Introduction to the MQI

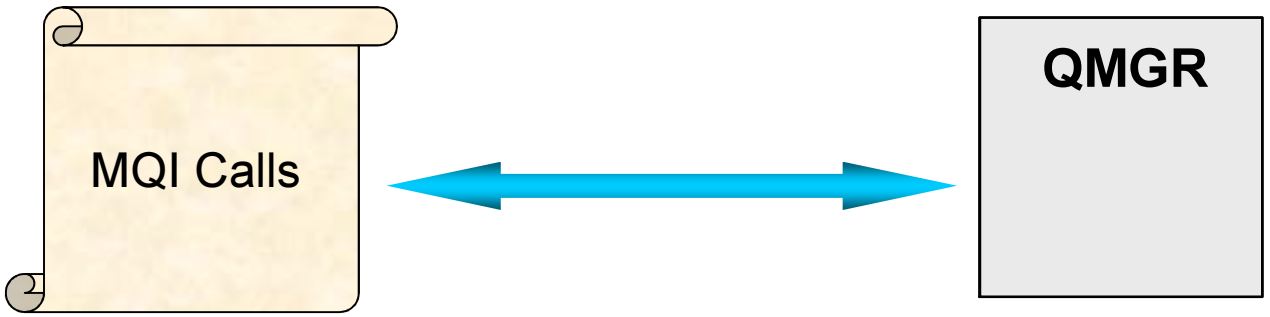
Morag Hughson – morag@mqgem.com

MQGem Software

COBOL ***C***

Agenda

- **MQI Concepts**
- **MQI Structures & Datatypes**
- **Basic MQI walkthrough**
  - With Demonstrations
  - A number of verbs we do not cover
    - MQCMIT, MQBACK, MQINQ, MQSET etc



Languages

- **Procedural (MQI)**
  - C
  - COBOL
  - Visual Basic
  - RPG
  - PL/1
  - Assembler
  - TAL
- **Object-Oriented (Classes)**
  - Java
  - JMS/XMS
  - C++
  - .NET languages
  - ActiveX (MQAX)
  - Perl

Interface

- **Simple ‘handle’ based interface**
  - Returned handle passed to subsequent call
- **Each verb returns**
  - Completion Code
    - `MQCC-OK` 0
    - `MQCC-WARNING` 1
    - `MQCC-FAILED` 2
  - Reason Code
    - `MQRC-xxxxxxx` 2xxx
    - `MQRC-NONE` 0
- **Make sure you check the reason codes!**

Data Structures

- **Programmers should be familiar with:**

Name	Description	Purpose
MQMD	**Message Descriptor**	**Attributes associated with a message**
MQOD	**Object Descriptor**	**Describes what object to open**
MQSD	**Subscription Descriptor**	**Describes what to subscribe to**
MQPMO	**Put Message Options**	**Describes how a message should be put**
MQGMO	**Get Message Options**	**Describes how a message should be got**

Data Structure Tips

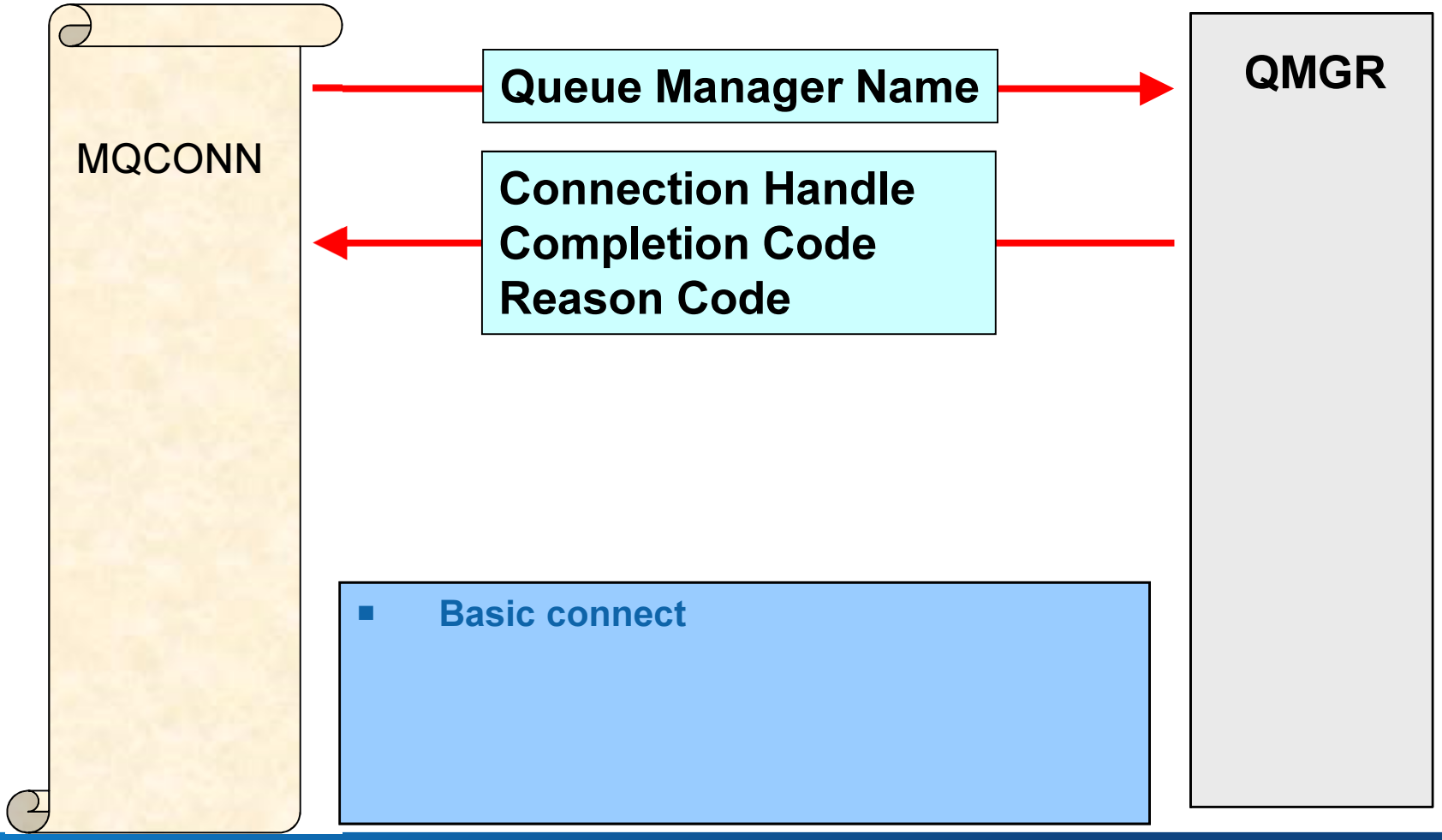
- **Use copybooks with value initialisers**
  - `01  MESSAGE-DESCRIPTOR.`
  - `    COPY CMQMDV.`
  - Initialise to version 1
- **Structures are versioned**
  - Set the minimum version you need
    - `MOVE MQMD-VERSION-2 to MQMD-VERSION.`
  - Don't use current version
    - `MOVE MQMD-CURRENT-VERSION to MQMD-VERSION.`
- **Bear in mind that some structures are input/output**
  - May need to reset values for subsequent call
    - Eg. MsgId & CorrelId fields of MQMD on MQGET call

MQ Elementary Data Types

- The main MQI data types

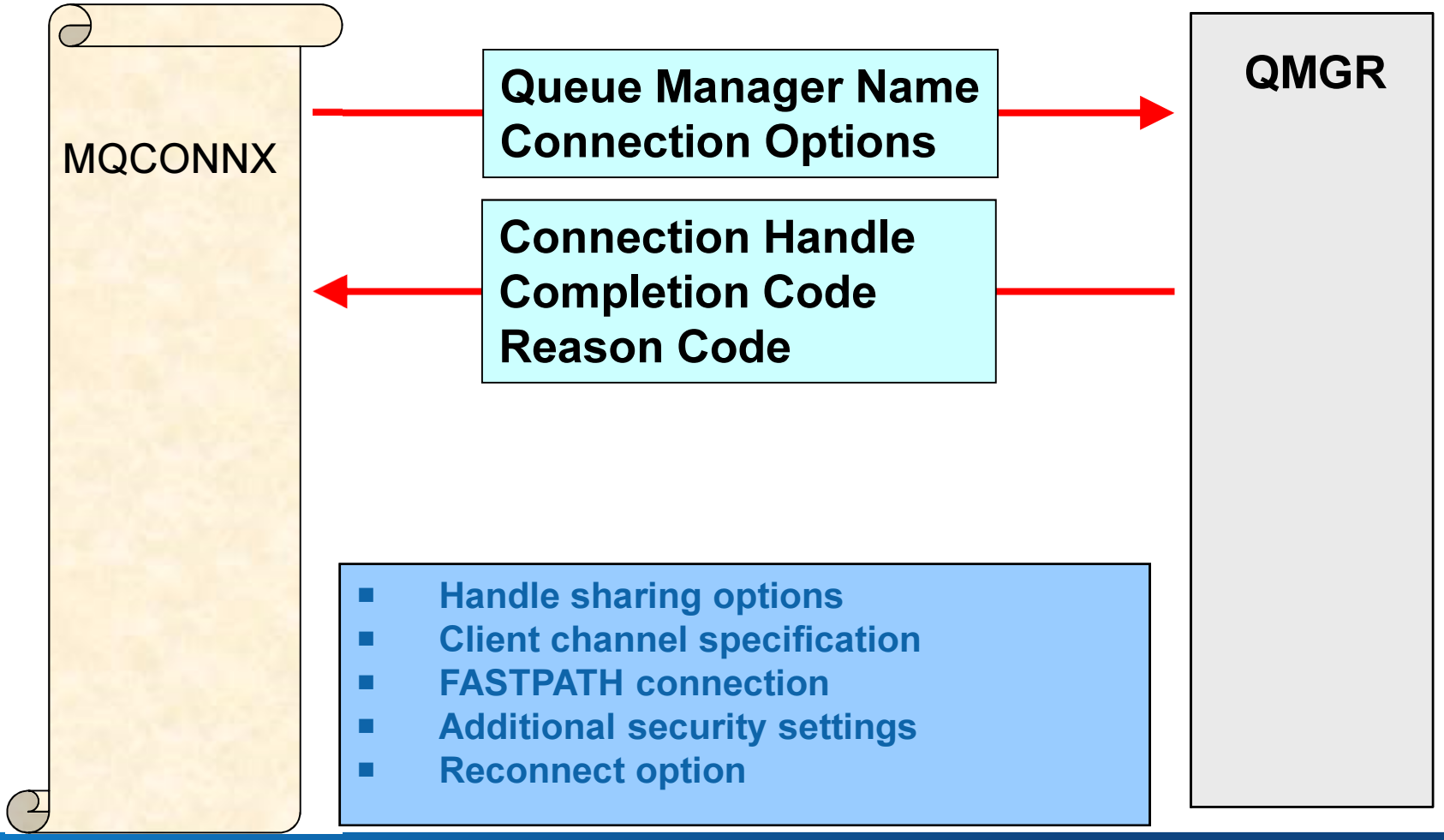
DataType	Purpose
MQHCONN	4-byte Connection Handle
MQHOBJ	4-byte Object Handle
MQLONG	4-byte binary integer
MQPTR	Pointer
MQCHARn	A series of “n” bytes containing character data
MQBYTEn	A series of “n” bytes containing binary data
MQCHARV	Variable length string

Connect



- Basic connect

Connect with extended options



Connecting

- **MQCONNX**
  - Don't hardcode QM name
  - Always check reason codes
- **Connections options**
  - Connection not thread specific
  - Client reconnect

```
 CALL 'MQCONNX'
  USING QM-NAME, CONNECT-OPTIONS,
  HCONN, MQCC, REASON.

 IF MQCC IS NOT EQUAL TO MQCC-OK
* report reason and
* stop if it failed
  END-IF.
```

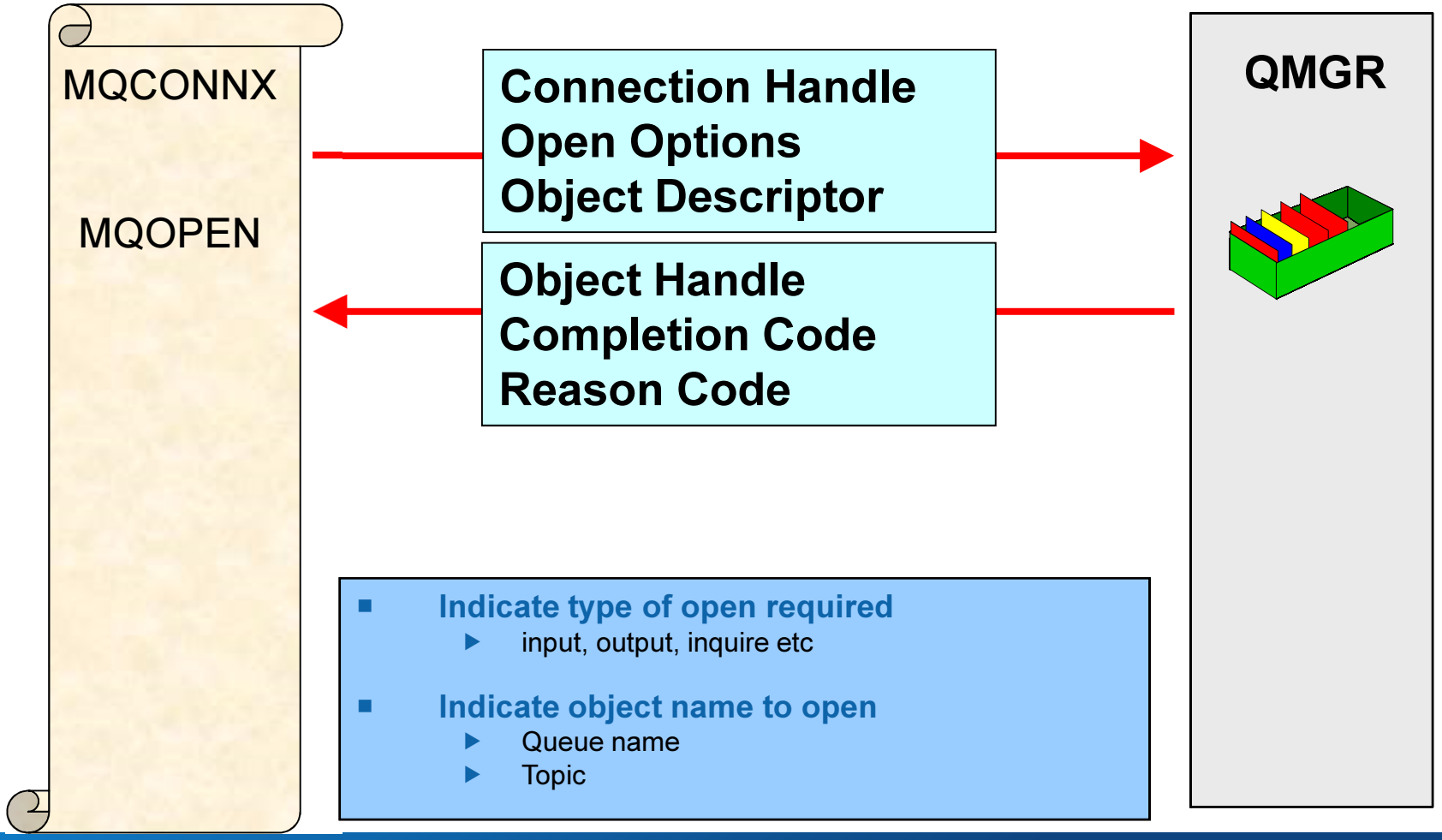
```
05 CONNECTION-OPTIONS.
   COPY CMQCNOV.

ADD MQCNO-HANDLE-SHARE-BLOCK MQCNO-RECONNECT
     GIVING CONNECT-OPTIONS.
```

MQCONN(X) Tips

- **Don't hardcode Queue Manager names**
  - Pass as parameter or configure in INI file
- **Best to use MQCONNX**
  - Has options structure should it be needed
- **Most expensive verb**
  - Don't issue it repeatedly for each request
    - Often problem for OO languages
- **If MQI handle need to be used on different threads**
  - Use connection options to indicate the MQI handle can be shared
  - Choose to block or reject any calls from another thread when handle is in use
- **If reconnecting use exponential back-off with random wait**
  - Try to avoid client storms
- **Can dynamically load MQ libraries if client or local binding**
  - Preferable to shipping two versions of the program

Open a Queue



- **Indicate type of open required**
  - input, output, inquire etc
- **Indicate object name to open**
  - Queue name
  - Topic

Open a queue

- MQOPEN a queue
- OpenOptions
  - MQOO- flags which are required
- MQOD describes a object to open
  - MQOD-OBJECTTYPE
    - MQOT-Q for point-to-point
    - MQOT-TOPIC for publish
  - MQOD-OBJECTSTRING
  - MQOD-OBJECTNAME

```
ADD MQOO-OUTPUT
    MQOO-FAIL-IF-QUIESCING
                      GIVING OPTIONS.
CALL 'MQOPEN'
     USING HCONN,
           OBJECT-DESCRIPTOR,
           OPTIONS,
           Q-HANDLE,
           MQCC,
           REASON.

 IF MQCC IS NOT EQUAL TO MQCC-OK
* report reason and
* stop if it failed
  END-IF.
```

```
01 OBJECT-DESCRIPTOR.
   COPY CMQODV.

MOVE MQOT-Q      TO MQOD-OBJECTTYPE.
MOVE PARM-QNAME  TO MQOD-OBJECTNAME.
```

Object Descriptor (MQOD)

Field	Description	Version
StrucId	Structure identifier	1
Version	Structure version number	
ObjectType	Object type	
ObjectName	Object name	
ObjectQMgrName	Object queue manager name	
DynamicQName	Dynamic queue name	
AlternateUserId	Alternate user identifier	
RecsPresent	Number of object records present	2
KnownDestCount	Number of local queues opened successfully	
UnknownDestCount	Number of remote queues opened successfully	
InvalidDestCount	Number of queues that failed to open	
ObjectRecOffset	Offset of first object record from start of MQOD	
ResponseRecOffset	Offset of first response record from start of MQOD	
ObjectRecPtr	Address of first object record	
ResponseRecPtr	Address of first response record	
AlternateSecurityId	Alternate security identifier	3
ResolvedQName	Resolved queue name	
ResolvedQMgrName	Resolved queue manager name	
ObjectString	Long object name	4
SelectionString	Selection string	
ResObjectString	Resolved long object name	
ResolvedType	Resolved object type	

Open Options

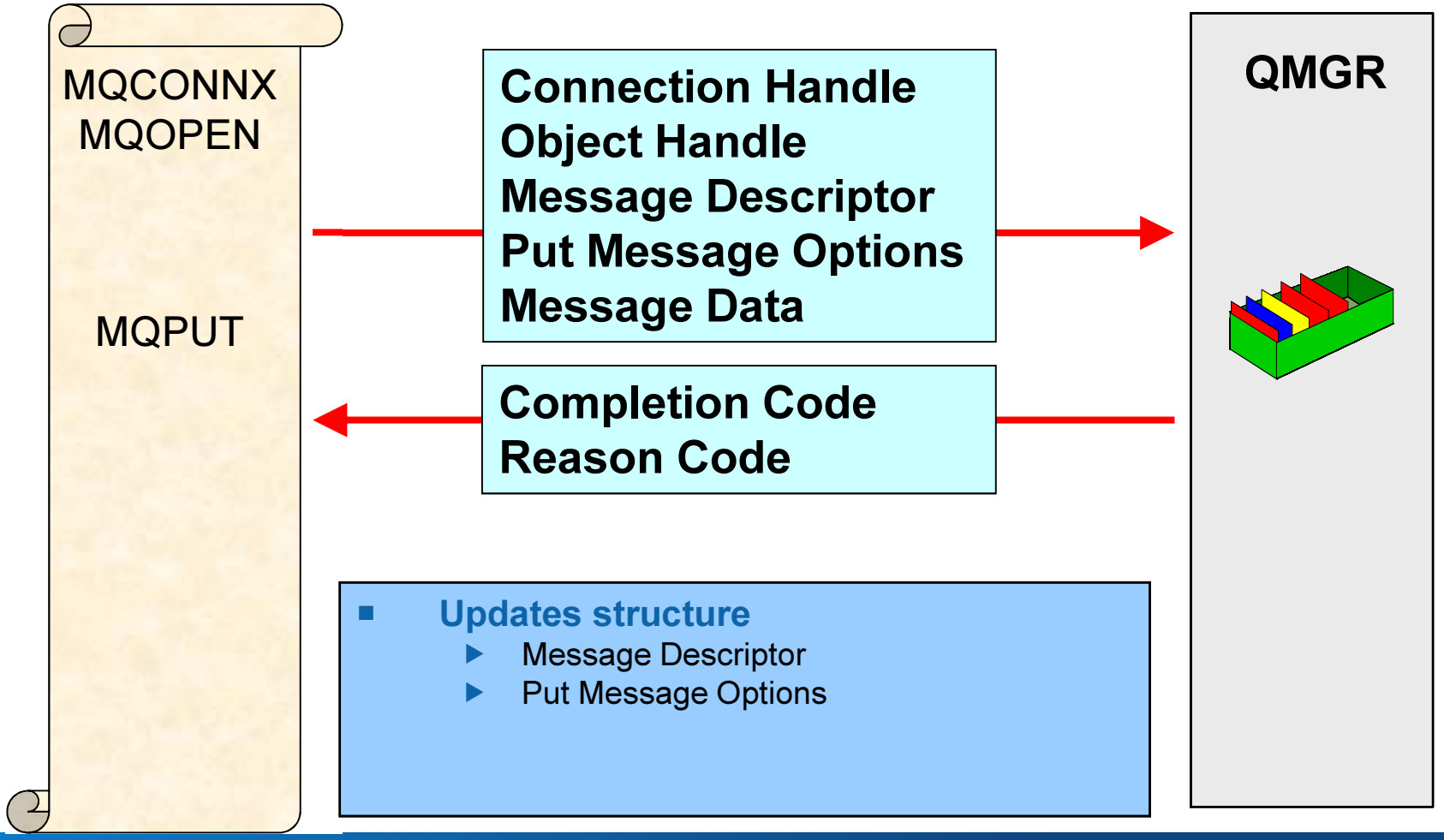
```
10  MQOO-BIND-AS-Q-DEF              PIC S9(9) BINARY VALUE 0.
10  MQOO-READ-AHEAD-AS-Q-DEF        PIC S9(9) BINARY VALUE 0.
10  MQOO-INPUT-AS-Q-DEF             PIC S9(9) BINARY VALUE 1.
10  MQOO-INPUT-SHARED               PIC S9(9) BINARY VALUE 2.
10  MQOO-INPUT-EXCLUSIVE            PIC S9(9) BINARY VALUE 4.
10  MQOO-BROWSE                     PIC S9(9) BINARY VALUE 8.
10  MQOO-OUTPUT                     PIC S9(9) BINARY VALUE 16.
10  MQOO-INQUIRE                    PIC S9(9) BINARY VALUE 32.
10  MQOO-SET                        PIC S9(9) BINARY VALUE 64.
10  MQOO-SAVE-ALL-CONTEXT           PIC S9(9) BINARY VALUE 128.
10  MQOO-PASS-IDENTITY-CONTEXT      PIC S9(9) BINARY VALUE 256.
10  MQOO-PASS-ALL-CONTEXT           PIC S9(9) BINARY VALUE 512.
10  MQOO-SET-IDENTITY-CONTEXT       PIC S9(9) BINARY VALUE 1024.
10  MQOO-SET-ALL-CONTEXT            PIC S9(9) BINARY VALUE 2048.
10  MQOO-ALTERNATE-USER-AUTHORITY   PIC S9(9) BINARY VALUE 4096.
10  MQOO-FAIL-IF-QUIESCING          PIC S9(9) BINARY VALUE 8192.
10  MQOO-BIND-ON-OPEN               PIC S9(9) BINARY VALUE 16384.
10  MQOO-BIND-NOT-FIXED             PIC S9(9) BINARY VALUE 32768.
10  MQOO-CO-OP                      PIC S9(9) BINARY VALUE 131072.
10  MQOO-NO-READ-AHEAD              PIC S9(9) BINARY VALUE 524288.
10  MQOO-READ-AHEAD                 PIC S9(9) BINARY VALUE 1048576.
```

- Options can be added together as required

MQOPEN Tips

- **Try not to hardcode queue/topic names**
- **Try not to open queues exclusively**
  - Will reduce options for workload balancing
- **Use MQPUT1 if only opening queue to put one message**
- **Consider queue cache for common used queues**
  - MQOPEN is relatively expensive – load and security check
- **Use read ahead for performance gain**
  - If client and non-persistent messaging
- **If opening model reply queues**
  - Be aware of how many instances of queues you may be creating
    - Particularly large numbers of clients.
  - May be better to share reply queue

Put a message



- **Updates structure**
  - Message Descriptor
  - Put Message Options

Putting Application

- MQOPEN a queue
- MQPUT a message
  - Simple Hello World message
  - Set message format to string
  - Put of syncpoint

```
ADD MQOO-OUTPUT
    MQOO-FAIL-IF-QUIESCING
                        GIVING OPTIONS.
CALL 'MQOPEN'
     USING HCONN,
           OBJECT-DESCRIPTOR,
           OPTIONS,
           Q-HANDLE,
           MQCC,
           REASON.
CALL 'MQPUT'
     USING HCONN,
           Q-HANDLE,
           MESSAGE-DESCRIPTOR,
           PMOPTIONS,
           BUFFER-LENGTH,
           BUFFER,
           MQCC,
           REASON.
```

```
01 MESSAGE-DESCRIPTOR.
   COPY CMQMDV.
01 PMOPTIONS.
   COPY CMQPMOV.
MOVE MQFMT-STRING TO MQMD-FORMAT.
ADD MQPMO-FAIL-IF-QUIESCING MQPMO-NO-SYNCPOINT GIVING MQPMO-OPTIONS.
MOVE 'Hello World!' TO BUFFER.
MOVE 12 TO BUFFER-LENGTH.
```

Message Descriptor (MQMD)

Field (V1)	Description
StrucId	Structure identifier
Version	Structure version number
Report	Options for report messages
MsgType	Message Type
Expiry	Message lifetime
Feedback	Feedback or reason code
Encoding	Numeric encoding of message data
CodedCharSetId	Character set identifier of message data
Format	Format name of message data
Priority	Message priority
Persistence	Message persistence
MsgId	Message identifier
CorrelId	Correlation identifier
BackoutCount	Backout counter
ReplyToQ	Name of reply queue
ReplyToQMgr	Name of reply queue manager
UserIdentifier	User identifier
AccountingToken	Accounting token
ApplIdentityData	Application data relating to identity
PutApplType	Type of application that put the message
PutApplName	Name of application that put the message
PutDate	Date when message was put
PutTime	Time when message was put
ApplOriginData	Application data relating to origin

Field (V2)	Description
GroupId	Group identifier
MsgSeqNumber	Sequence number of logical message within group
Offset	Offset of data in physical message from start of logical message
MsgFlags	Message flags
OriginalLength	Length of original message

Put Message Options (MQPMO)

Field	Description	Version
StrucId	Structure identifier	1
Version	Structure version number	
Options	Options that control the action of MQPUT and MQPUT1	
Context	Object handle of input queue	
KnownDestCount	Number of messages sent successfully to local queues	
UnknownDestCount	Number of messages sent successfully to remote queue	
InvalidDestCount	Number of messages that could not be sent	
ResolvedQName	Resolved name of destination queue	
ResolvedQMgrName	Resolved name of destination queue manager	
RecsPresent	Number of put messages records or response records present	2
PutMsgRecFields	Flags indicating which MQPMR fields are present	
PutMsgRecOffset	Offset of first put-message records from start of MQPMO	
ResponseRecOffset	Offset of first response record from start of MQPMO	
PutMsgRecPtr	Address of first put message record	
ResponseRecPtr	Address of first response record	
OriginalMsgHandle	Original message handle	3
NewMsgHandle	New message handle	
Action	Type of put being performed and the relationship between the original message and the new message	
PubLevel	Level of subscription targeted by the publication	

Put Options

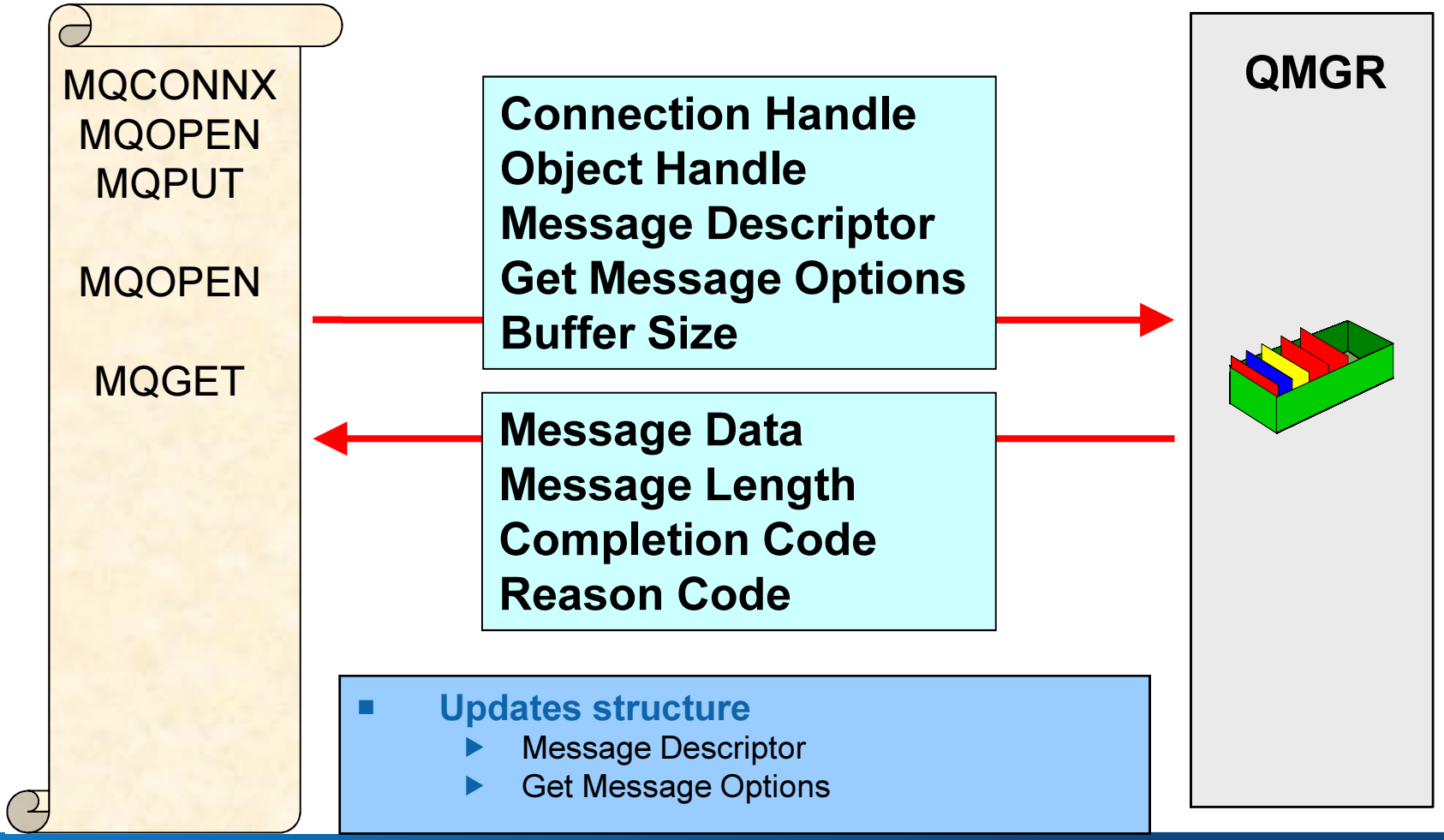
```
10  MQPMO-SYNCPOINT                PIC S9(9) BINARY VALUE 2.
10  MQPMO-NO-SYNCPOINT              PIC S9(9) BINARY VALUE 4.
10  MQPMO-DEFAULT-CONTEXT           PIC S9(9) BINARY VALUE 32.
10  MQPMO-NEW-MSG-ID                PIC S9(9) BINARY VALUE 64.
10  MQPMO-NEW-CORREL-ID             PIC S9(9) BINARY VALUE 128.
10  MQPMO-PASS-IDENTITY-CONTEXT     PIC S9(9) BINARY VALUE 256.
10  MQPMO-PASS-ALL-CONTEXT          PIC S9(9) BINARY VALUE 512.
10  MQPMO-SET-IDENTITY-CONTEXT      PIC S9(9) BINARY VALUE 1024.
10  MQPMO-SET-ALL-CONTEXT           PIC S9(9) BINARY VALUE 2048.
10  MQPMO-ALTERNATE-USER-AUTHORITY  PIC S9(9) BINARY VALUE 4096.
10  MQPMO-FAIL-IF-QUIESCING         PIC S9(9) BINARY VALUE 8192.
10  MQPMO-NO-CONTEXT                PIC S9(9) BINARY VALUE 16384.
10  MQPMO-LOGICAL-ORDER             PIC S9(9) BINARY VALUE 32768.
10  MQPMO-ASYNC-RESPONSE            PIC S9(9) BINARY VALUE 65536.
10  MQPMO-SYNC-RESPONSE             PIC S9(9) BINARY VALUE 131072.
10  MQPMO-RESOLVE-LOCAL-Q           PIC S9(9) BINARY VALUE 262144.
10  MQPMO-WARN-IF-NO-SUBS-MATCHED   PIC S9(9) BINARY VALUE 524288.
10  MQPMO-RETAIN                    PIC S9(9) BINARY VALUE 2097152.
10  MQPMO-MD-FOR-OUTPUT-ONLY        PIC S9(9) BINARY VALUE 8388608.
10  MQPMO-SCOPE-QMGR                PIC S9(9) BINARY VALUE 67108864.
10  MQPMO-SUPPRESS-REPLYTO          PIC S9(9) BINARY VALUE 134217728.
10  MQPMO-NOT-OWN-SUBS              PIC S9(9) BINARY VALUE 268435456.
10  MQPMO-RESPONSE-AS-Q-DEF         PIC S9(9) BINARY VALUE 0.
10  MQPMO-RESPONSE-AS-TOPIC-DEF     PIC S9(9) BINARY VALUE 0.
```

- Options can be added together as required

MQPUT Tips

- **Always use explicit syncpoint setting**
  - Defaults are not the same on z/OS and Distributed
  - Generally
    - Syncpoint when persistent
    - No syncpoint when non-persistent
- **Try not to use extreme message sizes**
  - QM optimized for message 4K – 1MB
- **Consider async put response for performance gain**
  - If on client and sending many non-persistent messages

Get a message



Getting Application

- MQOPEN a queue
- MQGET a message
  - Syncpoint if persistent
  - Always ask for convert
  - Wait for message
    - up to one minute

```
ADD MQOO-INPUT-SHARED
    MQOO-FAIL-IF-QUIESCING
                    GIVING OPTIONS.
CALL 'MQOPEN'
     USING HCONN,
           OBJECT-DESCRIPTOR,
           OPTIONS,
           Q-HANDLE,
           MQCC,
           REASON.

CALL 'MQGET'
     USING HCONN,
           Q-HANDLE,
           MESSAGE-DESCRIPTOR,
           GMOPTIONS,
           BUFFER-LENGTH,
           BUFFER,
           DATA-LENGTH,
           MQCC,
           REASON.
```

```
01 MESSAGE-DESCRIPTOR.
   COPY CMQMDV.
01 GMOPTIONS.
   COPY CMQGMOV.
MOVE 60000 TO MQGMO-WAITINTERVAL.
ADD MQGMO-SYNCPOINT-IF-PERSISTENT MQGMO-CONVERT
    MQGMO-FAIL-IF-QUIESCING       MQGMO-WAIT  GIVING MQGMO-OPTIONS.
```

Get Message Options (MQGMO)

Field	Description	Version
StrucId	Structure identifier	1
Version	Structure version number	
Options	Options that control the action of MQGET	
WaitInterval	Wait Interval	
Signal1	Signal	
Signal2	Signal identifier	
ResolvedQName	Resolved name of destination queue	
MatchOptions	Options controlling selection criteria used for MQGET	2
GroupStatus	Flag indicating whether message retrieved is in a group	
SegmentStatus	Flag indicating whether message retrieved is a segment of a logical message	
Sementation	Flag indicating whether further segmentation is allowed for the message retrieved	
MsgToken	Message token	3
ReturnedLength	Length of message data returned (bytes)	
MsgHandle	The handle to a message that is to be populated with the properties of the message being retrieved from the queue.	4

Get Options

```
10  MQGMO-WAIT                    PIC S9(9) BINARY VALUE 1.
10  MQGMO-NO-WAIT                 PIC S9(9) BINARY VALUE 0.
10  MQGMO-SET-SIGNAL              PIC S9(9) BINARY VALUE 8.
10  MQGMO-FAIL-IF-QUIESCING       PIC S9(9) BINARY VALUE 8192.
10  MQGMO-SYNCPOINT               PIC S9(9) BINARY VALUE 2.
10  MQGMO-SYNCPOINT-IF-PERSISTENT PIC S9(9) BINARY VALUE 4096.
10  MQGMO-NO-SYNCPOINT            PIC S9(9) BINARY VALUE 4.
10  MQGMO-MARK-SKIP-BACKOUT       PIC S9(9) BINARY VALUE 128.
10  MQGMO-BROWSE-FIRST            PIC S9(9) BINARY VALUE 16.
10  MQGMO-BROWSE-NEXT             PIC S9(9) BINARY VALUE 32.
10  MQGMO-BROWSE-MSG-UNDER-CURSOR PIC S9(9) BINARY VALUE 2048.
10  MQGMO-MSG-UNDER-CURSOR        PIC S9(9) BINARY VALUE 256.
10  MQGMO-LOCK                    PIC S9(9) BINARY VALUE 512.
10  MQGMO-UNLOCK                  PIC S9(9) BINARY VALUE 1024.
10  MQGMO-ACCEPT-TRUNCATED-MSG    PIC S9(9) BINARY VALUE 64.
```

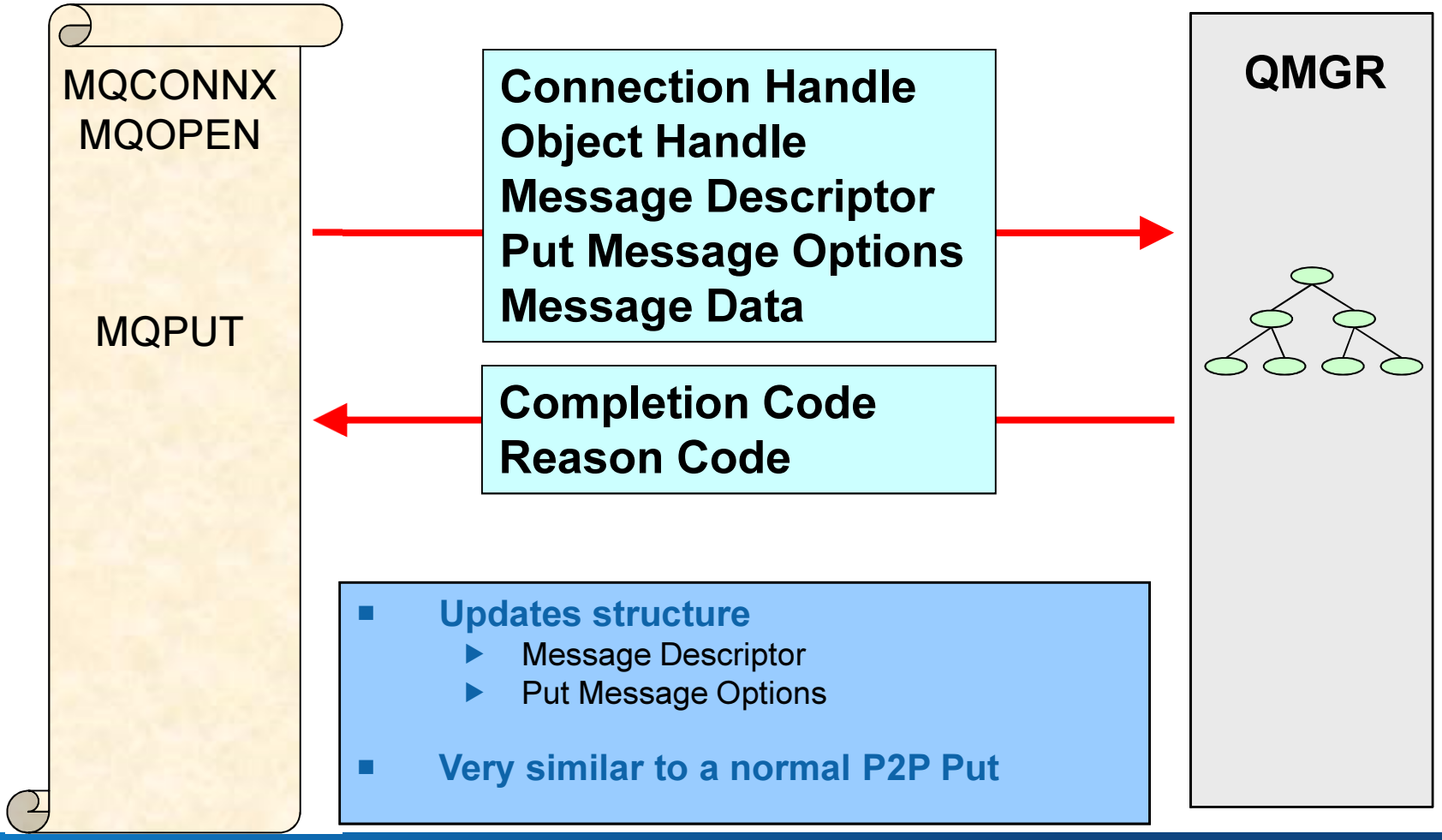
```
10  MQGMO-CONVERT                 PIC S9(9) BINARY VALUE 16384.
10  MQGMO-LOGICAL-ORDER           PIC S9(9) BINARY VALUE 32768.
10  MQGMO-COMPLETE-MSG            PIC S9(9) BINARY VALUE 65536.
10  MQGMO-ALL-MSGS-AVAILABLE      PIC S9(9) BINARY VALUE 131072.
10  MQGMO-ALL-SEGMENTS-AVAILABLE  PIC S9(9) BINARY VALUE 262144.
10  MQGMO-MARK-BROWSE-HANDLE      PIC S9(9) BINARY VALUE 1048576.
10  MQGMO-MARK-BROWSE-CO-OP       PIC S9(9) BINARY VALUE 2097152.
10  MQGMO-UNMARK-BROWSE-CO-OP     PIC S9(9) BINARY VALUE 4194304.
10  MQGMO-UNMARK-BROWSE-HANDLE    PIC S9(9) BINARY VALUE 8388608.
10  MQGMO-UNMARKED-BROWSE-MSG     PIC S9(9) BINARY VALUE 16777216.
10  MQGMO-PROPERTIES-FORCE-MQRFH2 PIC S9(9) BINARY VALUE 33554432.
10  MQGMO-NO-PROPERTIES           PIC S9(9) BINARY VALUE 67108864.
10  MQGMO-PROPERTIES-IN-HANDLE    PIC S9(9) BINARY VALUE 134217728.
10  MQGMO-PROPERTIES-COMPATIBILITY PIC S9(9) BINARY VALUE 268435456.
10  MQGMO-PROPERTIES-AS-Q-DEF     PIC S9(9) BINARY VALUE 0.
```

- Options can be added together as required

MQGET Tips

- **Avoid using default syncpoint setting**
  - Defaults are not the same on z/OS and Distributed
  - Generally
    - MQGMO_SYNCPOINT_IF_PERSISTENT
- **Use MQGMO_FAIL_IF_QUIESCING**
  - Ensure your application ends promptly
- **Generally use MQGMO_CONVERT**
  - Even if you 'think' you don't need it
- **Remember to reset MsgId & CorrelId fields**
  - These fields are used for selection and are returned
- **Handle 'poison message'**
  - Look at BackoutCount in MQMD
- **Consider using MQCB to consume messages instead**
  - Callback semantics, often easier to code

Publish a message



- **Updates structure**
  - Message Descriptor
  - Put Message Options
- **Very similar to a normal P2P Put**

Publishing Application

- MQOPEN a topic
- MQOD describes a topic to publish to
  - MQOD-OBJECTTYPE
    - MQOT-Q for point-to-point
    - MQOT-TOPIC for publish
  - MQOD-OBJECTSTRING
  - MQOD-OBJECTNAME
- MQPUT a message

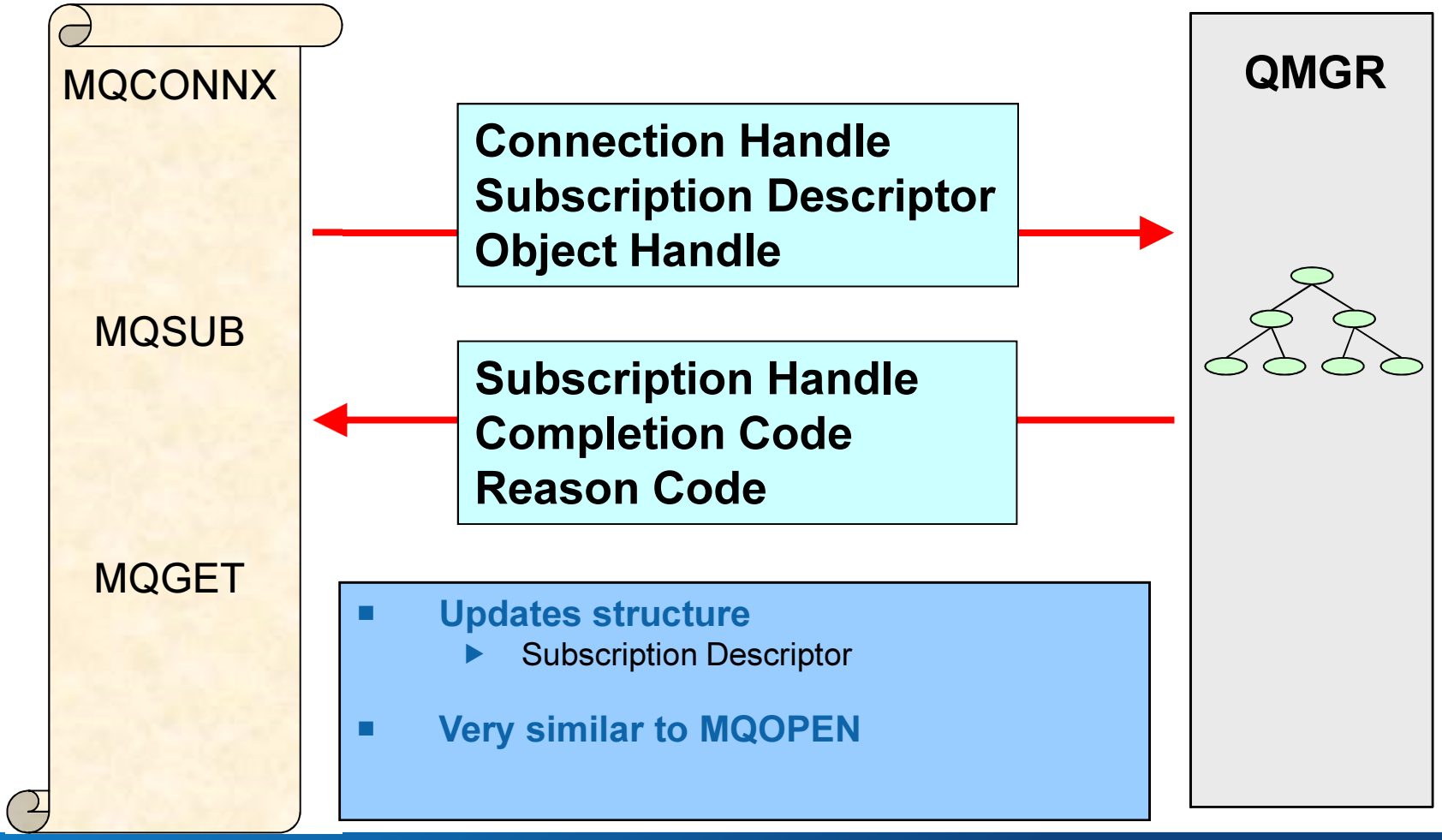
```
ADD MQOO-OUTPUT
    MQOO-FAIL-IF-QUIESCING
                      GIVING OPTIONS.
CALL 'MQOPEN'
     USING HCONN,
           OBJECT-DESCRIPTOR,
           OPTIONS,
           Q-HANDLE,
           MQCC,
           REASON.
CALL 'MQPUT'
     USING HCONN,
           Q-HANDLE,
           MESSAGE-DESCRIPTOR,
           PMOPTIONS,
           BUFFER-LENGTH,
           BUFFER,
           MQCC,
           REASON.
```

```
01 OBJECT-DESCRIPTOR.
   COPY CMQODV.
MOVE MQOT-TOPIC             TO MQOD-OBJECTTYPE.
MOVE MQOD-VERSION-4         TO MQOD-VERSION.
MOVE 'Price/Fruit/Apples'  TO TARGET-TOPIC.
SET  MQOD-OBJECTSTRING-VSPTR TO ADDRESS OF TARGET-TOPIC.
MOVE 18 TO MQOD-OBJECTSTRING-VSLENGTH.
```

Publishing Tips

- **Choose topic string carefully**
  - Use sensible topic hierarchy
    - Based on context of published data
  - Don’t use different topic for each publish
    - This is probably meta data, use message property
  - Topic strings can be up to 10K bytes
    - But don’t use long topics unless necessary
- **Consider using Topic object and Topic string**
  - Administer can set point in topic tree
    - Known as ‘topic tree isolation’

Susbcribe to a topic



- Updates structure
  - Subscription Descriptor
- Very similar to MQOPEN

Subscribing Application

- MQSUB verb
- Subscription Descriptor (MQSD) describes the topic
  - MQSD-OBJECTSTRING
  - MQSD-OBJECTNAME
- Consume publications from the returned Q-HANDLE
  - when MQSO-MANAGED used

```
CALL 'MQSUB'
     USING HCONN,
           SUB-DESCRIPTOR,
           Q-HANDLE,
           SUB-HANDLE,
           MQCC,
           REASON.

CALL 'MQGET'
     USING HCONN,
           Q-HANDLE,
           MESSAGE-DESCRIPTOR,
           GMOPTIONS,
           BUFFER-LENGTH,
           BUFFER,
           DATA-LENGTH,
           MQCC,
           REASON
```

```
01 SUB-DESCRIPTOR.
   COPY CMQSDV.
ADD MQSO-CREATE MQSO-MANAGED MQSO-FAIL-IF-QUIESCING
                                    GIVING MQSD-OPTIONS.
MOVE 'Price/Fruit/Apples'    TO TARGET-TOPIC.
SET  MQSD-OBJECTSTRING-VSPTR TO ADDRESS OF TARGET-TOPIC.
MOVE 18 TO MQSD-OBJECTSTRING-VSLENGTH.
```

Subscription Descriptor (MQSD)

Field	Description
StrucId	Structure identifier
Version	Structure version number
Options	Options that control the action of MQSUB
ObjectName	Object Name
AlternateUserId	Alternate User Id
AlternateSecurityId	Alternate Security Id
SubExpiry	Subscription expiry
ObjectString	Object string
SubName	Subscription name
SubUserData	Subscription user data
PubPriority	Publication priority
PubAccountingToken	Publication accounting token
PubAppIdentityData	Publiation application identity data
SelectionString	String providing selection criteria
SubLevel	Subscription Level
ResObjectString	Resolved object string

Subscribe Options

```
10  MQSO-NON-DURABLE                PIC S9(9) BINARY VALUE 0.
10  MQSO-READ-AHEAD-AS-Q-DEF        PIC S9(9) BINARY VALUE 0.
10  MQSO-ALTER                      PIC S9(9) BINARY VALUE 1.
10  MQSO-CREATE                     PIC S9(9) BINARY VALUE 2.
10  MQSO-RESUME                     PIC S9(9) BINARY VALUE 4.
10  MQSO-DURABLE                    PIC S9(9) BINARY VALUE 8.
10  MQSO-GROUP-SUB                  PIC S9(9) BINARY VALUE 16.
10  MQSO-MANAGED                    PIC S9(9) BINARY VALUE 32.
10  MQSO-SET-IDENTITY-CONTEXT       PIC S9(9) BINARY VALUE 64.
10  MQSO-FIXED-USERID               PIC S9(9) BINARY VALUE 256.
10  MQSO-ANY-USERID                 PIC S9(9) BINARY VALUE 512.
10  MQSO-PUBLICATIONS-ON-REQUEST    PIC S9(9) BINARY VALUE 2048.
10  MQSO-NEW-PUBLICATIONS-ONLY      PIC S9(9) BINARY VALUE 4096.
10  MQSO-FAIL-IF-QUIESCING          PIC S9(9) BINARY VALUE 8192.
10  MQSO-ALTERNATE-USER-AUTHORITY   PIC S9(9) BINARY VALUE 262144.
10  MQSO-WILDCARD-CHAR              PIC S9(9) BINARY VALUE 1048576.
10  MQSO-WILDCARD-TOPIC             PIC S9(9) BINARY VALUE 2097152.
10  MQSO-SET-CORREL-ID              PIC S9(9) BINARY VALUE 4194304.
10  MQSO-SCOPE-QMGR                 PIC S9(9) BINARY VALUE 67108864.
10  MQSO-NO-READ-AHEAD              PIC S9(9) BINARY VALUE 134217728.
10  MQSO-READ-AHEAD                 PIC S9(9) BINARY VALUE 268435456.
```

- Options can be added together as required

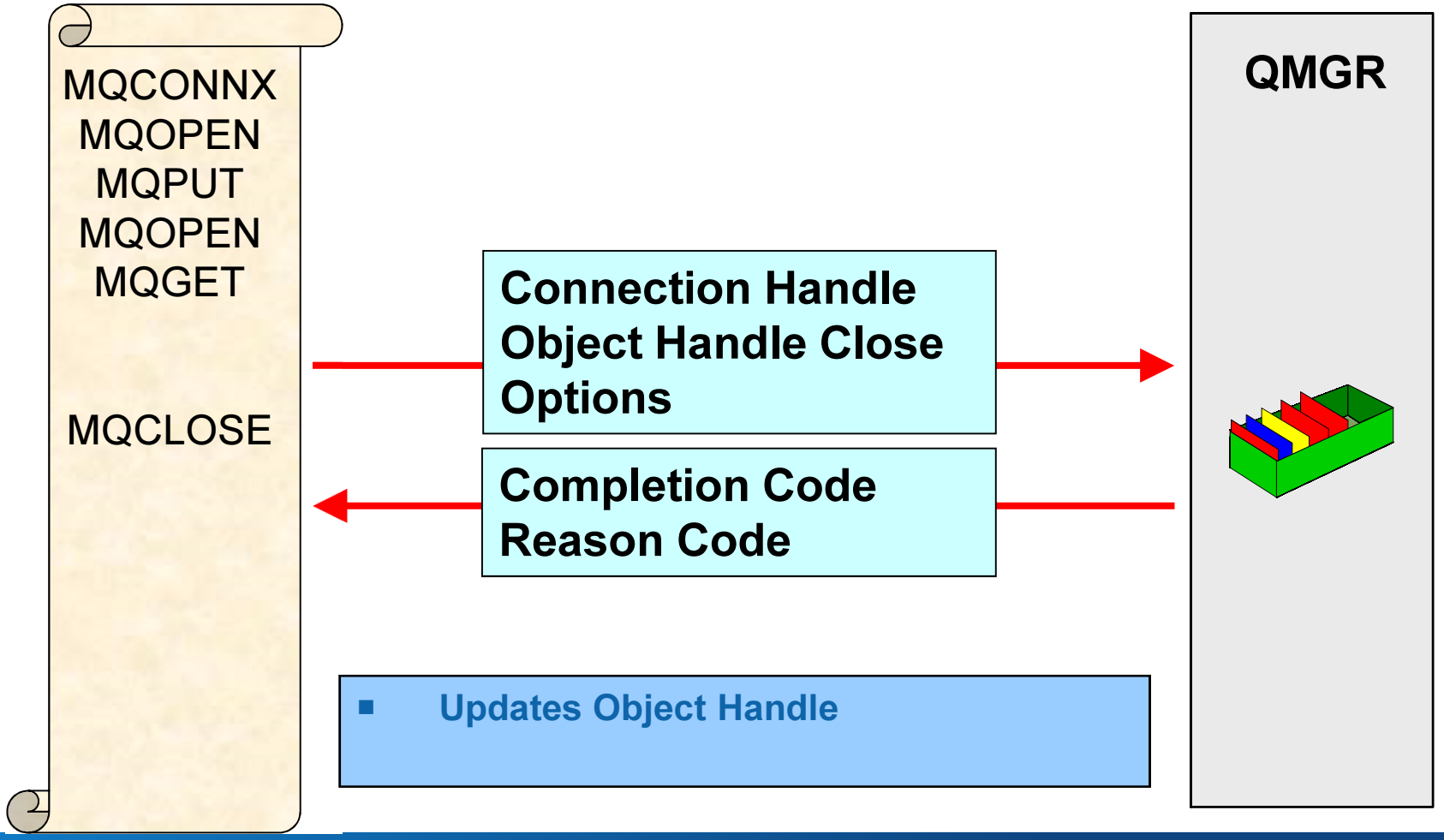
Subscribing Tips

- **Managed handles make things simpler**

- **Only use durable subscriptions when necessary**
  - Avoid build up of messages

- **For durable subscriptions**
  - Combine the create and resume options to make it simpler

Close a handle



- Updates Object Handle

Closing Application

- MQOPEN a queue
- MQCLOSE a queue
  - Normally we'd do something !

```
ADD MQOO-INPUT_SHARED
    MQOO-FAIL-IF-QUIESCING
                      GIVING OPTIONS.
CALL 'MQOPEN'
     USING HCONN,
           OBJECT-DESCRIPTOR,
           OPTIONS,
           Q-HANDLE,
           MQCC,
           REASON.

< Issue some MQI calls here >

CALL 'MQCLOSE'
     USING HCONN,
           Q-HANDLE,
           OPTIONS,
           MQCC,
           REASON.
```

```
01 OBJECT-DESCRIPTOR.
   COPY CMQODV.
MOVE MQOT-Q      TO MQOD-OBJECTTYPE.
MOVE PARM-QNAME  TO MQOD-OBJECTNAME.

MOVE MQCO-NONE   TO OPTIONS.
```

Close Options

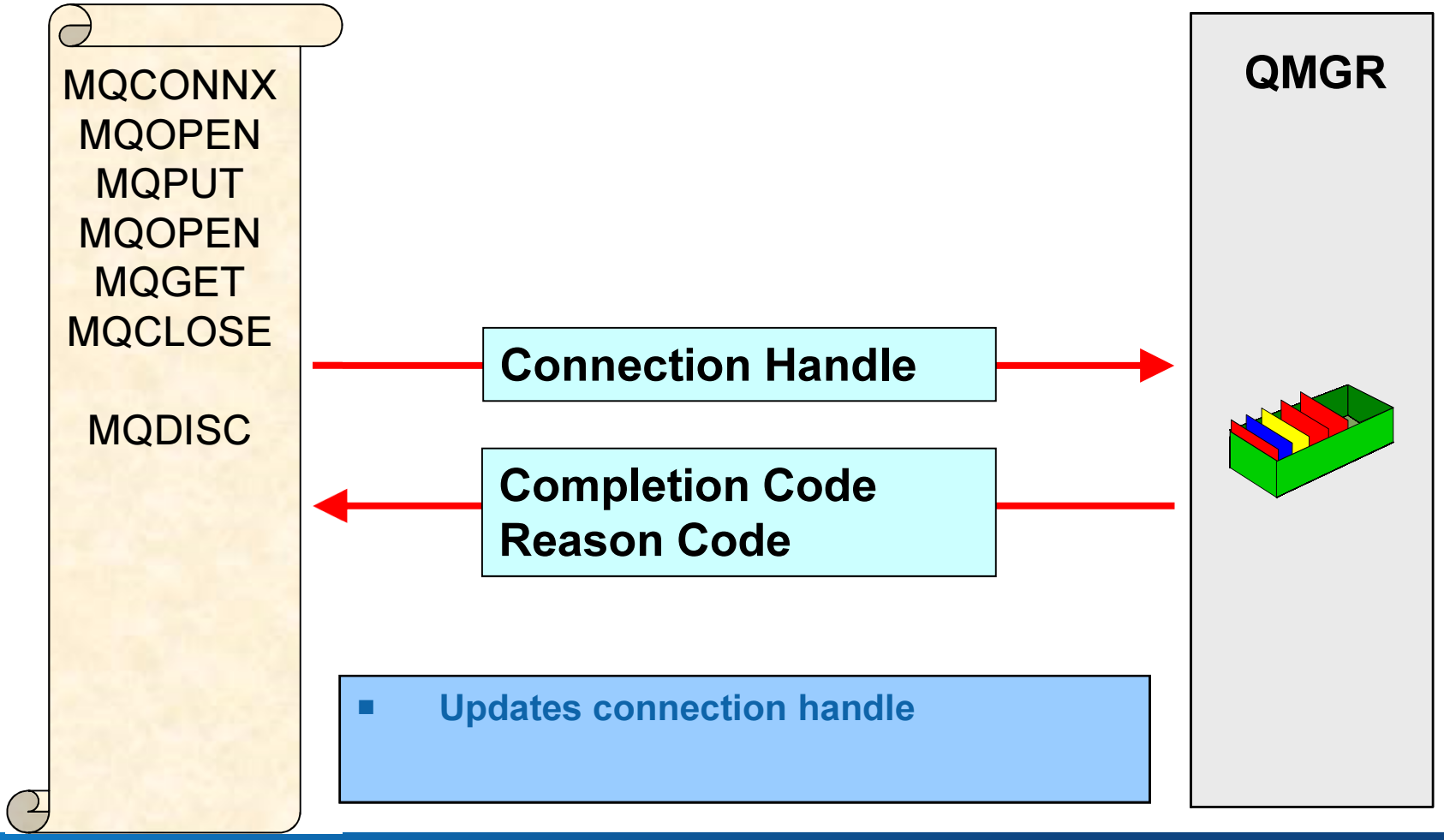
- **Options available depending on object type**

MQCO-DELETE	1	**Permanent Dynamic Queue**
MQCO-DELETE-PURGE	2	**Permanent Dynamic Queue**
MQCO-KEEP-SUB	4	**Durable Subscription**
MQCO-REMOVE-SUB	8	**Durable Subscription**
MQCO-QUIESCE	32	**Read Ahead input handle**

MQCLOSE Tips

- **In triggered applications**
  - Only close triggered queue if application ending
- **If implementing queue cache**
  - Close ‘rarely used’ queues in a timely fashion
    - Open queues can not be deleted/purged and use memory
- **For read ahead queues**
  - Use the quiesce close option to avoid message loss

Disconnect from Queue Manager



- Updates connection handle

Disconnecting Application

- MQCONN to Queue Manager
- MQDISC from Queue Manager
  - Normally we'd do something !

```
CALL 'MQCONNX'
  USING QM-NAME,
        CONNECT-OPTIONS,
        HCONN,
        MQCC,
        REASON.

< Issue some MQI calls here >


CALL 'MQDISC'
  USING HCONN,
        MQCC,
        REASON.
```

```
05 CONNECTION-OPTIONS.
   COPY CMQCNOV.

ADD MQCNO-HANDLE-SHARE-BLOCK MQCNO-RECONNECT
     GIVING CONNECT-OPTIONS.
```

MQDISC Tips

- **Ensure application disconnects if QM quiescing**
  - Will prevent Queue Manager from ending
- **MQDISC will close all queues/topics and subscriptions**
  - May wish to close some queues individually
- **MQDISC is an implicit commit**
  - May want to consider issuing MQBACK() first
- **Still call MQDISC**
  - If MQI call returns with a connection broken reason code
- **Application ending without MQDISC**
  - Will backout on Distributed
  - Will commit or backout depending on exit reason on z/OS
  - Try to always do explicit MQDISC if possible

Summary

- **Simple MQI – very easy to get started**
  - Let most fields have default values
  - Keep things simple if you can, for example
    - do not try and monitor channels
    - do not try to inquire queue depths
- **Plenty of samples to help you along**
  - In a variety of languages
    - eg. <install dir>\Tools\cobol\Samples
    - <hlq>.SCSQCOBS
- **Check reason codes and log failures**
  - MQ trace can be useful
- **Also check out**
  - “An Introduction to and Comparison of the Different MQ APIs” by Matt Whitehead

Questions & Answers

Morag Hughson – morag@mqgem.com

MQGem Software