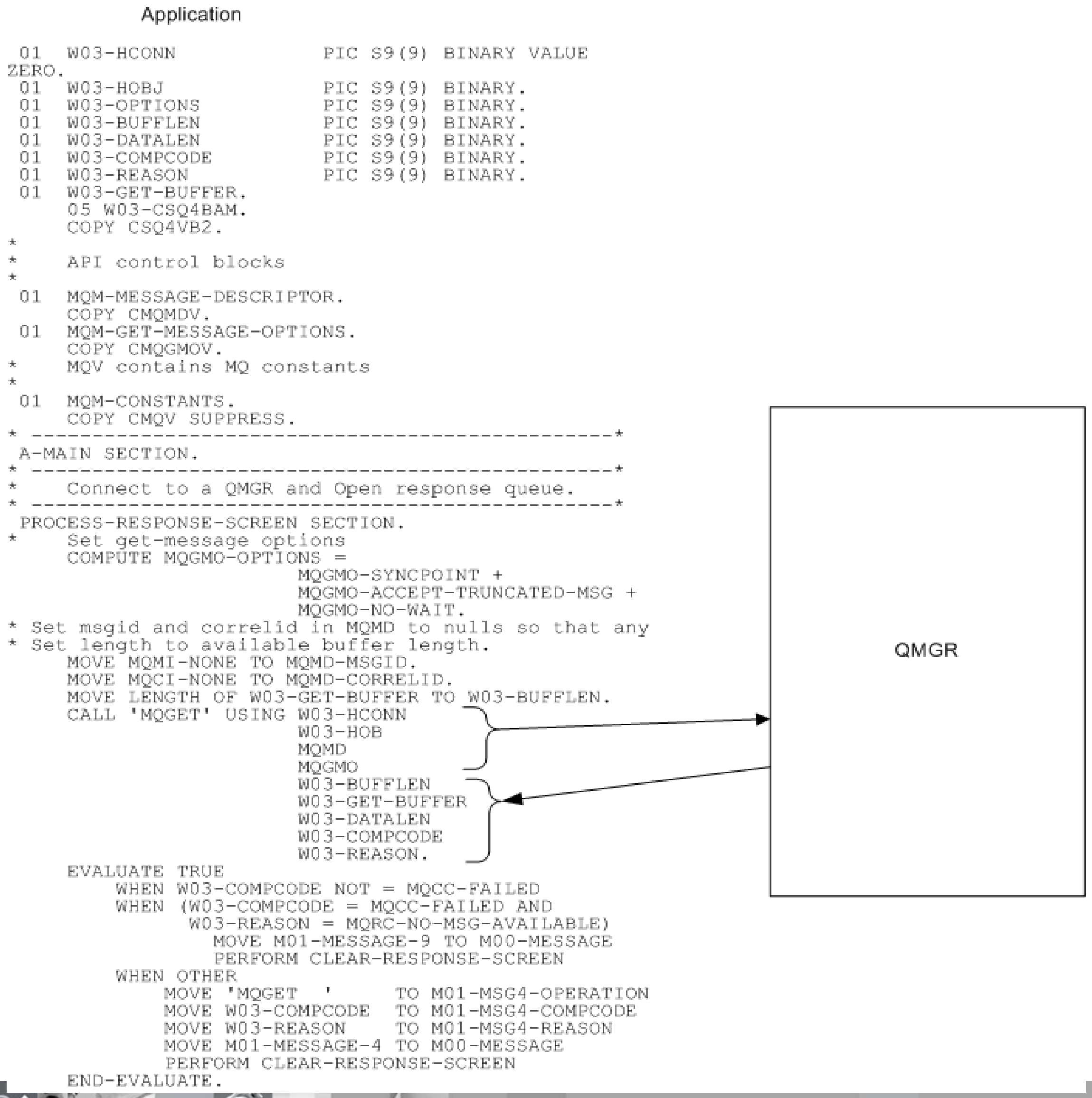
WebSphere MQ z/OS Programming

Messaging Calls

- Functions
  - MQGET – Gets a messages from a Queue
  - MQPUT – Write a message to a queue
  - MQPUT1 – Open, Write, Close a queue for a message
- Features
  - Destructive Gets
  - Browse
  - Transactional
  - Targeted Reads by Identifier

WebSphere MQ z/OS Programming

MQGET

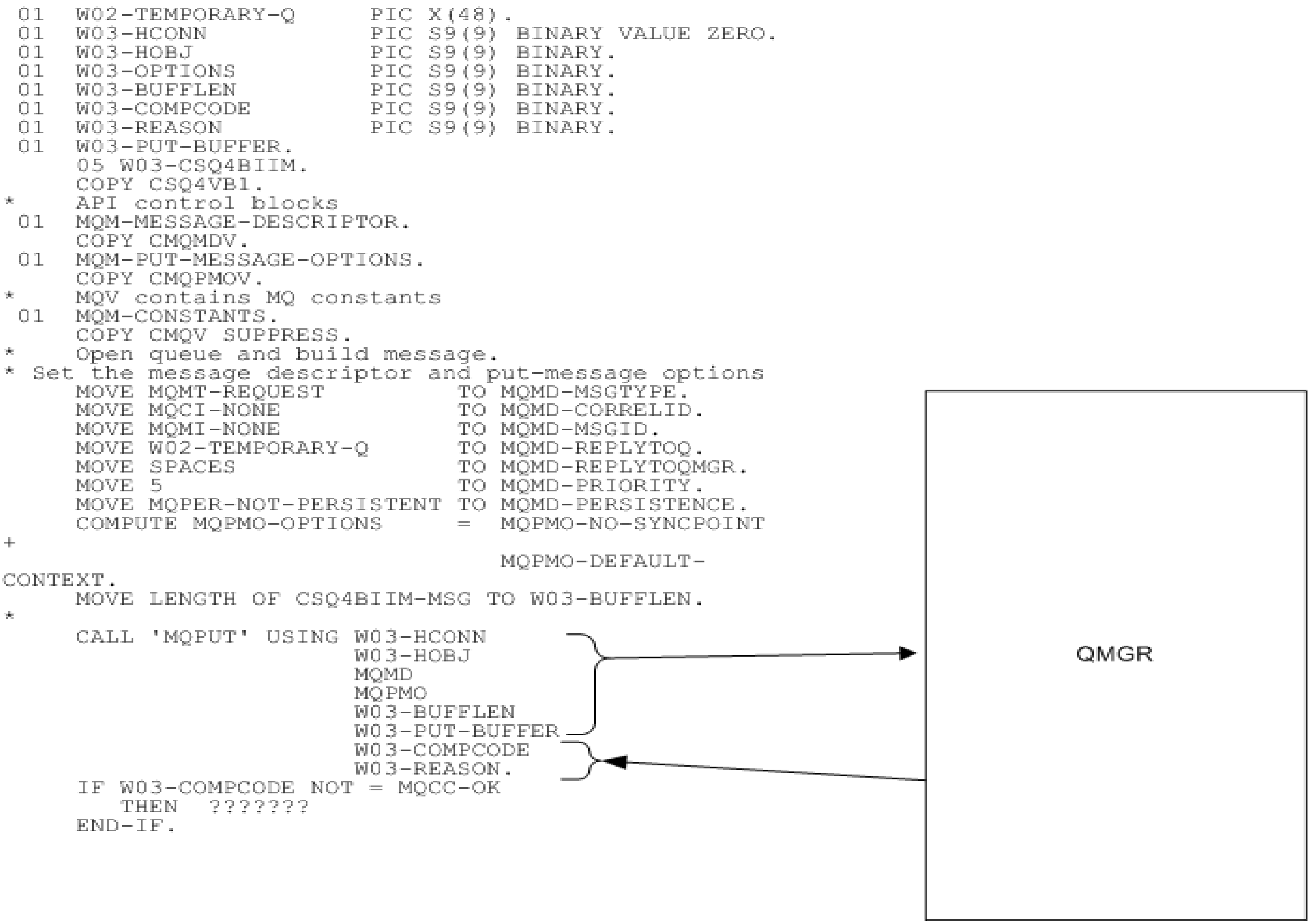


```
 01  W03-HCONN            PIC S9(9) BINARY VALUE
ZERO.
 01  W03-HOBJ             PIC S9(9) BINARY.
 01  W03-OPTIONS          PIC S9(9) BINARY.
 01  W03-BUFFLEN          PIC S9(9) BINARY.
 01  W03-DATALEN          PIC S9(9) BINARY.
 01  W03-COMPCODE         PIC S9(9) BINARY.
 01  W03-REASON           PIC S9(9) BINARY.
 01  W03-GET-BUFFER.
     05 W03-CSQ4BAM.
     COPY CSQ4VB2.
*
*    API control blocks
*
 01  MQM-MESSAGE-DESCRIPTOR.
     COPY CMQMDV.
 01  MQM-GET-MESSAGE-OPTIONS.
     COPY CMQGMOV.
*    MQV contains MQ constants
*
 01  MQM-CONSTANTS.
     COPY CMQV SUPPRESS.
* ------------------------------------------------*
 A-MAIN SECTION.
* ------------------------------------------------*
*    Connect to a QMGR and Open response queue.
* ------------------------------------------------*
 PROCESS-RESPONSE-SCREEN SECTION.
*    Set get-message options
     COMPUTE MQGMO-OPTIONS =
                        MQGMO-SYNCPOINT +
                        MQGMO-ACCEPT-TRUNCATED-MSG +
                        MQGMO-NO-WAIT.
* Set msgid and correlid in MQMD to nulls so that any
* Set length to available buffer length.
     MOVE MQMI-NONE TO MQMD-MSGID.
     MOVE MQCI-NONE TO MQMD-CORRELID.
     MOVE LENGTH OF W03-GET-BUFFER TO W03-BUFFLEN.
     CALL 'MQGET' USING W03-HCONN
                        W03-HOB
                        MQMD
                        MQGMO
                        W03-BUFFLEN
                        W03-GET-BUFFER
                        W03-DATALEN
                        W03-COMPCODE
                        W03-REASON.
     EVALUATE TRUE
         WHEN W03-COMPCODE NOT = MQCC-FAILED
         WHEN (W03-COMPCODE = MQCC-FAILED AND
               W03-REASON = MQRC-NO-MSG-AVAILABLE)
                 MOVE M01-MESSAGE-9 TO M00-MESSAGE
                 PERFORM CLEAR-RESPONSE-SCREEN
         WHEN OTHER
             MOVE 'MQGET   '     TO M01-MSG4-OPERATION
             MOVE W03-COMPCODE  TO M01-MSG4-COMPCODE
             MOVE W03-REASON    TO M01-MSG4-REASON
             MOVE M01-MESSAGE-4 TO M00-MESSAGE
             PERFORM CLEAR-RESPONSE-SCREEN
     END-EVALUATE.
```

WebSphere MQ z/OS Programming

MQPUT

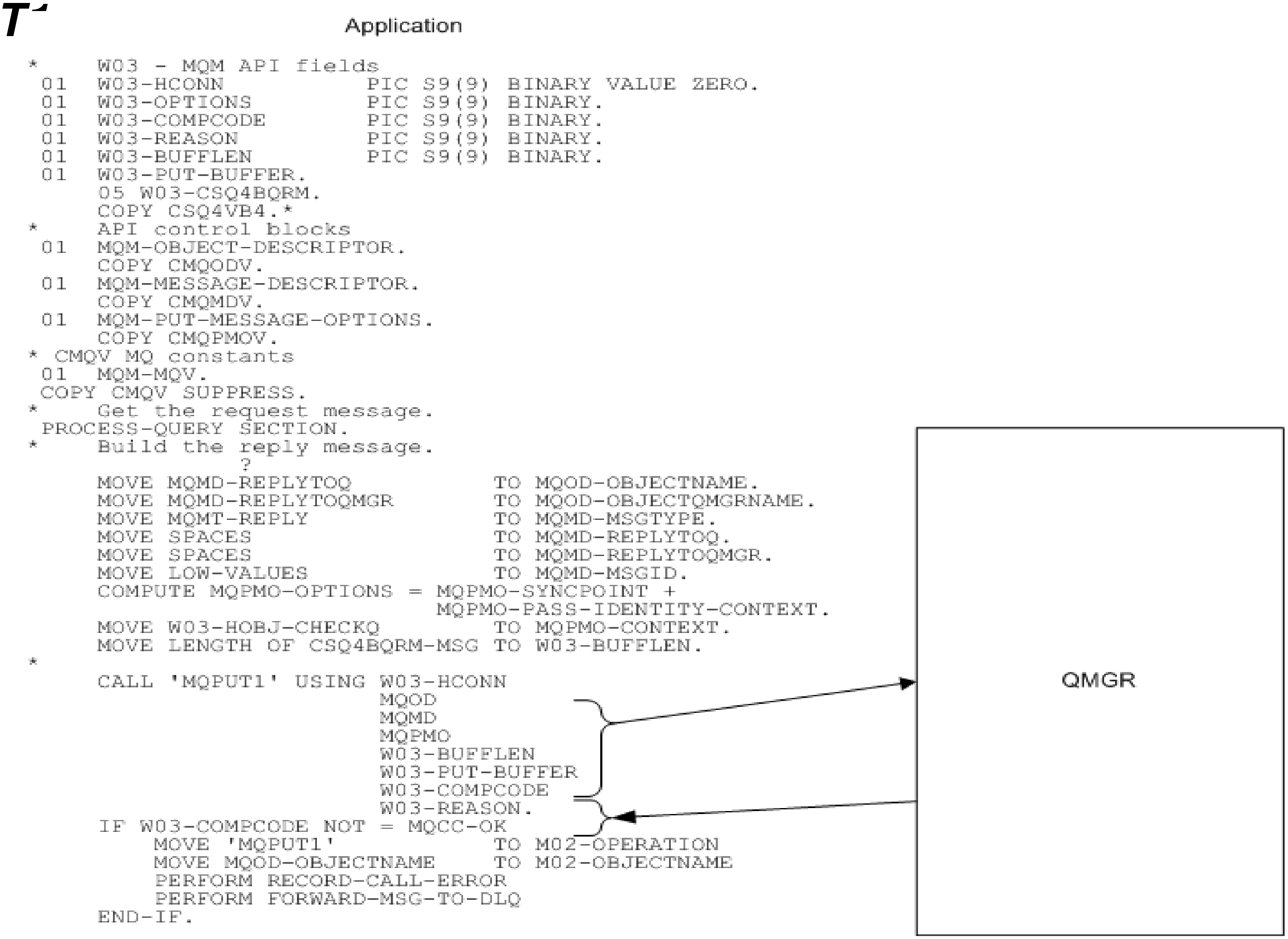
Application



```
 01  W02-TEMPORARY-Q     PIC X(48).
 01  W03-HCONN           PIC S9(9) BINARY VALUE ZERO.
 01  W03-HOBJ            PIC S9(9) BINARY.
 01  W03-OPTIONS         PIC S9(9) BINARY.
 01  W03-BUFFLEN         PIC S9(9) BINARY.
 01  W03-COMPCODE        PIC S9(9) BINARY.
 01  W03-REASON          PIC S9(9) BINARY.
 01  W03-PUT-BUFFER.
     05 W03-CSQ4BIIM.
     COPY CSQ4VB1.
*    API control blocks
 01  MQM-MESSAGE-DESCRIPTOR.
     COPY CMQMDV.
 01  MQM-PUT-MESSAGE-OPTIONS.
     COPY CMQPMOV.
*    MQV contains MQ constants
 01  MQM-CONSTANTS.
     COPY CMQV SUPPRESS.
*    Open queue and build message.
* Set the message descriptor and put-message options
     MOVE MQMT-REQUEST          TO MQMD-MSGTYPE.
     MOVE MQCI-NONE             TO MQMD-CORRELID.
     MOVE MQMI-NONE             TO MQMD-MSGID.
     MOVE W02-TEMPORARY-Q       TO MQMD-REPLYTOQ.
     MOVE SPACES                TO MQMD-REPLYTOQMGR.
     MOVE 5                     TO MQMD-PRIORITY.
     MOVE MQPER-NOT-PERSISTENT TO MQMD-PERSISTENCE.
     COMPUTE MQPMO-OPTIONS      =  MQPMO-NO-SYNCPOINT
+
                                  MQPMO-DEFAULT-
CONTEXT.
     MOVE LENGTH OF CSQ4BIIM-MSG TO W03-BUFFLEN.
*
     CALL 'MQPUT' USING W03-HCONN
                        W03-HOBJ
                        MQMD
                        MQPMO
                        W03-BUFFLEN
                        W03-PUT-BUFFER
                        W03-COMPCODE
                        W03-REASON.
     IF W03-COMPCODE NOT = MQCC-OK
        THEN   ???????
     END-IF.
```

WebSphere MQ z/OS Programming

MQPUT1



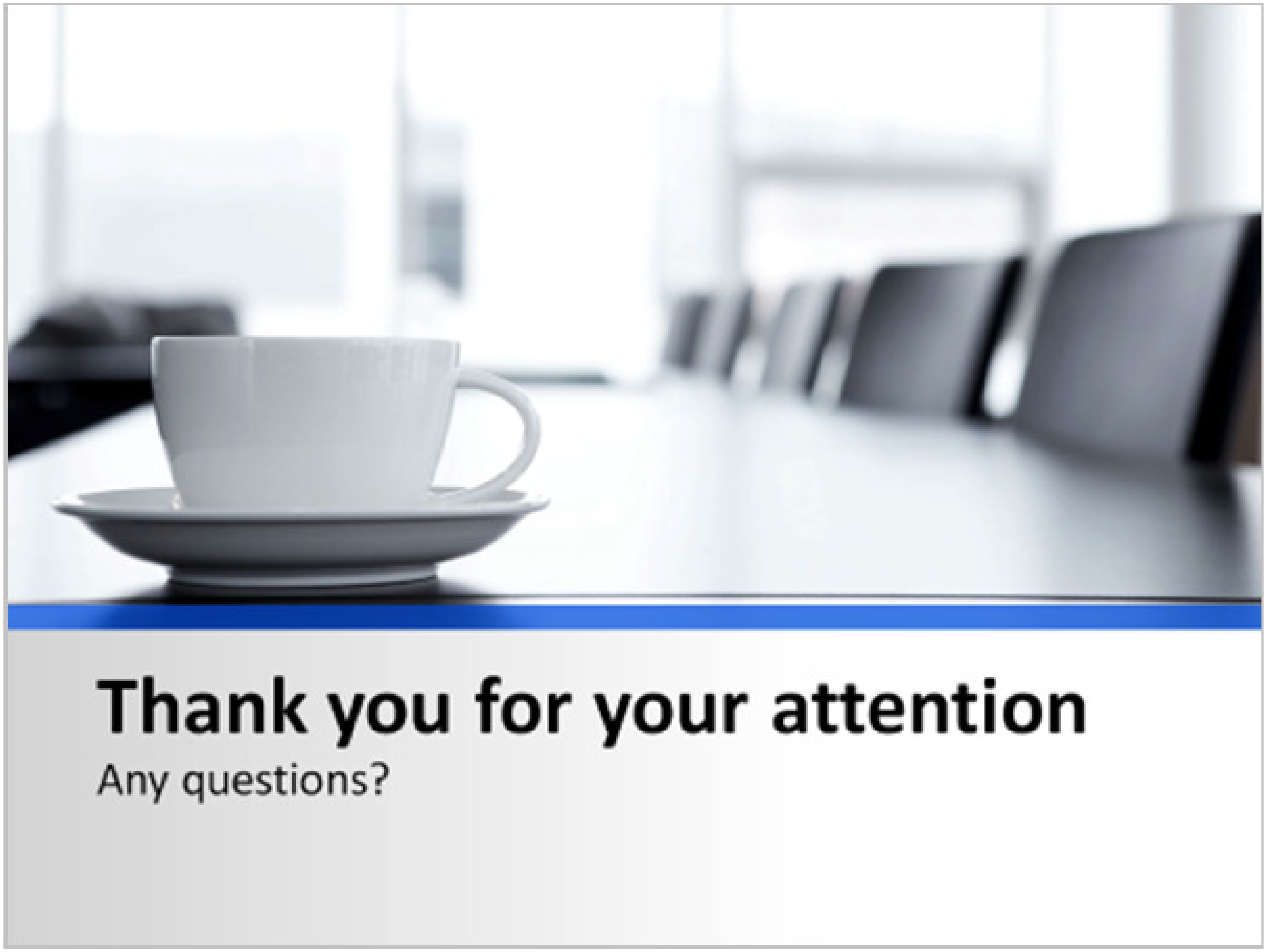
```
*     W03 - MQM API fields
 01   W03-HCONN              PIC S9(9) BINARY VALUE ZERO.
 01   W03-OPTIONS            PIC S9(9) BINARY.
 01   W03-COMPCODE           PIC S9(9) BINARY.
 01   W03-REASON             PIC S9(9) BINARY.
 01   W03-BUFFLEN            PIC S9(9) BINARY.
 01   W03-PUT-BUFFER.
      05 W03-CSQ4BQRM.
      COPY CSQ4VB4.*
*     API control blocks
 01   MQM-OBJECT-DESCRIPTOR.
      COPY CMQODV.
 01   MQM-MESSAGE-DESCRIPTOR.
      COPY CMQMDV.
 01   MQM-PUT-MESSAGE-OPTIONS.
      COPY CMQPMOV.
* CMQV MQ constants
 01   MQM-MQV.
 COPY CMQV SUPPRESS.
*     Get the request message.
 PROCESS-QUERY SECTION.
*     Build the reply message.
                ?
      MOVE MQMD-REPLYTOQ           TO MQOD-OBJECTNAME.
      MOVE MQMD-REPLYTOQMGR        TO MQOD-OBJECTQMGRNAME.
      MOVE MQMT-REPLY              TO MQMD-MSGTYPE.
      MOVE SPACES                  TO MQMD-REPLYTOQ.
      MOVE SPACES                  TO MQMD-REPLYTOQMGR.
      MOVE LOW-VALUES              TO MQMD-MSGID.
      COMPUTE MQPMO-OPTIONS = MQPMO-SYNCPOINT +
                              MQPMO-PASS-IDENTITY-CONTEXT.
      MOVE W03-HOBJ-CHECKQ         TO MQPMO-CONTEXT.
      MOVE LENGTH OF CSQ4BQRM-MSG TO W03-BUFFLEN.
*
      CALL 'MQPUT1' USING W03-HCONN
                          MQOD
                          MQMD
                          MQPMO
                          W03-BUFFLEN
                          W03-PUT-BUFFER
                          W03-COMPCODE
                          W03-REASON.
      IF W03-COMPCODE NOT = MQCC-OK
          MOVE 'MQPUT1'            TO M02-OPERATION
          MOVE MQOD-OBJECTNAME     TO M02-OBJECTNAME
          PERFORM RECORD-CALL-ERROR
          PERFORM FORWARD-MSG-TO-DLQ
      END-IF.
```

WebSphere MQ z/OS Programming

Additional MQI Calls

MQBACK – Back out changes
MQBEGIN – Begin unit of work
MQBUFMH – Convert buffer into message handle
MQCB – Manage callback
MQCB_FUNCTION – Callback function
MQCMIT – Commit changes
MQCONNX – Connect queue manager (extended)
MQCRTMH – Create message handle
MQCTL – Control callbacks
MQDLTMH – Delete message handle
MQDLTMP – Delete message property
MQINQ – Inquire object attributes
MQINQMP – Inquire message property
MQMHBUF – Convert message handle into buffer
MQSET – Set object attributes
MQSETMP – Set message property
MQSTAT – Retrieve status information
MQSUB – Register subscription
MQSUBRQ – Subscription request

WebSphere MQ z/OS Programming



Thank you for your attention

Any questions?