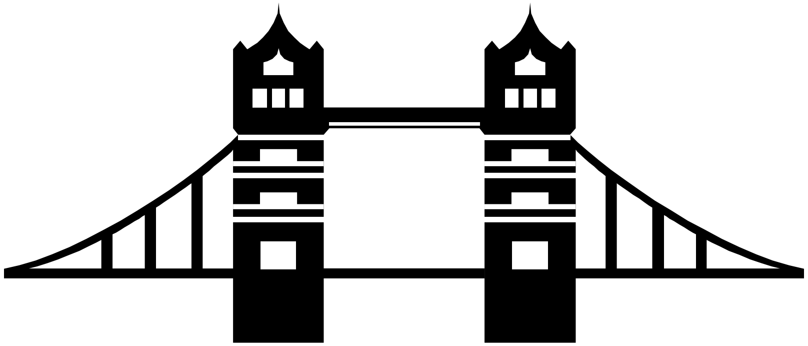
IBM MQ IMS Bridge and Adapter

Roger Lacroix
roger.lacroix@capitalware.com
http://www.capitalware.com

IBM MQ IMS Bridge Overview



Overview IMS Bridge

- The IBM MQ-IMS bridge is the component of IBM MQ for z/OS that allows direct access from IBM MQ applications to applications on your IMS system.
- The bridge enables **implicit** IBM MQ API support.

Overview IMS Bridge

- This means that you can re-engineer legacy applications that were controlled by 3270-connected terminals to be controlled by IBM MQ messages, without having to rewrite, recompile, or re-link them.
- The bridge is an IMS Open Transaction Manager Access (OTMA) client.

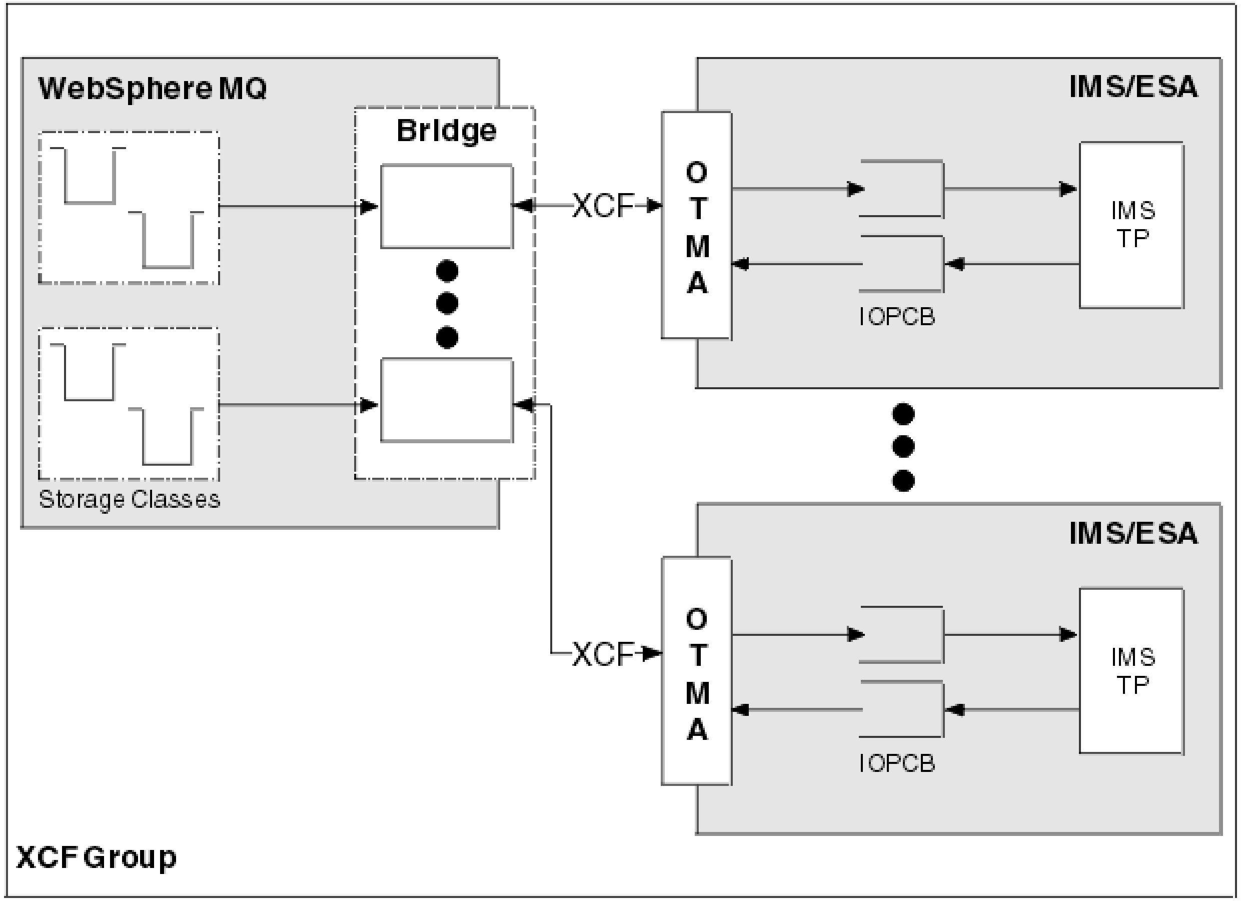
Overview IMS Bridge

- You can define IMS transactions as expecting single- or multi-segment input.
- The sending MQ application must build the IMS data following the MQIIH structure as one or more LLZZ-data segments.

  MQIIH LLZZ<trancode><data>[LLZZ<data>][LLZZ<data>]

- All segments of an IMS message must be contained in a single MQ message sent with a single MQPUT.
- The maximum length of an LLZZ-data segment is 32764 . The total MQ message length is the sum of the LL bytes, plus the length of the MQIIH structure.

Overview IMS Bridge



IMS Bridge Message Delivery Options

Expiry

- MQ 7.0.1 supports IMS Transaction Expiration
- MQ passes the remaining Expiry time to IMS as an IMS Transaction Expiration time

Programming with IMS Bridge

- In bridge applications there are no IBM MQ calls within the IMS application.
- The application gets its input using a GET UNIQUE to the IOPCB and sends its output using an INSERT to the IOPCB.
- IBM MQ applications use the IMS header (MQIIH) in the message data to ensure that the applications can execute as they did when driven by nonprogrammable terminals.

Programming with IMS Bridge

If a message cannot be put to the IMS queue, the following action is taken by IBM MQ:

- If a message cannot be put to the IMS queue because the message is invalid, the message is put to the dead-letter queue and a message is sent to the system console.

Programming with IMS Bridge

Continued:

- If the message is valid, but is rejected by IMS with a sense code of 001A and a DFS message, IBM MQ puts the original message to the dead-letter queue, and puts the DFS message to the reply-to queue. If IBM MQ is unable to put the DFS message to the reply-to queue, it is put to the dead-letter queue.

Programming with IMS Bridge

Continued:

- If the negative acknowledgment (NAK) from IMS represents a message error, an error message is sent to the system console, and the IBM MQ message is put to the dead-letter queue.

Programming with IMS Bridge

Something to remember:

- For proper data conversion, you must tell the queue manager what the format of the message is. If the message has an MQIIH structure, the Format in the MQMD must be set to the built-in format MQFMT_IMS, and the Format in the MQIIH must be set to the name of the format that describes your message data. If there is no MQIIH, set the Format in the MQMD to your format name.

Programming with IMS Bridge

Syncpoint calls:

- Use the IMS syncpoint facilities, such as the GU (get unique) to the IOPCB, CHKP (checkpoint), and ROLB (rollback) calls.
- You do not use IBM MQ’ MQBEGIN, MQCMIT or MQBACK under an IMS environment.

IBM MQ IMS Adapter Overview

IBM MQ IMS Adapter Overview

- The IMS control region is connected to one or more queue managers when it starts. This connection is controlled by IMS commands.
- The IMS adapter provided with IBM MQ gives access to IBM MQ from IMS.

IBM MQ IMS Adapter Overview

- The IMS adapter receives and interprets requests for access to IBM MQ using the External Subsystem Attach Facility (ESAF) provided by IMS.
- Usually, IMS connects to IBM MQ automatically without operator intervention.

IBM MQ IMS Adapter Overview

- With the IMS adapter, IBM MQ provides message queuing services for IMS-dependent regions.
- All application programs that run under IMS must have the stub CSQQSTUB and the IMS language interface module link-edited with them if they need to access IBM MQ.
- This stub provides the application with access to all MQI calls.

Programming with IMS Adapter

- Developers of message queuing IMS programs must use the MQCONN MQI call to specify the queue manager to which they want to connect.
- They can use the MQDISC call to disconnect from that queue manager.
- IMS programs can connect, consecutively or concurrently, to multiple queue managers on the same TCB.

Programming with IMS Adapter

- In the IMS environment, disconnection is forced when a program starts processing for a new user following a GU (get unique) IMS call.
- In the IMS environment, you can use the handle within the same task, but not within any subtasks.

Programming with IMS Adapter

- IMS applications should not specify an unlimited wait interval for a MQGET. This would prevent the IMS system terminating. When IMS terminates, it requires all dependent regions to end.
- Instead, IMS applications should specify a finite wait interval; then, if the call completes without retrieving a message after that interval, issue another MQGET call with the wait option.

Programming with IMS Adapter

Sample Compile JCL

```
//COBOL2  EXEC PGM=IGYCRCTL,
//             PARM='NODYNAM,LIB,MAP,OBJECT,RES,APOST,XREF'
//STEPLIB  DD  DSN=PP.COBOL2.V132.COB2COMP,DISP=SHR       <== CHANGE
//SYSLIB   DD  DSN=SYS1.MQM.V7R0M0.SCSQCOBC,DISP=SHR      <== CHANGE
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  DSN=&&LOADSET,DISP=(MOD,PASS),
//             UNIT=SYSDA,SPACE=(80,(500,200))
//SYSUT1   DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT2   DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT3   DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT4   DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT5   DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT6   DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT7   DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSIN    DD  DSN=TEST.DATASET(TEST01),DISP=SHR           <== CHANGE
```

Programming with IMS Adapter

Sample Link JCL

```
//LKED    EXEC PGM=IEWL,COND=(8,LE,COBOL2),
//             PARM='LIST,XREF,LET,RMODE=ANY,AMODE=31,REUS'
//SYSLIB   DD   DSN=PP.COBOL2.V132.COB2LIB,DISP=SHR
//RESLIB   DD   DSN=IMS.V51.R9602.DBDC.RESLIB,DISP=SHR      <== CHANGE
//SYSLMOD  DD   DSN=VIC.IB.V114.LOAD,DISP=SHR               <== CHANGE
//MQMLIB   DD   DSN=SYS1.MQM.V7R0M0.SCSQLOAD,DISP=SHR       <== CHANGE
//SYSUT1   DD   UNIT=SYSDA,DCB=BLKSIZE=1024,SPACE=(1024,(200,20))
//SYSPRINT DD   SYSOUT=*
//SYSLIN   DD   DSN=&&LOADSET,DISP=(OLD,DELETE)
//         DD   DDNAME=SYSIN
//SYSIN    DD   *
   INCLUDE RESLIB(DFSLI000)
   INCLUDE MQMLIB(CSQQSTUB)
   ENTRY DLITCBL
   NAME TEST01(R)
/*
```

Programming with IMS Adapter

Sample Execute JCL for MPP

```
//REGION   EXEC PGM=DFSRRC00,REGION=4M,
//     PARM=(MSG,080000000000,,A2,,A,00,3,5,,,,,,,'@,UPSI=00010000')
//*
//STEPLIB  DD   DISP=SHR,DSN=VIC.IB.V114.LOAD             <== CHANGE
//         DD   DISP=SHR,DSN=PP.COBOL2.V132.COB2LIB       <== CHANGE
//         DD   DISP=SHR,DSN=IMS.V51.R9602.DBDC.RESLIB    <== CHANGE
//         DD   DISP=SHR,DSN=SYS1.MQM.V7R0M0.SCSQAUTH     <== CHANGE
//         DD   DISP=SHR,DSN=SYS1.MQM.V7R0M0.SCSQLOAD     <== CHANGE
//SYSDBOUT DD   SYSOUT=*
//SYSUBOUT DD   SYSOUT=*
//SYSABOUT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
```

Programming with IMS Adapter

Sample Execute JCL for BMP

```
//BMP01    EXEC PGM=DFSRRC00,
//         PARM=('BMP,TEST01,PSBTST01,,,N00001,A1,,,,,15,10,,')
//*
//STEPLIB  DD   DISP=SHR,DSN=VIC.IB.V114.LOAD             <==  CHANGE
//         DD   DISP=SHR,DSN=PP.COBOL2.V132.COB2LIB       <==  CHANGE
//         DD   DISP=SHR,DSN=IMS.V51.R9602.DBDC.RESLIB    <==  CHANGE
//         DD   DISP=SHR,DSN=SYS1.MQM.V7R0M0.SCSQAUTH     <==  CHANGE
//         DD   DISP=SHR,DSN=SYS1.MQM.V7R0M0.SCSQLOAD     <==  CHANGE
//SYSDBOUT DD   SYSOUT=*
//SYSUBOUT DD   SYSOUT=*
//SYSABOUT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
```

Programming with IMS Adapter

Syncpoint calls:

- Transaction managers such as IMS, can participate in two-phase commit, coordinated with other recoverable resources. This means that the queuing functions provided by IBM MQ can be brought within the scope of a unit of work, managed by the transaction manager.

Programming with IMS Adapter

Syncpoint calls:

- Use the IMS syncpoint facilities, such as the GU (get unique) to the IOPCB, CHKP (checkpoint), and ROLB (rollback) calls.
- You do not use IBM MQ’ MQBEGIN, MQCMIT or MQBACK under an IMS environment.

IMS Trigger Monitor

IMS Trigger Monitor

- CSQQTRMN is an IBM MQ-supplied IMS application that starts an IMS transaction when an IBM MQ event occurs.
- CSQQTRMN must run as a non-message BMP.
- In an IMS environment, start an instance of CSQQTRMN to monitor an initiation queue and to retrieve the trigger messages from it as they arrive.

IMS Trigger Monitor

- CSQQTRMN schedules another IMS transaction by an ISRT to the IMS message queue.
- The IMS application reads the message from the message queue and then processes it.
- CSQQTRMN can only be used with the IBM MQ-IMS Adapter. It cannot be used with the IBM MQ-IMS Bridge.

Steps for IMS Trigger Monitor

1 The MQ IMS Trigger Monitor BMP (CSQQTRMN) is started

2 MQCONN to the MQ Queue Manager

3 MQOPEN the Initiation Queue

4 MQGET with Wait on the Initiation Queue

5 An MQ application MQPUTs a message to the triggered queue

6 MQ generates a trigger message and puts it on the initiation queue

Steps for IMS Trigger Monitor

7 MQ IMS Trigger Monitor BMP receives the trigger message

8 The MQ IMS Trigger Monitor BMP does CHNG/ISRT/PURG of the trigger message to the IMS Queue

9 The MQ IMS Trigger Monitor BMP issues a SYNC call

10IMS logs the trigger message

11IMS enqueues the trigger message to the IMS transaction

Steps for IMS Trigger Monitor

12 The IMS transaction is scheduled in an MPR

13 The IMS transaction does GU to the IOPCB and retrieves the trigger message

14 The IMS Transaction does MQCONN for the Queue Manager

15 The IMS Transaction does MQOPEN for the Input Queue

16 The IMS Transaction does MQGET for the real MQ message

Steps for IMS Trigger Monitor

17 The IMS Transaction processes the message including IMS and ESAF calls

18 The IMS Transaction does MQPUT1 for the MQ Reply message

19 The IMS Transaction does MQCLOSE for the MQ Input Queue

20 The IMS Transaction does MQDISC to the Queue Manager

21 The IMS Transaction does GU to the IOPCB to create an IMS syncpoint

IMS Trigger Monitor Gotchas

- The MQ IMS Trigger Monitor reads the MQ Trigger Message with NO_SYNCPOINT – the Trigger Message is deleted immediately
- If the BMP ABENDs before its SYNC call or IMS ABENDs before the message gets to the IMS message queue the Trigger Message is gone but the real message is still on the MQ queue
- If the triggering option was FIRST and this was the last message on the queue there will be no more Trigger Messages and the real message will not be retrieved until the TriggerInterval is reached
- If the triggering option is EVERY there will not be another trigger message until the next message arrives on the real queue
- The real message will not be processed until a new trigger message wakes up the MQ IMS Trigger Monitor

IMS Trigger Monitor Review

Advantages:

- It is provided by IBM
- Only the small trigger message is logged in IMS

Disadvantages:

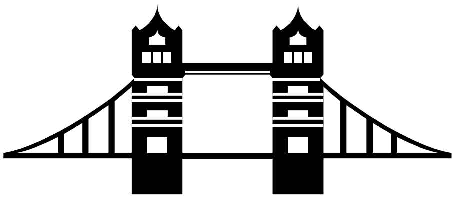
- A Trigger Monitor BMP can only wait on one Initiation Queue
- There are many steps for each message
- IBM MQ Triggering

Questions & Answers

IBM MQ IMS Bridge and Adapter

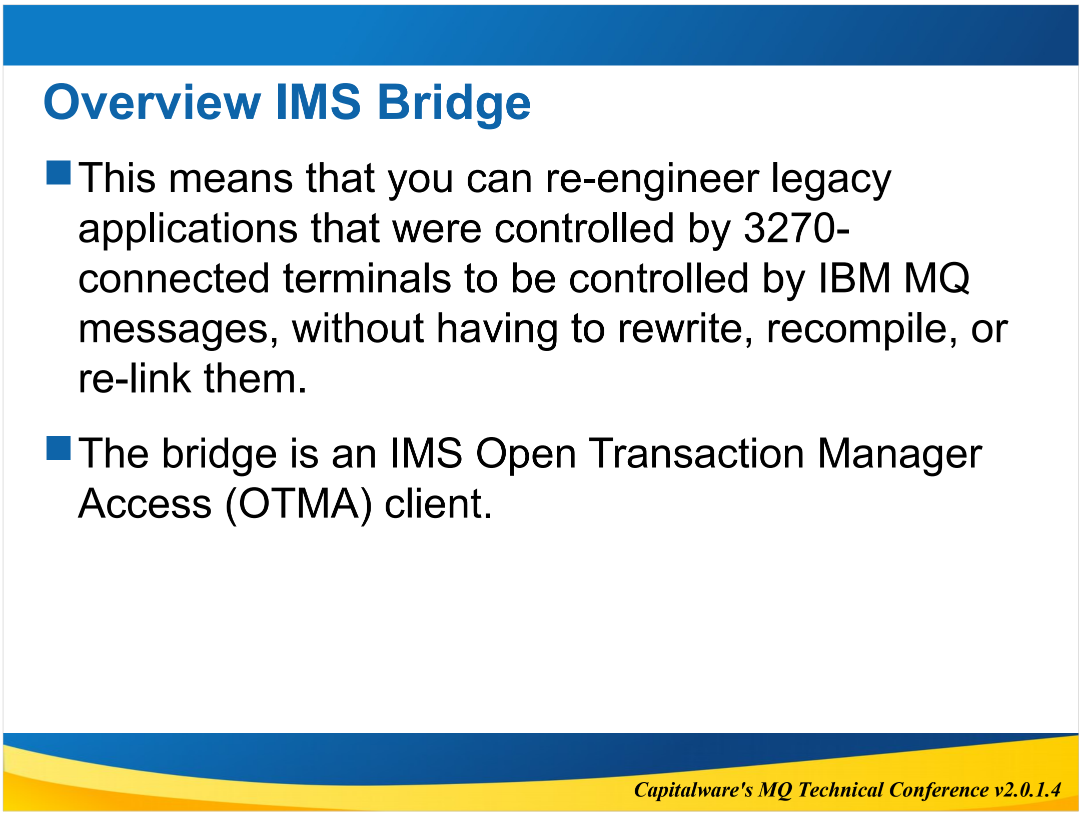
Roger Lacroix
roger.lacroix@capitalware.com
http://www.capitalware.com

IBM MQ IMS Bridge Overview



Overview IMS Bridge

- The IBM MQ-IMS bridge is the component of IBM MQ for z/OS that allows direct access from IBM MQ applications to applications on your IMS system.
- The bridge enables **implicit** IBM MQ API support.



Overview IMS Bridge

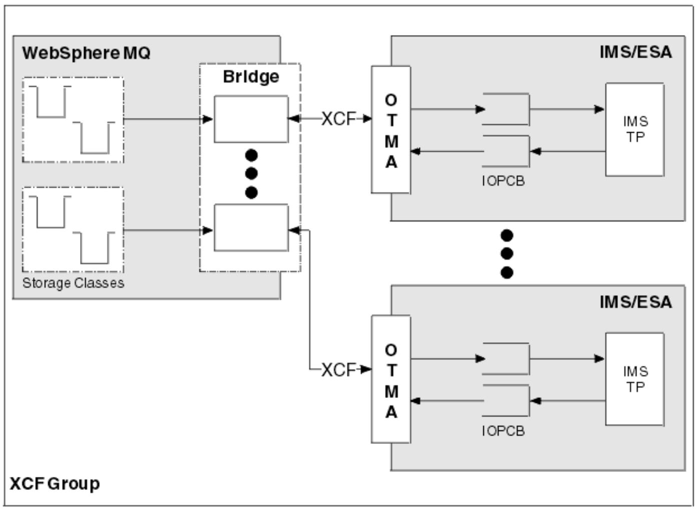
- This means that you can re-engineer legacy applications that were controlled by 3270-connected terminals to be controlled by IBM MQ messages, without having to rewrite, recompile, or re-link them.
- The bridge is an IMS Open Transaction Manager Access (OTMA) client.

Overview IMS Bridge

- You can define IMS transactions as expecting single- or multi-segment input.
- The sending MQ application must build the IMS data following the MQIIH structure as one or more LLZZ-data segments.

  MQIIH LLZZ<trancode><data>[LLZZ<data>][LLZZ<data>]
- All segments of an IMS message must be contained in a single MQ message sent with a single MQPUT.
- The maximum length of an LLZZ-data segment is 32764 . The total MQ message length is the sum of the LL bytes, plus the length of the MQIIH structure.

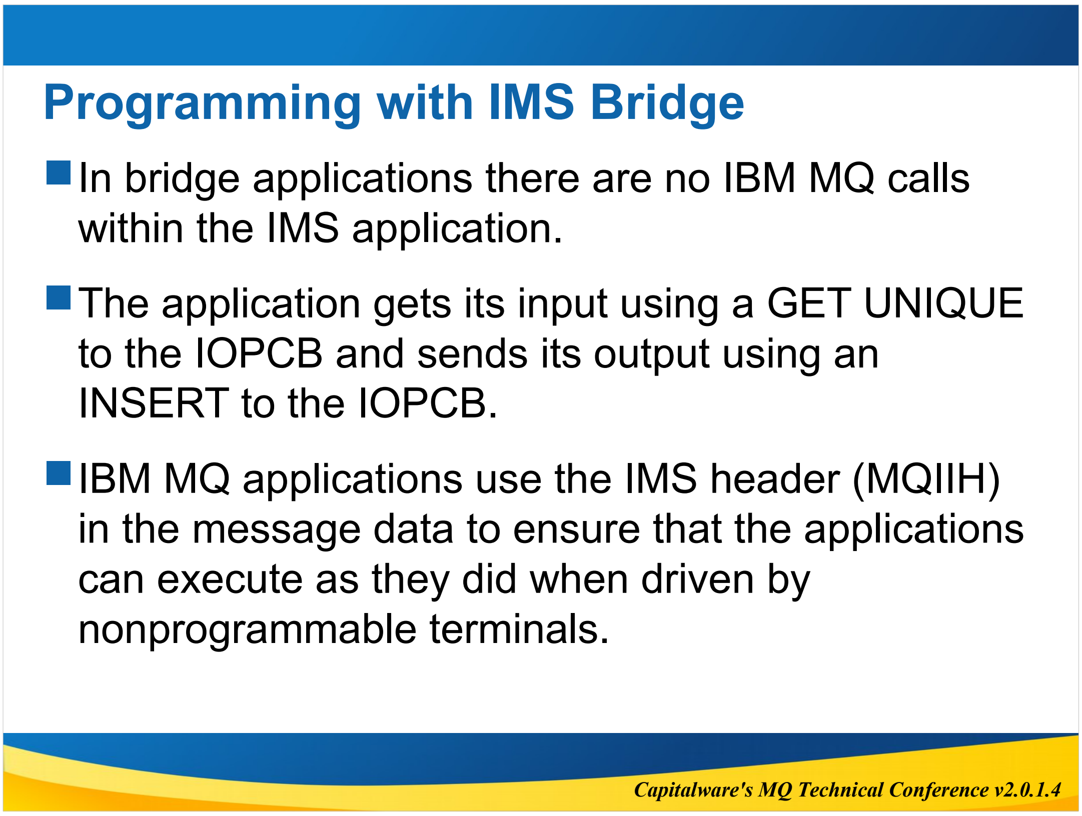
Overview IMS Bridge



IMS Bridge Message Delivery Options

Expiry

- MQ 7.0.1 supports IMS Transaction Expiration
- MQ passes the remaining Expiry time to IMS as an IMS Transaction Expiration time



Programming with IMS Bridge

- In bridge applications there are no IBM MQ calls within the IMS application.
- The application gets its input using a GET UNIQUE to the IOPCB and sends its output using an INSERT to the IOPCB.
- IBM MQ applications use the IMS header (MQIIH) in the message data to ensure that the applications can execute as they did when driven by nonprogrammable terminals.

Programming with IMS Bridge

If a message cannot be put to the IMS queue, the following action is taken by IBM MQ:

- If a message cannot be put to the IMS queue because the message is invalid, the message is put to the dead-letter queue and a message is sent to the system console.

Programming with IMS Bridge

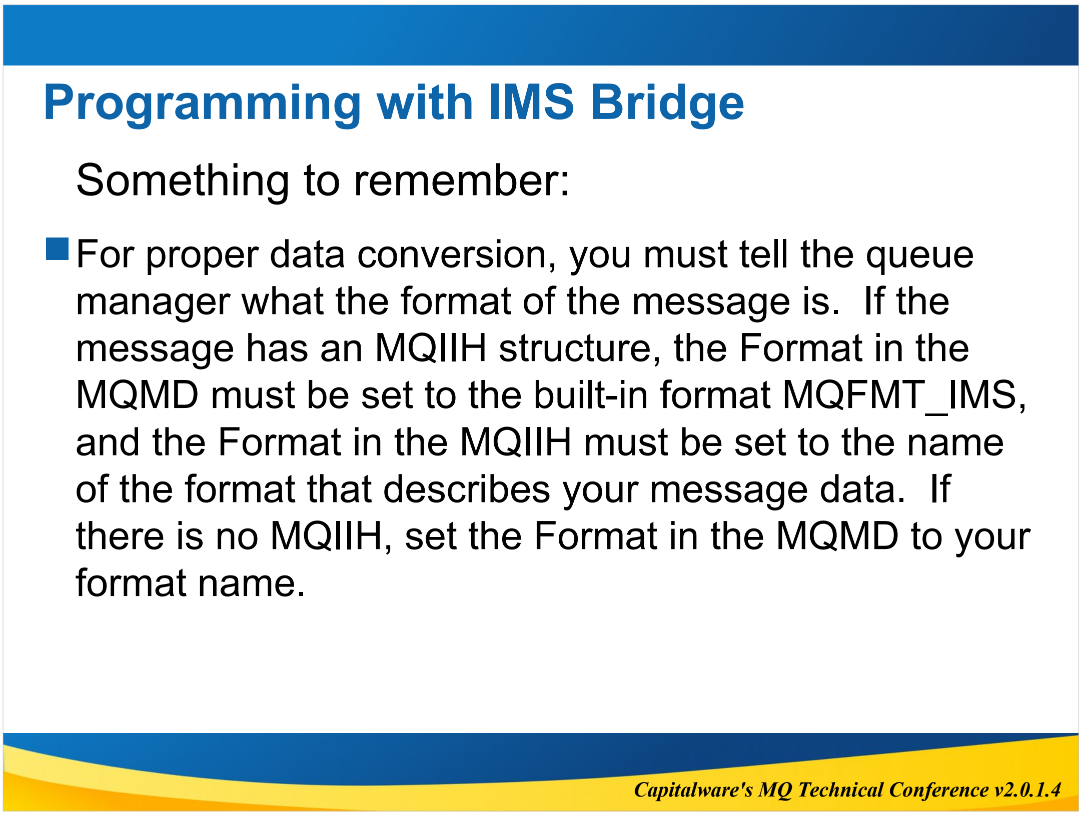
Continued:

- If the message is valid, but is rejected by IMS with a sense code of 001A and a DFS message, IBM MQ puts the original message to the dead-letter queue, and puts the DFS message to the reply-to queue. If IBM MQ is unable to put the DFS message to the reply-to queue, it is put to the dead-letter queue.

Programming with IMS Bridge

Continued:

- If the negative acknowledgment (NAK) from IMS represents a message error, an error message is sent to the system console, and the IBM MQ message is put to the dead-letter queue.



Programming with IMS Bridge

Something to remember:

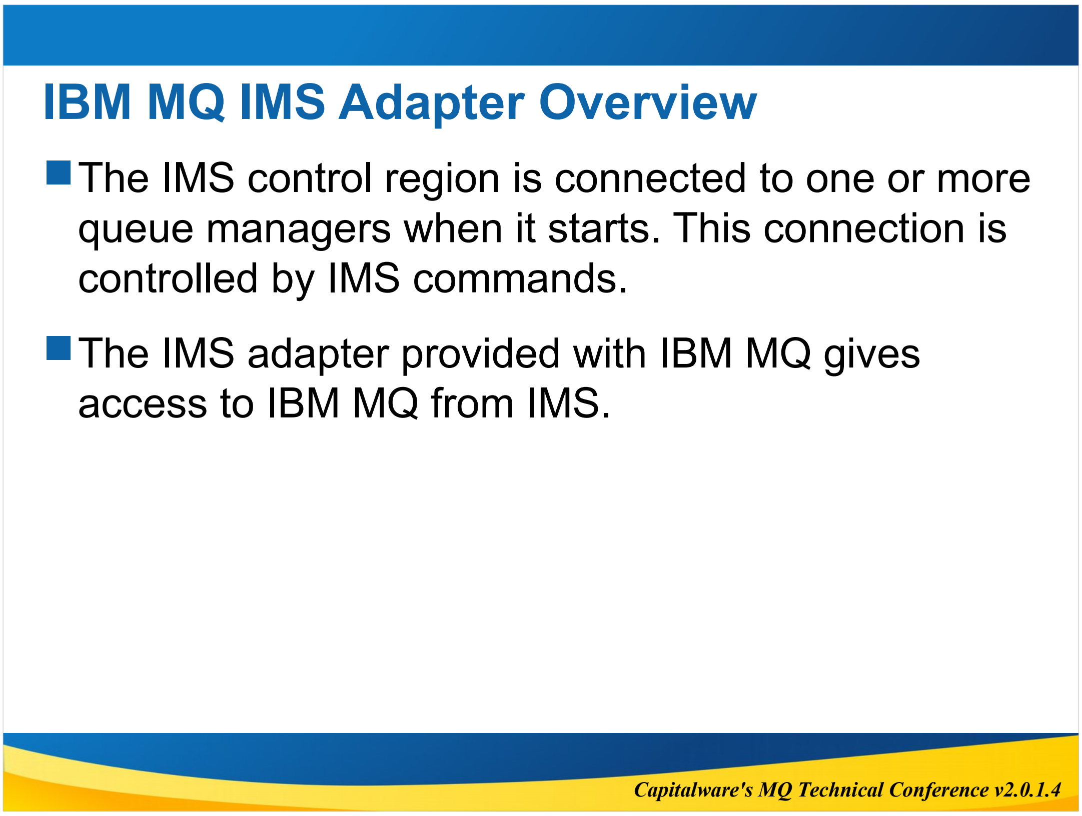
- For proper data conversion, you must tell the queue manager what the format of the message is. If the message has an MQIIH structure, the Format in the MQMD must be set to the built-in format MQFMT_IMS, and the Format in the MQIIH must be set to the name of the format that describes your message data. If there is no MQIIH, set the Format in the MQMD to your format name.

Programming with IMS Bridge

Syncpoint calls:

- Use the IMS syncpoint facilities, such as the GU (get unique) to the IOPCB, CHKP (checkpoint), and ROLB (rollback) calls.
- You do not use IBM MQ’ MQBEGIN, MQCMIT or MQBACK under an IMS environment.

IBM MQ IMS Adapter Overview



IBM MQ IMS Adapter Overview

- The IMS control region is connected to one or more queue managers when it starts. This connection is controlled by IMS commands.
- The IMS adapter provided with IBM MQ gives access to IBM MQ from IMS.

IBM MQ IMS Adapter Overview

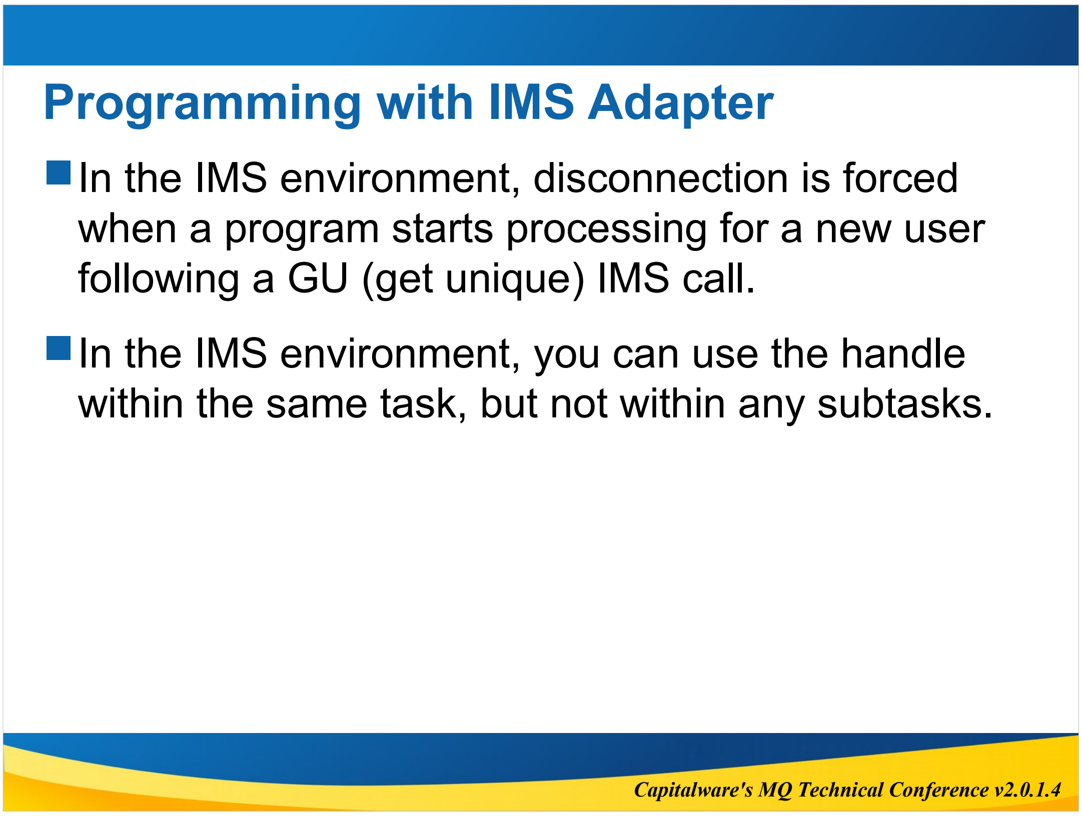
- The IMS adapter receives and interprets requests for access to IBM MQ using the External Subsystem Attach Facility (ESAF) provided by IMS.
- Usually, IMS connects to IBM MQ automatically without operator intervention.

IBM MQ IMS Adapter Overview

- With the IMS adapter, IBM MQ provides message queuing services for IMS-dependent regions.
- All application programs that run under IMS must have the stub CSQQSTUB and the IMS language interface module link-edited with them if they need to access IBM MQ.
- This stub provides the application with access to all MQI calls.

Programming with IMS Adapter

- Developers of message queuing IMS programs must use the MQCONN MQI call to specify the queue manager to which they want to connect.
- They can use the MQDISC call to disconnect from that queue manager.
- IMS programs can connect, consecutively or concurrently, to multiple queue managers on the same TCB.



Programming with IMS Adapter

- In the IMS environment, disconnection is forced when a program starts processing for a new user following a GU (get unique) IMS call.
- In the IMS environment, you can use the handle within the same task, but not within any subtasks.

Programming with IMS Adapter

- IMS applications should not specify an unlimited wait interval for a MQGET. This would prevent the IMS system terminating. When IMS terminates, it requires all dependent regions to end.
- Instead, IMS applications should specify a finite wait interval; then, if the call completes without retrieving a message after that interval, issue another MQGET call with the wait option.

Programming with IMS Adapter

Sample Compile JCL

```
//COBOL2   EXEC PGM=IGYCRCTL,
//              PARM='NODYNAM,LIB,MAP,OBJECT,RES,APOST,XREF'
//STEPLIB   DD  DSN=PP.COBOL2.V132.COB2COMP,DISP=SHR         <== CHANGE
//SYSLIB    DD  DSN=SYS1.MQM.V7R0M0.SCSQCOBC,DISP=SHR        <== CHANGE
//SYSPRINT DD   SYSOUT=*
//SYSLIN    DD  DSN=&&LOADSET,DISP=(MOD,PASS),
//              UNIT=SYSDA,SPACE=(80,(500,200))
//SYSUT1    DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT2    DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT3    DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT4    DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT5    DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT6    DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT7    DD  UNIT=SYSDA,SPACE=(460,(350,100))
//SYSIN     DD  DSN=TEST.DATASET(TEST01),DISP=SHR              <== CHANGE
```

Programming with IMS Adapter

Sample Link JCL

```
//LKED     EXEC PGM=IEWL,COND=(8,LE,COBOL2),
//              PARM='LIST,XREF,LET,RMODE=ANY,AMODE=31,REUS'
//SYSLIB    DD  DSN=PP.COBOL2.V132.COB2LIB,DISP=SHR
//RESLIB    DD  DSN=IMS.V51.R9602.DBDC.RESLIB,DISP=SHR      <== CHANGE
//SYSLMOD   DD  DSN=VIC.IB.V114.LOAD,DISP=SHR               <== CHANGE
//MQMLIB    DD  DSN=SYS1.MQM.V7R0M0.SCSQLOAD,DISP=SHR       <== CHANGE
//SYSUT1    DD  UNIT=SYSDA,DCB=BLKSIZE=1024,SPACE=(1024,(200,20))
//SYSPRINT DD   SYSOUT=*
//SYSLIN    DD  DSN=&&LOADSET,DISP=(OLD,DELETE)
//          DD  DDNAME=SYSIN
//SYSIN     DD  *
   INCLUDE RESLIB(DFSLI000)
   INCLUDE MQMLIB(CSQQSTUB)
   ENTRY DLITCBL
   NAME TEST01(R)
/*
```

Programming with IMS Adapter

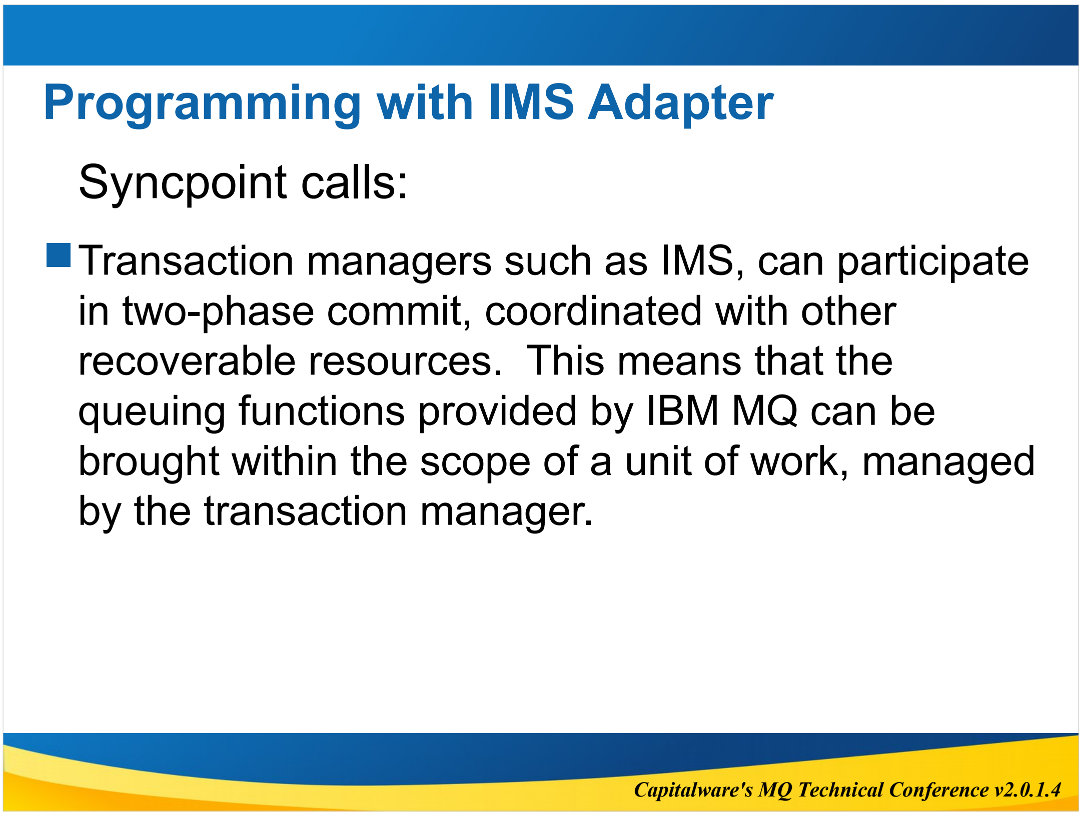
Sample Execute JCL for MPP

```
//REGION    EXEC PGM=DFSRRC00,REGION=4M,
//       PARM=(MSG,080000000000,,A2,,A,00,3,5,,,,,,,'@,UPSI=00010000')
//*
//STEPLIB   DD    DISP=SHR,DSN=VIC.IB.V114.LOAD             <== CHANGE
//          DD    DISP=SHR,DSN=PP.COBOL2.V132.COB2LIB       <== CHANGE
//          DD    DISP=SHR,DSN=IMS.V51.R9602.DBDC.RESLIB    <== CHANGE
//          DD    DISP=SHR,DSN=SYS1.MQM.V7R0M0.SCSQAUTH     <== CHANGE
//          DD    DISP=SHR,DSN=SYS1.MQM.V7R0M0.SCSQLOAD     <== CHANGE
//SYSDBOUT DD    SYSOUT=*
//SYSUBOUT DD    SYSOUT=*
//SYSABOUT DD    SYSOUT=*
//SYSPRINT DD    SYSOUT=*
//SYSOUT    DD    SYSOUT=*
```

Programming with IMS Adapter

Sample Execute JCL for BMP

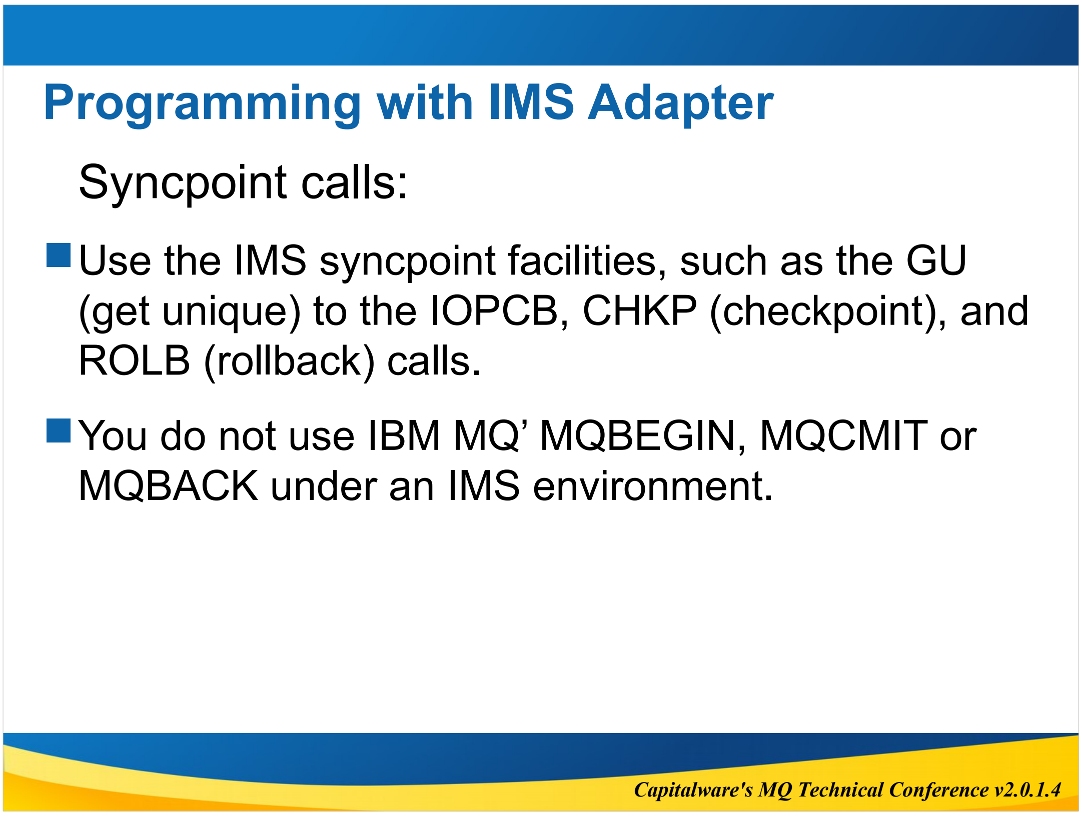
```
//BMP01     EXEC PGM=DFSRRC00,
//          PARM=('BMP,TEST01,PSBTST01,,,N00001,A1,,,,,15,10,,')
//*
//STEPLIB   DD   DISP=SHR,DSN=VIC.IB.V114.LOAD              <== CHANGE
//          DD   DISP=SHR,DSN=PP.COBOL2.V132.COB2LIB        <== CHANGE
//          DD   DISP=SHR,DSN=IMS.V51.R9602.DBDC.RESLIB     <== CHANGE
//          DD   DISP=SHR,DSN=SYS1.MQM.V7R0M0.SCSQAUTH      <== CHANGE
//          DD   DISP=SHR,DSN=SYS1.MQM.V7R0M0.SCSQLOAD      <== CHANGE
//SYSDBOUT DD   SYSOUT=*
//SYSUBOUT DD   SYSOUT=*
//SYSABOUT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
```



Programming with IMS Adapter

Syncpoint calls:

- Transaction managers such as IMS, can participate in two-phase commit, coordinated with other recoverable resources. This means that the queuing functions provided by IBM MQ can be brought within the scope of a unit of work, managed by the transaction manager.

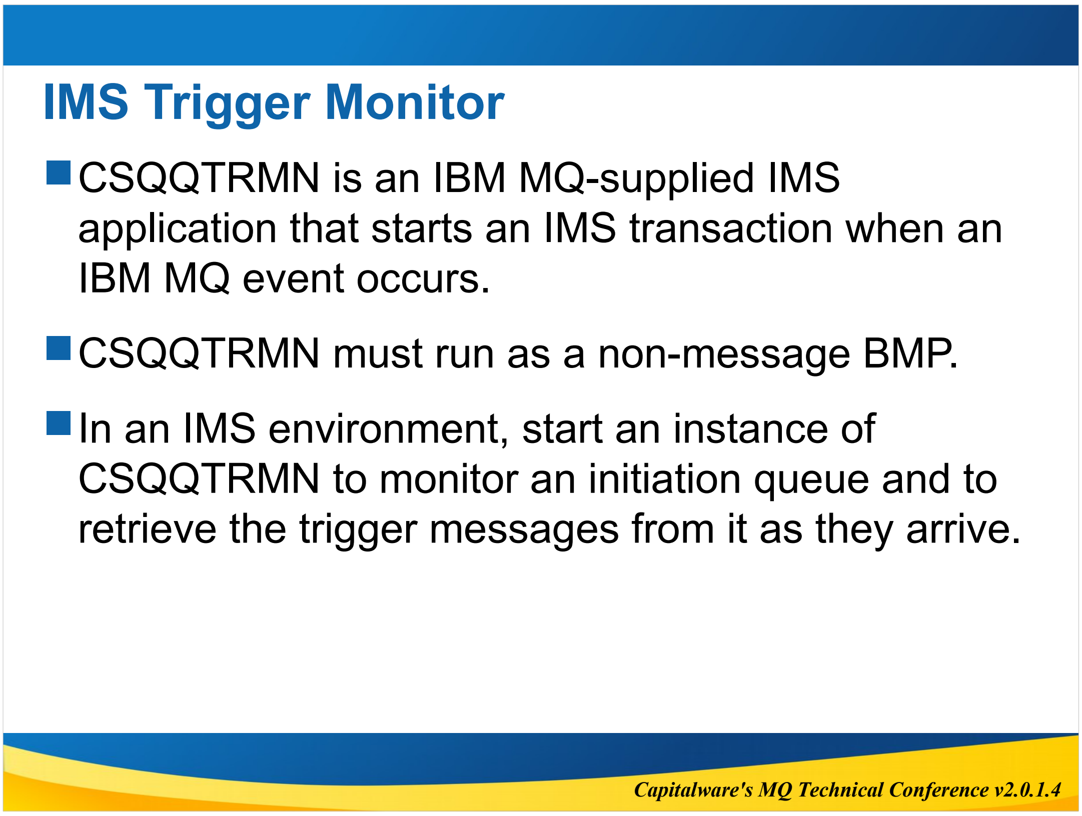


Programming with IMS Adapter

Syncpoint calls:

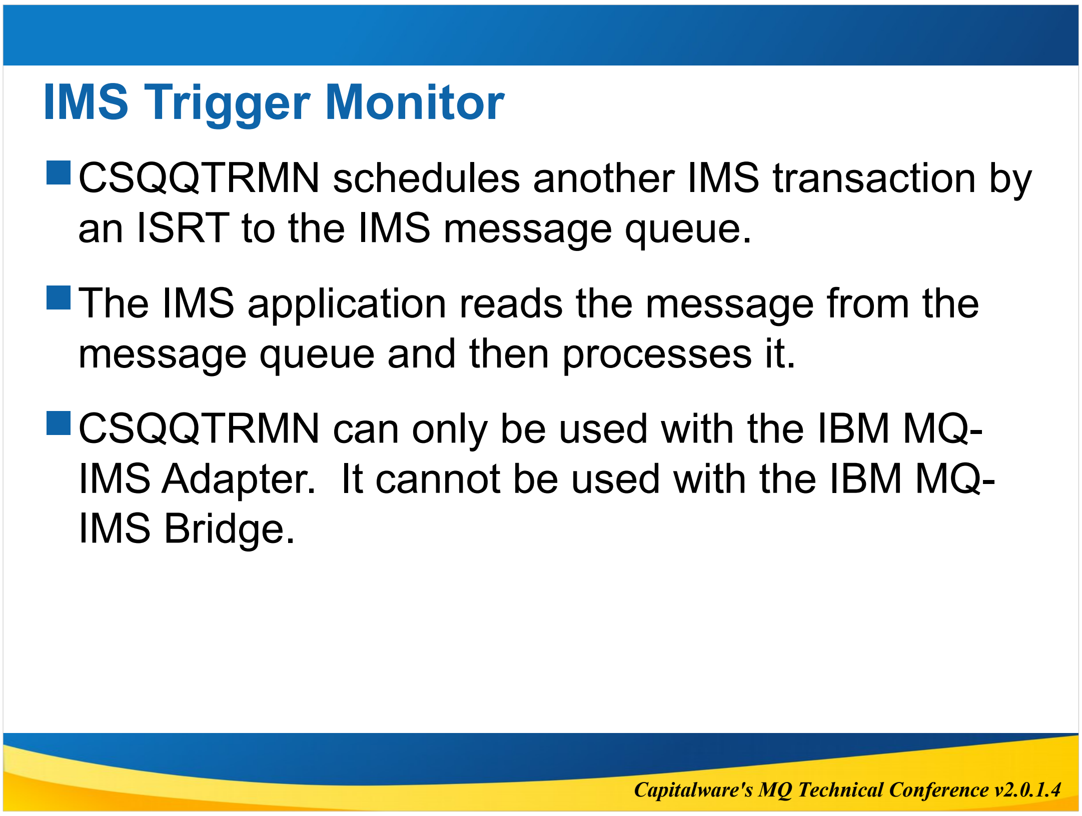
- Use the IMS syncpoint facilities, such as the GU (get unique) to the IOPCB, CHKP (checkpoint), and ROLB (rollback) calls.
- You do not use IBM MQ’ MQBEGIN, MQCMIT or MQBACK under an IMS environment.

IMS Trigger Monitor



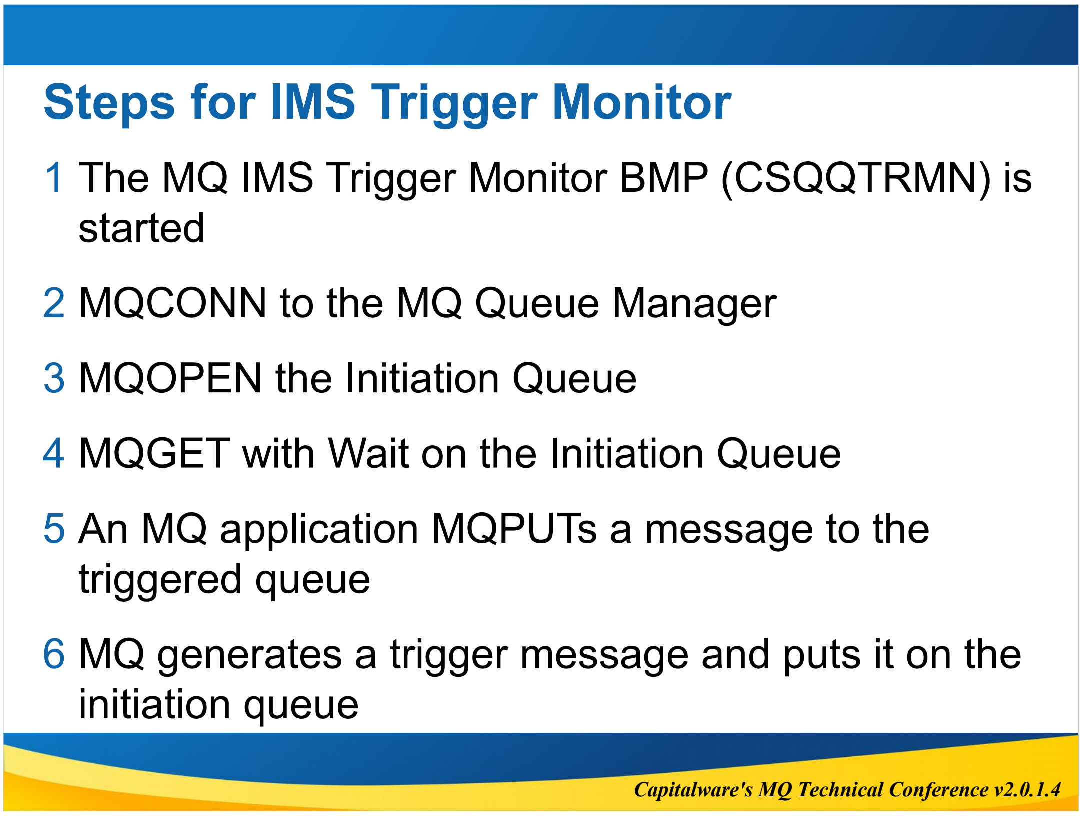
IMS Trigger Monitor

- CSQQTRMN is an IBM MQ-supplied IMS application that starts an IMS transaction when an IBM MQ event occurs.
- CSQQTRMN must run as a non-message BMP.
- In an IMS environment, start an instance of CSQQTRMN to monitor an initiation queue and to retrieve the trigger messages from it as they arrive.



IMS Trigger Monitor

- CSQQTRMN schedules another IMS transaction by an ISRT to the IMS message queue.
- The IMS application reads the message from the message queue and then processes it.
- CSQQTRMN can only be used with the IBM MQ-IMS Adapter. It cannot be used with the IBM MQ-IMS Bridge.



Steps for IMS Trigger Monitor

1 The MQ IMS Trigger Monitor BMP (CSQQTRMN) is started

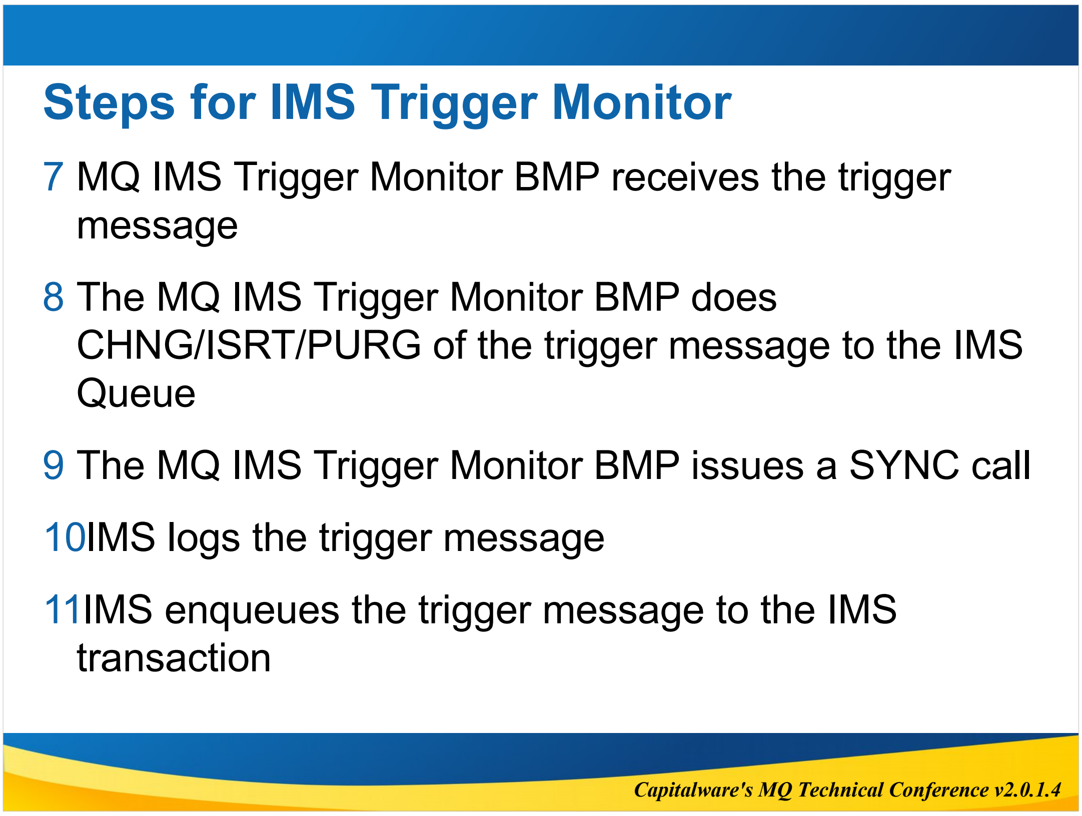
2 MQCONN to the MQ Queue Manager

3 MQOPEN the Initiation Queue

4 MQGET with Wait on the Initiation Queue

5 An MQ application MQPUTs a message to the triggered queue

6 MQ generates a trigger message and puts it on the initiation queue



Steps for IMS Trigger Monitor

7 MQ IMS Trigger Monitor BMP receives the trigger message

8 The MQ IMS Trigger Monitor BMP does CHNG/ISRT/PURG of the trigger message to the IMS Queue

9 The MQ IMS Trigger Monitor BMP issues a SYNC call

10IMS logs the trigger message

11IMS enqueues the trigger message to the IMS transaction

Steps for IMS Trigger Monitor

12 The IMS transaction is scheduled in an MPR

13 The IMS transaction does GU to the IOPCB and retrieves the trigger message

14 The IMS Transaction does MQCONN for the Queue Manager

15 The IMS Transaction does MQOPEN for the Input Queue

16 The IMS Transaction does MQGET for the real MQ message

Steps for IMS Trigger Monitor

17 The IMS Transaction processes the message including IMS and ESAF calls

18 The IMS Transaction does MQPUT1 for the MQ Reply message

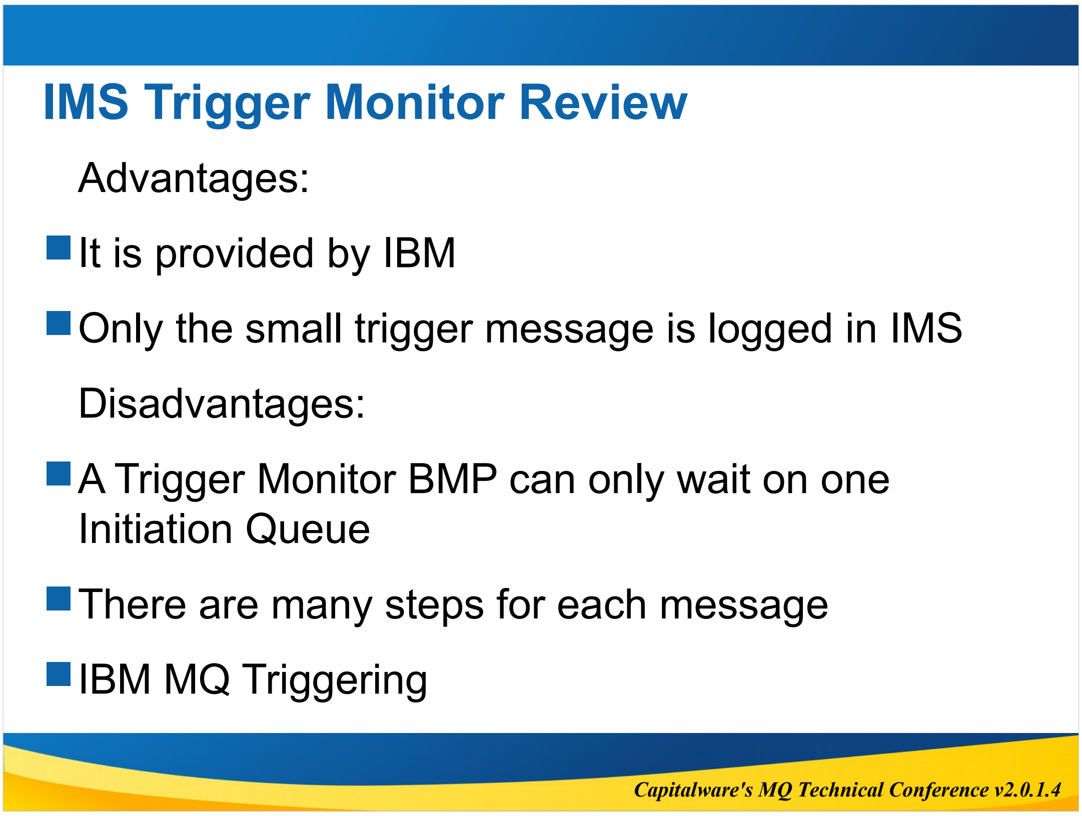
19 The IMS Transaction does MQCLOSE for the MQ Input Queue

20 The IMS Transaction does MQDISC to the Queue Manager

21 The IMS Transaction does GU to the IOPCB to create an IMS syncpoint

IMS Trigger Monitor Gotchas

- The MQ IMS Trigger Monitor reads the MQ Trigger Message with NO_SYNCPOINT – the Trigger Message is deleted immediately
- If the BMP ABENDs before its SYNC call or IMS ABENDs before the message gets to the IMS message queue the Trigger Message is gone but the real message is still on the MQ queue
- If the triggering option was FIRST and this was the last message on the queue there will be no more Trigger Messages and the real message will not be retrieved until the TriggerInterval is reached
- If the triggering option is EVERY there will not be another trigger message until the next message arrives on the real queue
- The real message will not be processed until a new trigger message wakes up the MQ IMS Trigger Monitor



IMS Trigger Monitor Review

Advantages:

- It is provided by IBM
- Only the small trigger message is logged in IMS

Disadvantages:

- A Trigger Monitor BMP can only wait on one Initiation Queue
- There are many steps for each message
- IBM MQ Triggering

Questions & Answers