

# CICS-MQ Update

John Tilling  
CICS Technical Planning and Strategy  
IBM UK Laboratories Hursley Park

Thursday 11<sup>th</sup> August 2011  
Session 9602

- © IBM Corporation 2011. All Rights Reserved.
- The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
- All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.
- The following are trademarks of the International Business Machines Corporation in the United States and/or other countries:
- [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml) AIX, CICS, CICSplex, DataPower, DB2, DB2 Universal Database, i5/OS, IBM, the IBM logo, IMS/ESA, Power Systems, Lotus, OMEGAMON, OS/390, Parallel Sysplex, pureXML, Rational, Redbooks, Sametime, SMART SOA, System z, Tivoli, WebSphere, and z/OS.
- A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml).
- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
- ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office
- Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

# Agenda

- CICS TS 3.2
  - Move of the CICS-MQ adapter, trigger monitor and bridge into CICS
  - Exploitation of OTE
- CICS TS 4.1
  - Group Attach support
- CICS TS 4.2
  - Group UR support
- CICS TS 3.2 and CICS TS 4.1
  - Support for new MQI verbs in WebSphere MQ V7

# CICS-MQ Integration in CICS TS 3.2

# CICS-MQ Interfaces

- CICS Transaction Server V3.2 supplies the following:
  - CICS-MQ Adapter
  - MQ trigger monitor for CICS
  - MQ bridge (includes the DPL bridge and link 3270 bridge)
- In CICS TS V3.2 CICS-MQ Adapter is enhanced to use OTE
  - The CICS-MQ TRUE is enabled as OPENAPI
  - MQ API commands from CICS applications are threadsafe
    - Reduced TCB switching for threadsafe applications – same as for DB2
  - CICS Level2 and Level 3 will service CICS shipped components
- Websphere MQ will continue to ship the above components for use with CICS TS 3.1 and below
  - Until such time that all releases of CICS TS prior to CICS TS 3.2 are out of service
  - Limited enhancements over time. No OTE support.
  - Will functionally stabilize
  - MQ Level2 and Level3 will continue to service MQ shipped components

# CICS-MQ CICS TS V3.2 Integration – name changes

- Transferred modules are renamed from CSQCxxxx to DFHMQxxx with CSQCxxxx aliases when required
- Exception: API crossing exit CSQCAPX.
  - Source supplied as CSQCAPX unchanged in SDFHSAMP
- API stub is DFHMQSTB but with alias CSQCSTUB and all other aliases currently supported, eg CSQCGET, CSQCPUT.....
  - Stub is shipped in SDFHLOAD and SDFHAUTH
  - No recompile and no re-linkedit of applications is required
- CICS-MQ TRUE is renamed from CSQCTRUE to DFHMQTRU
  - However TRUE has same entryname of MQM
  - DFHRMCAL TO=MQM will work unchanged
  - EXTRACT EXIT PROGRAM(CSQCTRUE) ENTRYNAME(MQM) commands work
    - *At runtime CICS substitutes DFHMQTRU as the program name*
  - INQUIRE EXITPROGRAM(CSQCTRUE) ENTRYNAME(MQM) commands work
    - *At runtime CICS substitutes DFHMQTRU as the program name*

# CICS-MQ CICS TS V3.2 Integration – name changes

- CICS supplies adapter, trigger monitor & bridge definitions in CSD group DFHMQ
  - DFHMQ group is part of list DFHLIST
  - Program names will be DFHMQxxx apart from crossing exit, stub aliases and aliases for those programs that can be EXEC CICS LINKed to by user applications
  - CKxx transaction names remain the same but refer to DFHMQxxx programs
- Users need to remove existing groups CSQCAT1 and CSQCKB containing CSQCxxxx definitions from the CSD (unless they are sharing the CSD)
  - Websphere MQ shipped PK39200 on V531 and PK42616 on V6 to disallow use of MQ shipped Attach with CICS TS 3.2 and above.
    - Produces message CSQC330E. A Technote was provided on the subject
- A CICS TS 3.2 CSD can be shared by lower level releases
  - Users should edit group CSQCAT1 & remove TDQUEUE CKQQ
    - CKQQ is now provided in group DFHDCTG
  - Groups CSQCAT1 and CSQCKB can then be installed on lower level releases
    - Overrides definitions in group DFHMQ

# CICS-MQ CICS TS V3.2 Integration

## – name changes



- Messages are renamed. CICS Message Domain used.
  - CSQCxxx become DFHMQxxx but the number will be retained
    - *Eg CSQC409I will become DFHMQ409I*
  - Terminal end user and TDQ messages are NLS enabled as for other CICS components
  - Kanj and Simplified Chinese versions shipped
  - CKQQ TD destination kept for connection/disconnect msgs
  - New CMQM TD destination (indirect to CSSL) provided in group DFHDCTG
- Qxxx abend codes renamed AMQx
  - Documented in CICS InfoCenter
- CICS-MQ documentation ported over from WMQ Infocenter
  - New CICS Integration with WebSphere MQ plugin to Infocenter
    - Information tailored to CICS TS 3.2 shipped components





# CICS-MQ CICS TS V3.2 Integration – Statistics and Monitoring

- MQCONN Global statistics supplied via DFH0STAT and DFHSTUP
  - CICS DSECT is DFHMQGDS
  - CPSM Base table is MQCONN
  - CPSM WUI views provided
- CICS Monitoring - Performance Class Data
  - Two new fields in DFHDATA group
    - WMQREQCT (count) – Total number of MQ requests
    - WMQGETWT (clock and count) MQ GETWAIT wait time
      - *Records CICS dispatcher wait identified as MqSeries GETWAIT*
    - NB. MqSeries TASKSWCH wait will no longer occur as TRUE uses L8 open TCBs
  - Existing group DFHRMI (activated via RMI=YES in MCT) is unaffected
    - Already contains RMIMQM – The total elapsed time spent in the CICS RMI for WebSphere MQ requests
      - *RMIMQM is a subset of total elapsed time spent in the RMI for all TRUEs (RMITIME in DFHTASK group)*

# CICS-MQ CICS TS V3.2 Integration – name changes

- TRACE
  - EXEC CICS ENTER TRACENUM previously used
    - Required master and user trace flags to be on for non exception traces
  - Components changed to use CICS Trace domain
    - Exception traces issued non-conditionally
    - Non exception traces controlled via Level 1 and Level 2 CETR component flags
      - *RI (RMI) component flags for Attach modules operating the other side of the RMI*
      - *New RA (Resource Manager Adapter) component flag for all other non RMI modules*
      - *BR component flags used for the MQ bridge*
    - Improved trace content – eg trace content of control blocks
- DUMP
  - CICS dump domain services used.
  - CICS IPCS verbexit enhanced
    - New MQ keyword to dump out CICS-MQ Attach control blocks
    - Formatting very similar to that provided for CICS-DB2 Attach

## CICS-MQ : Migration

- To migrate to CICS TS 3.2 or above and use MQ, ensure:
  - CSD groups CSQCAT1 and CSQCKB are **NOT** installed, and that CSD group DFHMQ is installed. Likewise for CPSM BAS ensure old definitions are not installed.
  - The following WebSphere MQ libraries are **AFTER** CICS libraries in the CICS STEPLIB and DFHRPL concatenations.
    - SCSQANLx \* (where x is the language letter of your national language)
    - SCSQCICS \*
    - SCSQLOAD
    - SCSQAUTH
  - \* The SCSQANLx and SCSQCICS libraries are only required if you wish to run the WebSphere MQ supplied samples
  - INITPARM system initialization table (SIT) override is changed from:

INITPARM=(CSQCPARM='SN=xxxx,TN=yyy,IQ=zzzzz')

to

INITPARM=(DFHMQPRM='SN=xxxx,IQ=zzzzz')

- In CICS TS 4.1 INITPARM is removed altogether. See later slide

# CICS-MQ Migration...



- For WMQ V6 apply apars PK42616 and PK59397
  - PK42616 polices use of the correct adapter.
  - PK59397 fixes data conversion problem. Requires CICS apar PK58227

# OTE exploitation across the adapters



- **CICS-DB2 Attach, CICS-MQ Attach, Comms Server Sockets (with OTE=YES) all use L8 TCBs**
  - **CICS TS 4.2 and IMS 12 add CICS-DBCTL to this list**
  - **An Application using all three will use one L8 TCB**
  
- **With the File Control Threadsafe enablement interleaving FC with DB2 or MQ will not cause TCB switching for threadsafe applications.**

# CICS-MQ Group Attach in CICS TS 4.1

# CICS TS 4.1: CICS-MQ Group Attach

- What it is
  - Ability to connect to any active member of a WMQ Queue Sharing Group (QSG) on the same LPAR rather than specifying an individual queue manager (QMGR)
    - NB. CICS still connects to a single QMGR at a time
- Why was it done
  - Customer demand!
  - Allows cloning of CICS-MQ definitions. Reduces reliance on single queue manager and thus reduces single point of failure
- Changed Externals
  - New RDO definitions
  - Removal of INITPARM use
  - Changes to WMQ adapter code
  - SPI and RDO support in CPSM WUI & CICS Explorer

# CICS TS 4.1: CICS-MQ Group Attach

- Design split into two parts:
  - Changes to resource definition and SPI
    - RDO – new MQCONN resource
    - Support via CEDA, DFHCSDUP, CPSM BAS, CREATE, DISCARD, CICS Explorer
  - Changes to WMQ Adapter
    - Replace INITPARM with use of MQCONN definition
    - Change connect code to query CSQGxxxx XCF group
    - CICS implements group attach selection mechanism and retry code.
      - *CICS uses 'randomising algorithm' to pick an active QMGR on the LPAR*
      - *No WMQ changes required. Works with WMQ 6.0 and 7.0*
      - *No peer recovery for indoubts so RESYNCMEMBER parm (same as for DB2)*



# Changes to RDO

- New MQCONN resource
  - Usual RDO name (1-8 chars), group and description
  - MQNAME 1-4 characters (specifies a QMGR or QSG name),
  - RESYNCMEMBER (YES | NO ) - applies only when indoubt UOWs are outstanding
    - YES – if indoubts outstanding override group attach and go back to QMGR last connected to
    - NO – if indoubts outstanding try last QMGR connected to. If not active then use group attach
  - INITQNAME 1-48 character initiation queue name for CKTI trigger monitor
- Full RDO, DFHCSDUP, CREATE, DISCARD, CPSM BAS and CICS Explorer support
- MQINI resource
  - Not definable in this release
  - Implicitly installed when INITQNAME is specified on the MQCONN
    - Uses reserved name of DFHMQINI
  - Only one MQINI in CICS TS 4.1
  - Allows for future expansion and future triggering enhancements in a later release

# Changes to SPI

- CEMT/EXEC CICS INQUIRE MQCONN
  - CONNECTST – returns status of connection
  - MQNAME – the name specified for a QMGR or QSG
  - MQQMGR – the name of the QMGR connected to
    - If not using group attach will be the same as MQNAME
    - If using group attach will show the QMGR selected
  - RESYNCMEMBER – indicates current setting
  - TASKS – returns the number of tasks (including trigger monitor tasks) using MQ
  - TRIGMONTASKS – returns the number of trigger monitor tasks using MQ
- CEMT/EXEC CICS INQUIRE MQINI
  - Returns name of Initiation queue for the one MQINI
- CEMT/EXEC CICS INQUIRE SYSTEM
  - MQCONN attribute returns name of installed MQCONN definition

# Changes to SPI

- EXEC CICS SET MQCONN
  - CONNECTST(CONNECTED) – to start the CICS-MQ connection
  - CONNECTST(NOTCONNECTED) – to stop the CICS-MQ connection
    - BUSY(WAIT) – quiesce and wait for completion
    - BUSY(NOWAIT) – initiate quiesce and return control
    - BUSY(FORCE) – force shutdown and wait for completion
      - *Will forcepurge tasks currently using MQ*
  - MQNAME – set name of QMGR or QSG – only when notconnected
  - RESYNCMEMBER – RESYNC or NORESYNC
- Same support for CEMT except shutdown uses NOWAIT or FORCE
- CKQC transaction still supported for startup, shutdown and display
  - Uses values from MQCONN defn and not INITPARM
- SIT parm MQCONN=YES unchanged.
  - Specifies that CICS should connect to MQ during startup (MQCONN defn must be in GRPLIST on cold or initial start. On restart MQCONN is recovered from the CICS global catalog)

## Changes to statistics

- New fields added to DFHMQGDS DSECT
  - MQCONN name
  - MQNAME
  - RESYNCMEMBER setting
  - Connect time (GMT)
  - Connect time (Local)
  - Disconnect time (GMT)
  - Disconnect time (Local)
- DFHSTUP and DFH0STAT updated to print new fields
- New fields added to CPSM views and to CICS Explorer views

# Changes to CKQC

- MQNAME value added to DISPLAY panel

```

cics
File Edit View Communication Actions Window Help
[Icons]
CKQCM2 Display Connection panel
Read connection information. Then press F12 to cancel.
CICS Applid = IYK2Z2G1 Connection Status = Connected QMgr name= MQD2
Mqname = DEV1 Tracing = On API Exit = Off
Initiation Queue Name = JTILLI1.INITQ
-----
Number of in-flight tasks = 1 Total API calls = 2
Number of running CKTI = 1
APIs and flows analysis Syncpoint Recovery
-----
Run OK 1 MQINQ 0 Tasks 0 Indoubt 0
Futile 0 MQSET 0 Backout 0 UnResol 0
MQOPEN 1 ----- Flows ----- Commit 0 Commit 0
MQCLOSE 0 Calls 6 S-Phase 0 Backout 0
MQGET 1 SyncComp 5 2-Phase 0
GETWAIT 1 SuspReqd 0
MQPUT 0 Msg Wait 1
MQPUT1 0 Switched 0
F1=Help F12=Cancel Enter=Refresh
MA d 01/001
Connected to remote server/host winmvsql.hursley.ibm.com using lu/pool IYAQTD34 and port 23
  
```

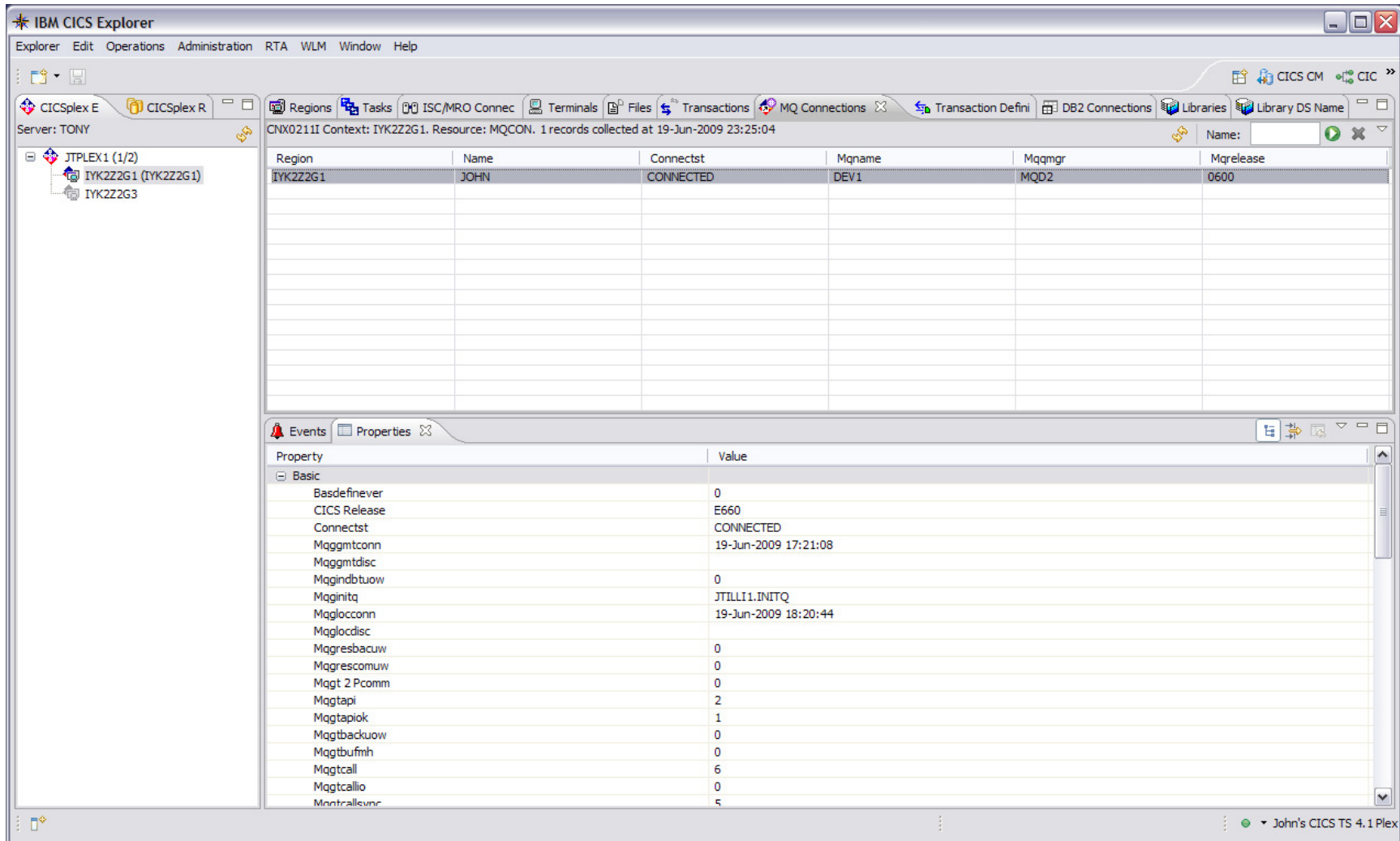
## Changes to CPSM

- Existing MQCONN basetable and WUI view uses information returned by CICS-MQ statistics (DFHMQGDS)
  - Still supported.
- New MQCON (one 'N') and MQINI base table and WUI views added. Returns same information as EXEC CICS INQUIRE MQCONN and MQINI
- MQCON view has support for startup and shutdown of CICS-MQ interface
- Views added to Operational view menus
- CPSM BAS supports defining and installing MQCONN definitions

# CICS Explorer support

- CICS-MQ supported via data interface (readonly) and Restful (read write)
- Support for viewing MQCON, MQINI and statistics information
- Support for SET MQCONN including CICS-MQ startup and shutdown (using RestFul interface)
- Support for installing MQCONN

# CICS Explorer support



IBM CICS Explorer

Explorer Edit Operations Administration RTA WLM Window Help

Regions Tasks ISC/MRO Connec Terminals Files Transactions MQ Connections Transaction Defini DB2 Connections Libraries Library DS Name

Server: TONY

CNX0211I Context: IYK2Z2G1. Resource: MQCON. 1 records collected at 19-Jun-2009 23:25:04

Region	Name	Connectst	Mqname	Mqqmgr	Mqrelease
IYK2Z2G1	JOHN	CONNECTED	DEV1	MQD2	0600

Events Properties

Property	Value
Basdefinever	0
CICS Release	E660
Connectst	CONNECTED
Mggmtconn	19-Jun-2009 17:21:08
Mggmtdisc	
Mgindbtuow	0
Mginitq	JTILLI1.INITQ
Mgloconn	19-Jun-2009 18:20:44
Mglocdisc	
Mgresbacuw	0
Mgrescomuw	0
Mgt 2 Pcomm	0
Mgtapi	2
Mgtapiok	1
Mgtbackuow	0
Mgtbufmh	0
Mgtcall	6
Mgtcallio	0
Montrallsvnc	5

John's CICS TS 4.1 Plex



# Changes to CICS-MQ Connect processing

- Do not use same name for a QSG and a QMGR
  - MQ documentation states this should not be done
  - QMGR will take precedence over a QSG
  - WMQ will disallow this in a future WMQ release
- CICS will attempt to connect to MQNAME passed as if it were a QMGR
- If QMGR exists and is active the connection will work (non group attach)
- If connect fails with MQRC\_Q\_MGR\_NAME\_ERROR (2058)
  - CICS issues IXCQUERY for group name CSQGxxxx via CICS SVC
  - Query status of qmgrs in group to ascertain which are active on same LPAR
  - If multiple are active check for RESYNCMEMBER(YES) and UOW status to determine whether we need or should connect to a particular member – or wait if not active
  - If not then pick a QMGR (uses randomising algorithm)
- Attempt to connect to chosen QMGR
  - If fails then depending upon return code choose next member, go through selection loop again
- If no queue managers active issue multiple connects to list of queue managers and wait on ECBLIST until first becomes available

# Migration to CICS TS 4.1

- CICS-MQ INITPARM no longer supported
  - Warning msg DFHMQ0218 issued if INITPARM detected stating values are ignored
  - Must install MQCONN before starting the CICS-MQ Attach else connect will fail with msg DFHMQ0317
  - INITPARM=(DFHMQPRM='SN=*queuemanager*,IQ=*initiation\_queue*') should be migrated to an MQCONN definition with attributes MQNAME(*queuemanager*) and INITQNAME(*initiation\_queue*)
- Group Attach uses new CICS SVC service
  - Must install new level of CICS TS 4.1 SVC and IPL else group attach will fail with msg DFHMQ0325

# CICS-MQ Group UR support in CICS TS 4.2

# Group Units of Recovery (GROUP UR)

- WMQ Group UR provides for peer recovery for in-doubt units of work in a queue-sharing group (QSG):
  - one WMQ queue manager (QMGR) can resolve in-doubts on behalf of another QMGR in the QSG.
- Existing WMQ for z/OS V7.0.1 Group UR support
  - XA clients only
  - No support for CICS TS
- Statement of Direction in the CICS TS 4.2 announcement letter:
  - IBM intends in the future for IBM WebSphere MQ for z/OS to provide Group Units of Recovery (Group UR) support for IBM CICS Transaction Server for z/OS (CICS TS).

# Group Units of Recovery (GROUP UR)

- New RESYNCMEMBER (GROUPRESYNC) setting on MQCONN defn:
  - Uses [WMQ-provided group attach function](#) and peer recovery
  - Requires a future release WMQ QMGR with Group UR enabled
- Existing RESYNCMEMBER settings (YES and NO) still supported:
  - Use existing [CICS-provided group attach function](#) and no peer recovery
- Changing RESYNCMEMBER settings take effect next time CICS connects to WMQ
- Do not change to/from GROUPRESYNC when indoubts outstanding - prevents automatic resolution of in-doubts:
  - CICS emits a message (DFHMQ2065 or DFHMQ2066) when this occurs
  - Revert to previous setting to get indoubts resolved

# Websphere MQ V7 exploitation in CICS TS 3.2 , 4.1 and 4.2

# New WebSphere MQ V7 MQI support

- CICS shipped new function APARs in December 2009 to support new MQI verbs with WMQ V7.01 and above
  - CICS TS 3.2 apar PK66866
  - CICS TS 4.1 apar PK89844 plus CPSM 4.1 apars PK89845 & PM03697
  - CICS TS 4.2 – in the base
- Recommended maintenance on WebSphere MQ V7.01
  - PK97364 & PK97972
- New V7 MQI support in CICS shipped adapters only
  - No support for older CICS TS releases in the MQ shipped adapter

# New MQI support

- Support for Publish/Subscribe (Pub/SUB), Message Properties and Asynchronous Consume
- For PUB/SUB, MQI supported is:
  - MQSUB, MQSUBRQ plus new options on MQOPEN, MQCLOSE, MQPUT and MQGET
- For Message Properties, MQI supported is:
  - MQBUFMH, MQCRTMH, MQDLTMH, MQDLTMP, MQINQMP, MQMHBUF, MQSETMP
- For Asynchronous Consume, MQI supported is:
  - MQCB, MQCTL, MQSTAT
- New entry points and aliases shipped with CICS-MQ stub
  - So must be linked with CICS provided stub

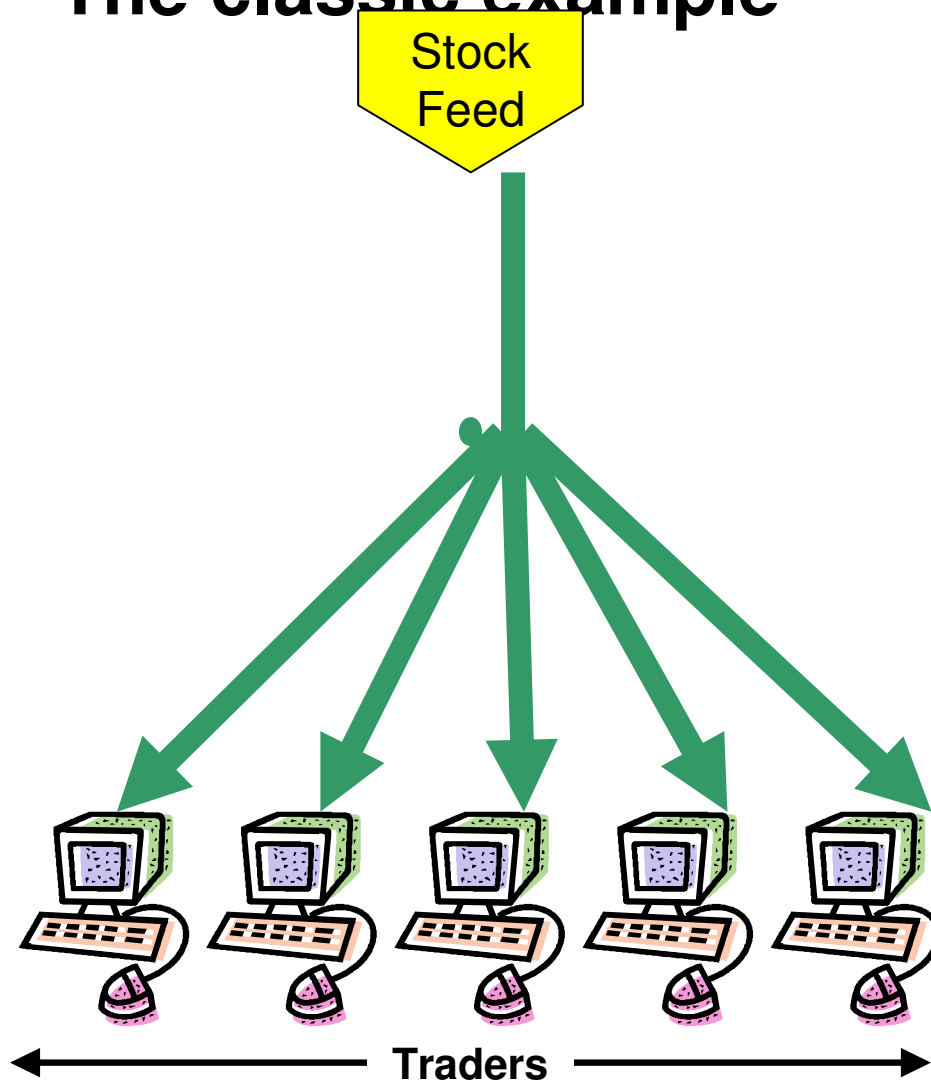


## What is Publish/Subscribe ?

**Publish/Subscribe is a term used to define an application model in which the provider of some information is decoupled from the consumers of that information.**

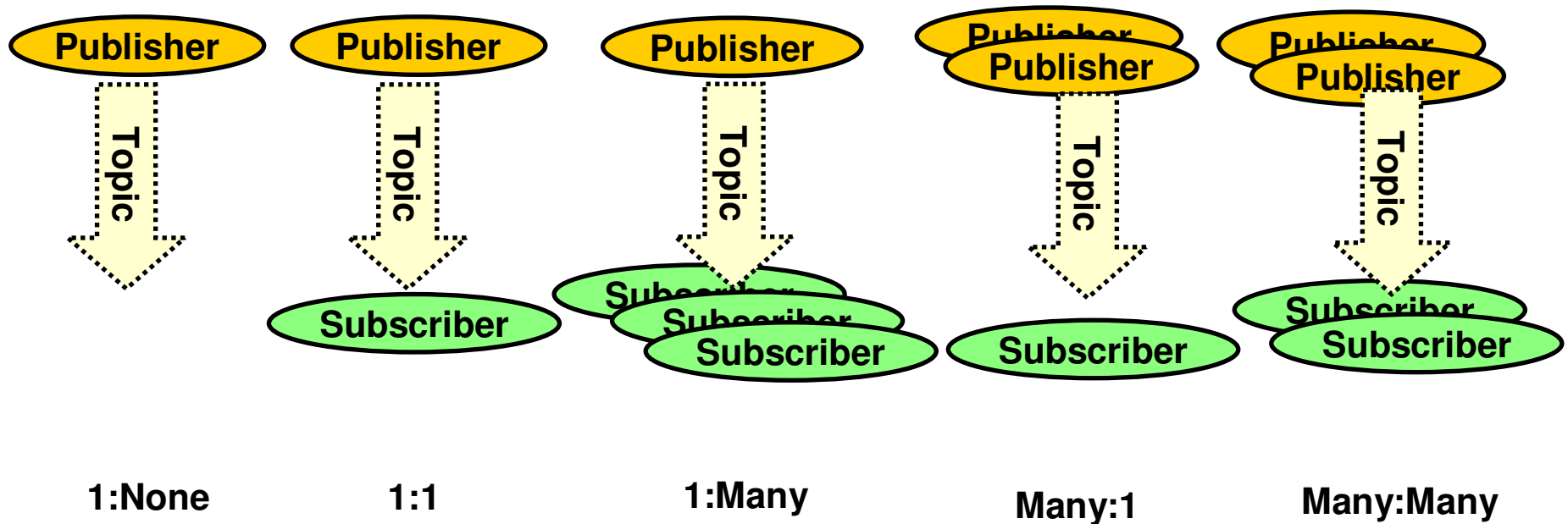
- providers of information need have no knowledge of consumers
- consumers of information need have no knowledge of providers
- new providers/consumers can be added without disruption
- Providers of information are called **publishers**
- Consumers of information are called **subscribers**

# The classic example



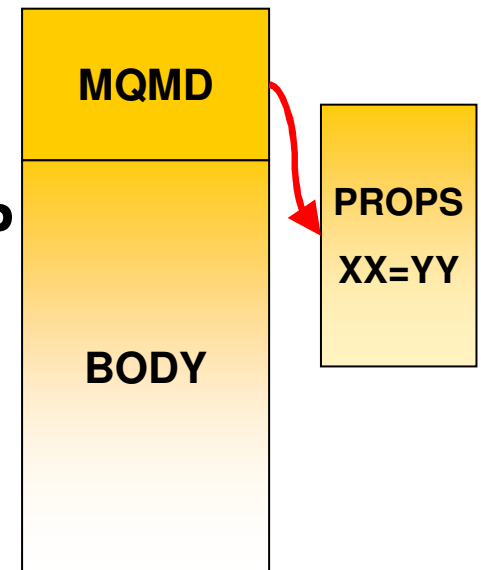
- A "feed" provides a continuous flow of information which is pushed to interested parties
- Traders consume this information and use it as a basis for the buying and selling stock

# Loose Coupling with Publish/Subscribe



# Message Properties

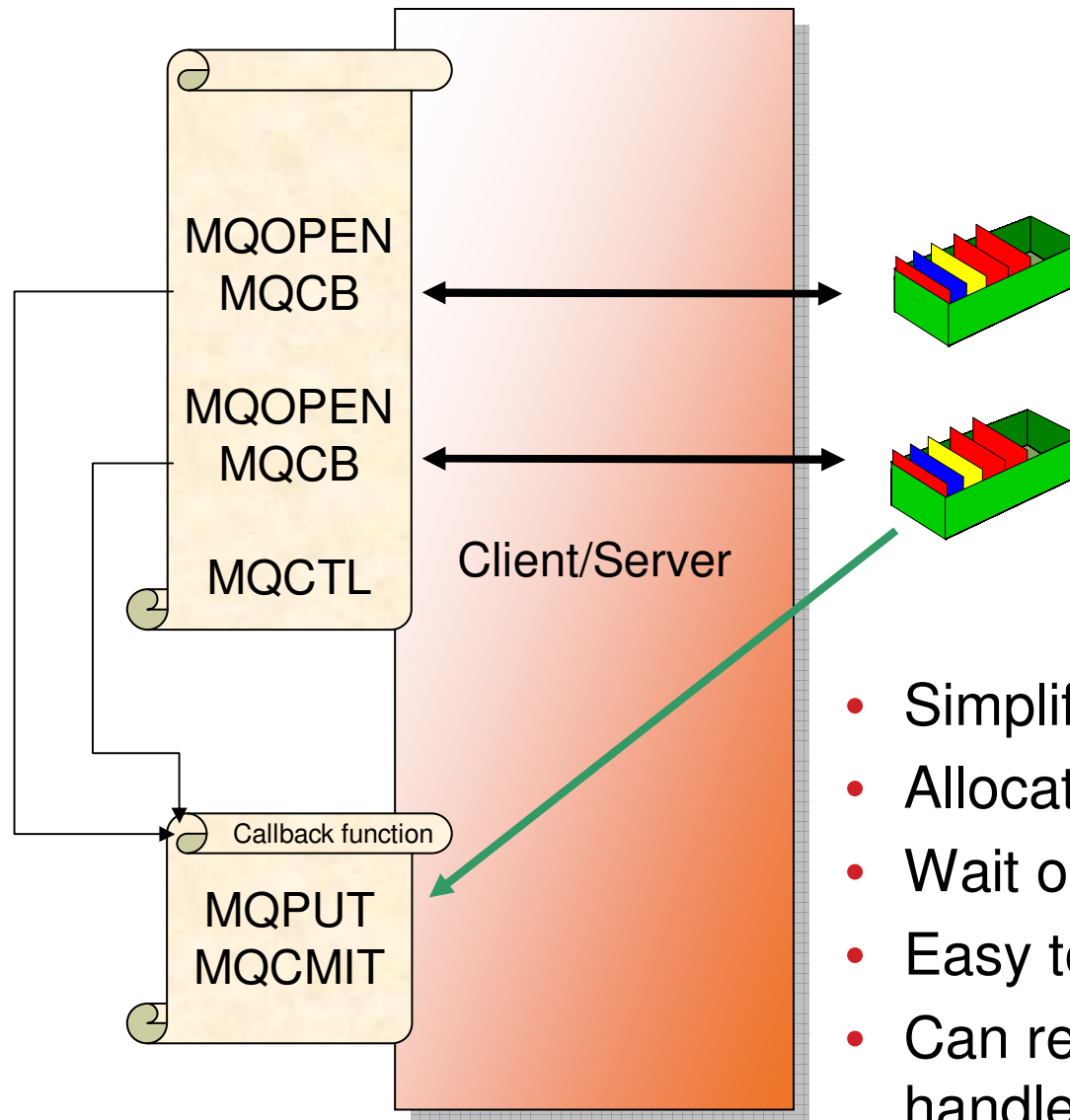
- Arbitrary values associated with the message but not part of the body
  - Like a user-extendable MQMD
  - Already part of JMS now part of the MQI in WMQ V7
- New verbs including **MQSETMP** and **MQINQMP**
  - Properties can be integers, strings, boolean, etc.
- Easier to use than RFH2 folders
  - Receiving apps do not see them unless they want
  - No need to parse and skip over message headers
- Configuration options for compatibility
  - Queue and channel attributes define behaviour
  - Defaults will create RFH2 folders for pre WMQ V7
- Permits explicit statement of relationships between messages
  - eg Message X is a **REPLY** to Message Y
  - Messages referred to by handles



# Message Properties

- Message properties are arbitrary values associated with the message but not part of the body of the message. You can think of them rather like a user-extendable message descriptor (MQMD). This concept is already part of the JMS API. Properties can be any of any basic type, for example integers, strings, or Booleans, as expressed in the appropriate programming language.
- Setting and Inquiring of message properties in a message is part of the MQI with the introduction of new verbs to allow this. Message handles are introduced to allow references to be made to a message within which message properties are to be manipulated.
- For applications whose source cannot be changed, the decision to discard message properties can be made based on the consumer of messages at consumption time rather than by the producer of the messages who doesn't necessarily know whether the consumer will be able to accept message properties.
- Message properties are exposed as MQRFH2 headers when they are sent to older queue managers, as they are today. Configuration attributes also allow removal of these properties as they are transferred.
- The introduction of a message handle also permits the application to state an explicit relationship between messages. For example, Message X is a REPLY to Message Y. This then potentially allows monitoring products to tie messages together, where previously they could only guess or have external configuration.

# Asynchronous Consumption of Messages



- Simplifies programming
- Allocates message buffers
- Wait on multiple queues
- Easy to cancel
- Can register an Event handler

# Asynchronous Consumption of Messages - Notes

- Asynchronous consumer allows the application to register an interest in messages of a certain type and identify a callback routine which should be invoked when a message arrives. This has the following advantages to the traditional MQGET application.
- **Simplifies programming**  
The application can continue to do whatever it was doing without needing to tie up a thread sitting in an MQGET call.
- **Allocates message buffers**  
The application does not need to 'guess' the size of the next message and provide a buffer just large enough. The system will pass the application a message already in a buffer.
- **Wait on multiple queues**  
The application can register an interest in any number of queues. This is very much simpler than using MQGET where one generally ended up polling round the queues.
- **Easy to cancel**  
The application can use either MQCTL or MQCB to stop consuming from a queue at any time. This is awkward to achieve when an application is using MQGET
- **Can register an Event handler**  
The application is notified of events such as Queue Manager quiescing or Communications failure.

# Asynchronous consume in CICS TS

- Use MQCB call to register/deregister programs known as callback routines
  - Callback routines can handle messages for multiple message destinations, including input queues and subscriptions.
  - When a suitable message is sent to the destination, it is passed to the callback routine.
  - You can also set up event handlers to be notified of conditions such as a queue manager quiescing.
- Use MQCTL to control starting, stopping, suspending, resuming callback routines
- Callback routine is a standard CICS command level application program
  - CICS obtains the messages passed by WebSphere MQ and links to the callback routine using EXEC CICS LINK passing the message data to the callback routine in a set of containers on a channel.



# CICS Callback routine requirements

- Can be in any programming language supported by CICS, but Java for a callback routine causes excessive TCB switching
- Compile and link-edit your callback routine with the RENT option and the options AMODE(31), RMODE(ANY).
- Code your callback routine to threadsafe standards
  - define it to CICS with the attributes CONCURRENCY(THREADSAFE) and API(CICSAPI)
  - or CONCURRENCY(REQUIRED) API (CICSAPI) with CICS TS 4.2
- Define your callback routine as a local program, not as remote or dynamic. Remote and dynamic programs are not supported

## Data passed to a callback routine

- Channel name is MQ\_ASYNC\_CONSUME

Container name	Content
<i>MsgDesc</i>	<i>MQMD2 - Message descriptor version 2</i>
<i>GetMsgOpts</i>	<i>MQGMO - Get Message Options</i>
<i>Buffer</i>	<i>The message. Options specified in MQGMO, such as truncation and conversion, have the same effects on the message as they do with an MQGET call.</i>
<i>Context</i>	<i>MQCBC - Callback Context version 2</i>

# Cobol sample showing Asynchronous consume in CICS TS

- WebSphere MQ V7.01 apar PM06722 shipped in March 2010 provides CICS async consume sample
- New program and transaction definitions in CSQ4SAMP
- Source code in SCSQSAMP, Load modules in SCSQLOAD
- Documentation in WMQ and CICS Infocenters:
  - [http://publib.boulder.ibm.com/infocenter/cicsts/v4r2/topic/com.ibm.cics.ts.wmq.adapter.doc/topics/async\\_sample.html](http://publib.boulder.ibm.com/infocenter/cicsts/v4r2/topic/com.ibm.cics.ts.wmq.adapter.doc/topics/async_sample.html)
- A Registration client program registers three Callback handlers (an Event Handler, and two Message Consumers) and starts Asynchronous Consume. A Messaging client program PUTs suitable messages for consumption by the two Message Consumers.

# Cobol sample showing Asynchronous consume in CICS TS

- Registration Client CSQ4CVRG should be started under a CICS transaction. It takes no input.
- When started, the Registration Client will register the following Callback handlers:
  - CSQ4CVEV as an Event Handler
  - CSQ4CVCN as a Message Consumer on a Topic
  - CSQ4CVCT as a Message Consumer on a Queue. The Registration Client passes to this handler a data structure containing the names of all three registered Callback handlers, together with the object handles associated with the two Message consumers.
- Having registered the Callback handlers the Registration Client will issue an MQCTL START\_WAIT to start Asynchronous Consume, and suspend until control is returned to it (e.g. by one of the Callback handlers issuing an MQCTL STOP).

# Cobol sample showing Asynchronous consume in CICS TS

- Event handler CSQ4CVEV
  - When driven, the Event Handler displays a message indicating the Calltype (START, REGISTER etc.).
  - When driven for MQ reason code CONNECTION\_QUIESCING, the Event Handler issues an MQCTL STOP to terminate Asynchronous Consume and return control to the Registration Client.
- Message Consumer CSQ4CVCN ("Simple" Message Consumer)
  - When driven, this Message Consumer displays a message indicating the Calltype (START, REGISTER etc.).
  - When driven for Calltype MSG\_REMOVED, the Message Consumer retrieves the inbound message and echoes it to the CICS joblog.

# Cobol sample showing Asynchronous consume in CICS TS

- Control Message Consumer CSQ4CVCT
  - When driven, this Message Consumer displays a message indicating the Calltype (START, REGISTER etc.).
  - When driven for Calltype MSG\_REMOVED, the Message Consumer retrieves the inbound message and the data structure passed by the Registration Client. Based on the message content it will issue appropriate MQCB or MQCTL commands to
    - STOP Asynchronous Consume (returning control to the Registration Client)
    - SUSPEND, RESUME or Deregister a named Callback handler (including itself)
- Messaging Client CSQ4CVPT has two functions:
  - To publish a message to a Topic for consumption by Message Consumer CSQ4CVCN
  - To put a control message to a Queue for consumption by the Control Message Consumer CSQ4CVCT, resulting in a potential change in behaviour of the overall sample

THE END!