

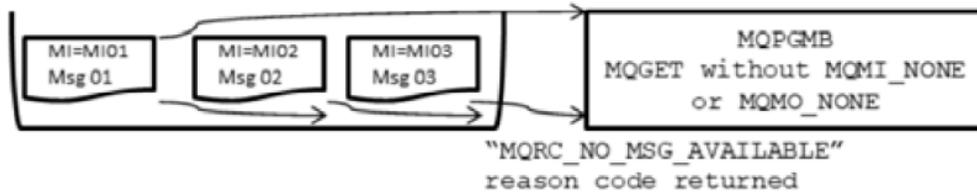


Welcome to the Virtual CICS user group newsletter. The Virtual CICS user group at www.fundi.com/virtualcics is an independently-operated vendor-neutral site run by and for the CICS user community.

Applications must always use either MQMI_NONE or MQPMO_NEW_MSG_ID for MQPUT



Applications must always use either MQMI_NONE or MQMO_NONE for MQGET



The latter situation is more of an issue since only one message is retrieved and all of the other messages are skipped

Figure 1: MQ usage in a COBOL program

Virtual CICS user group presentation

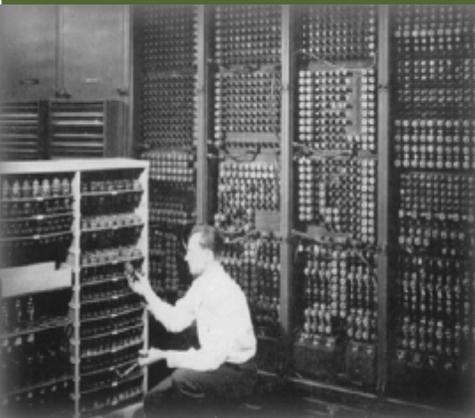
The latest webinar from the Virtual CICS user group was entitled, "The CICS/MQ Interface", and was presented by Tom Dunlap, CTO at Themis, Inc.

Tom has been working with CICS since its first

Throughout his career with CICS, he has developed applications, worked as a system programmer and administrator, been a lead architect for commercial software products that worked in CICS, and provided both applications and systems-level training to many corporations.

Contents:

Virtual CICS user group presentation	1
Meeting dates	4
Recent CICS articles	5
CICS news	5
About the Virtual CICS user group	5



The extent of his coding exposure in CICS includes, COBOL, Assembler, PL/I, C, and Java. Beyond his CICS background, he has work with z/OS (and its predecessors back to OS MFT) and many related components at both the applications and systems levels. This operating system experience includes efforts as a lead architect for commercial software products. Also included during his career is extensive work with DB2, performance and capacity planning efforts, large system integration efforts, and problem solving for large corporate application and systems environments. For the past 20 years, he has also been involved with work on distributed environments including UNIX, Linux, and Windows. These efforts include working with products like TXSeries (CICS on distributed systems), DB2, IBM MQ, plus many other supportive products. These efforts also include the use of application development tools like IBM RAD and IDz, CICS Explorer, DB2 Data Studio, plus the IBM Problem Determination Tools.

Tom Dunlap started his presentation by looking at the application program. He said that each application program that executes using MQ call statements requires a few basic inclusions of

generated code. The COPY statements required by all applications using MQ functions are CMQODV, CMQMDV, CMQGMOV, CMQPMOV, and CMQV SUPPRESS.

Looking at WS-VARIABLES, the MQ-CONN would normally be the handle acquired by an MQCONN function, but that call along with the MQDISC call are ignored by the CICS MQ interface. It is CICS itself that has performed the MQCONN to the queue manager. The MQ-HOBJ-I and MQ-HOBJ-O represent the handles, acquired on MQOPEN for a queue, used to specify which queue the other calls relate to. The MQ-COMPCODE and MQ-REASON are the variables that the application must test after every MQ call to determine its success.

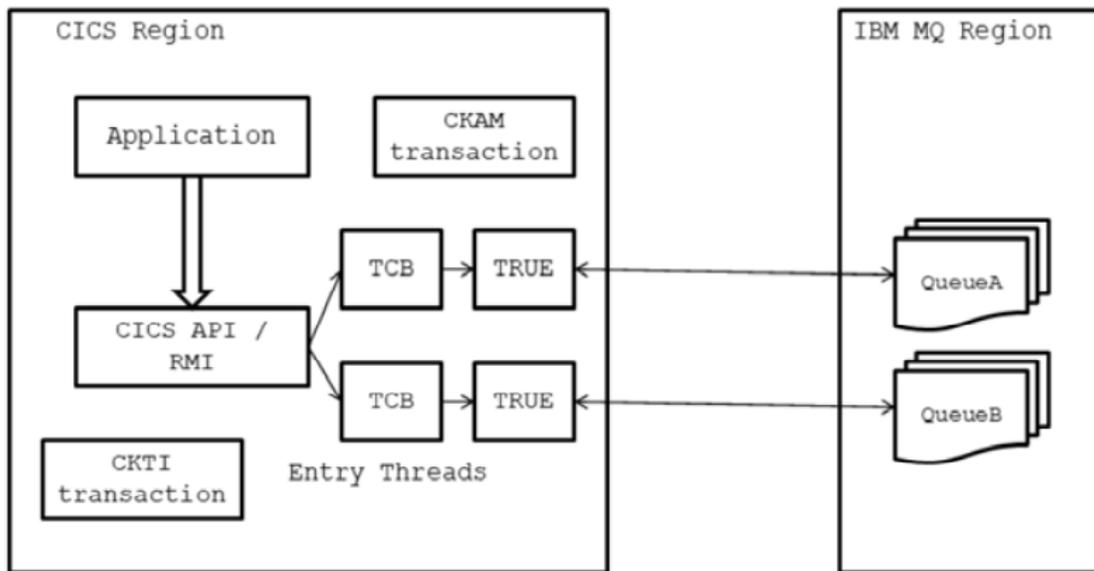
Tom recommended that larger message areas be put into the LINKAGE SECTION. The WORKING-STORAGE area is fine for messages of 200-300K. However, that size will vary depending on your CICS environment. When the application is using message areas in the LINKAGE SECTION, it will be responsible for performing a CICS GETMAIN command to acquire the storage for the message area.

The MQPUT1 call is a combination of an MQOPEN,

MQPUT, and MQCLOSE in a single call. Often, a CICS application will work with a single message within a task, so this call saves you a bit of code, plus interacting with the MQ interface twice. This does mean less overhead to the executing task. However, this savings disappears when you process more than one message with an MQPUT1 call. The actual MQPUT1 call itself is a "static call", which means it will resolve to an actual module at Linkage Edit time. You cannot use the COBOL DYNAM option with MQ applications since the name on the call is not the actual module name.

Tom's presentation included lots of other examples for application programs.

Figure 1 shows MQ usage in a COBOL program. The MQMD_MSGID will always be a unique, generated value as long as the applications do their part. The application is required to always set the attribute to MQMI_NONE or include the option MQPMO_NEW_MSG_ID for every MQPUT or MQPUT1 call. If they do not provide this code, the MsgId generated on the first call is reused for all subsequent calls. This is because the generated MsgId is passed back to the application by MQ in the MQMD-MSGID attribute. This is intended to provide the application with a unique



The IBM MQ interface provides “multi-threaded” access to the connected MQ region

MQ will accept requests and process them against the specified queue

Figure 2: The CICS MQ interface

“tracking marker” for each message.

When retrieving messages, the MQMD_MSGID attribute should be set to MQMI_NONE before the MQGET call, or the MQGMO_MATCHOPTIONS attribute set with MQMO_NONE.

If the application does not provide this code, then only the first message is retrieved and the second MQGET call issued will completely pass through the rest of the messages in the queue and return an error. This is because in the first MQGET call, the Msgld is populated with a valid value, so when the second MQGET is issued the queue manager acts like you are searching for a particular message which will

not be found.

Tom Dunlap then turned his attention to using trace for an application program. He then looked at CEDF screens for the MQ interface. He next turned his attention to statistics for the MQ interface.

Looking at an example, he explained that the top portion contained basic information about the interface including the name of the queue manager, its MQ release, name of the initiation queue, plus the connection status and when the connection was established. It also shows the current number of tasks using MQ and the number of calls issued while the connection was not

active (futile attempts). The next portion contained the total number of API calls issued, the number that have completed successfully, plus the name of the API crossing exit. The next portion contained the number of API calls by each type, from OPEN (MQOPEN) to SET (MQSET) requests. And the lower portion provided the number of internal MQ calls issued, number completed successfully, number of I/Os required, and the number of times a TCB switch was necessary.

Tom next looked at MQ resource definitions.

Figure 2 gives a simple view of how CICS will interface to resource managers

like IBM MQ or Db2. The key to working with other resource managers is the Task Related User Exit or TRUE programs. These are provided by CICS to provide the cross-region communication with the resource manager in the other address space.

The CICS application will execute the command to access one of the defined resource managers. These commands are the normal API functions expected by that resource manager, which include:

- CALL 'MQxxxx' – for WebSphere MQ requests
- EXEC SQL – for Db2 requests
- EXEC DLI – for IMS requests

CICS maintains a pool of Task Control Blocks or TCBs to process these requests, thus freeing up the normal CICS TCBs to perform other work. There is an SI parameter, MAXOPENTCBS, that specified the number of TCBs CICS could use. All of the work performed on these TCBs is “threadsafe” to minimize TCB switching and improve performance.

The CKAM transaction is the “alert monitor” transaction for the MQ adapter. The transaction handles unscheduled events that

are produced by the queue manager. This includes such events as the queue manager being shut down or waiting for the queue manager to start up.

The CKTI transaction is the CICS trigger monitor. The trigger monitors whether MQ performs MQGETs with wait on a define initiation queue (INITQ) associated with the application. In our case for the CICS region that CKTI is running in. When a trigger event occurs, the queue manager puts a message into the INITQ. This will wake up CKTI, which then analyzes the message contents (it is known as a trigger message) and will start the application it indicates.

Tom looked at these MQ resource definitions in some detail. He then explained SVC dump information for the MQ interface.

The upper portion of an “CICS MQ Interface Summary” from an SVC dump simply contains information about the MQ connection. This includes the name of the queue manager involved and its version. Another key piece of information is the name of the specified initiation queue. You can also see the current status of the connection. The “TRANSACTION SUMMARY” shows all the

tasks in the CICS region that have or are currently accessing MQ resources. Much of the information is of minimal interest, such as TcaAddr, TieAddr, and Uowid. The LotAddr is the control blocks used for the task during execution. The Tcb in MQ indicates whether the task currently has an MQ request outstanding.

Tom then looked in detail at other parts of the SVC dump information.

To finish up, Tom looked at the different screens associated with the CKQC transaction and how they could be used.

A copy of Tom Dunlap’s presentation is available for download from the Virtual CICS user group Web site at <http://fundi.com/virtualcics/presentations/CICSMQinterfaceMay19.pdf>

You can see and hear the whole user group meeting on YouTube at https://youtu.be/Vpn_ztynWN4.

Meeting dates

The following meeting dates have been arranged for the Virtual CICS user group:

- On 9 July 2019, Colin Pearce will be discussing, “How to solve CICS Enqueue problems”.

- The following meeting is on 10 September 2019, when Glenn Schneck, Principal Technical Architect at GT Software will be speaking.

We are using Webex for the user group meetings.

Recent CICS articles

All new Node.js in CICS Z Trial now available! by Natasha Mckenzie-Kelly on CICS DevCenter (26 April 2019). You can find the article at: <https://developer.ibm.com/cics/2019/04/26/node-js-in-cics-ztrial/>

New enhancements to CICS Performance Analyzer v540 by Satish Tanna on CICS DevCenter (29 March 2019). You can find the article at: <https://developer.ibm.com/cics/2019/03/29/new-enhancements-to-cics-performance-analyzer-v540/>

Managing enterprise-wide deployment of CICS Explorer by Dave Nice on CICS DevCenter (25 March 2019). You can find the article at: <https://developer.ibm.com/cics/2019/03/25/managing-enterprise-wide-deployments-of-cics-explorer/>

CICS news

IBM has announced CICS Statistics Visualizer V1.0.1

with more deployment options besides the original Docker deployment method. The tool allows users to check the health status of CICS regions faster by generating charts and graphs for their CICS statistics reports. Three types of report are available: Indicators Overview, Regions Overview, and Regions Trend Report. CICS Statistics Visualizer is provided free for CICS PA users only. More information can be found at <https://developer.ibm.com/cics/2019/03/08/cics-statistics-visualizer/>

Dynatrace has announced that it has extended Dynatrace, its AI-powered platform, to include IBM Z support for CICS, IMS, and middleware. This gives customers precise information about the performance of digital services across hybrid environments; from modern cloud applications to the mainframe. Dynatrace provides software intelligence to simplify enterprise cloud complexity and accelerate digital transformation. It provides end-to-end visibility by automatically discovering and mapping every transaction with a single AI-powered solution. This real time visibility, from cloud to the mainframe, gives enterprises a competitive advantage – they can

eliminate inefficiencies and consequently, innovate at a faster rate. More information can be found at <https://www.dynatrace.com/news/press-release/dynatrace-extends-ai-powered-software-intelligence-platform-to-hybrid-mainframe-environments/>

About the Virtual CICS user group

The Virtual CICS user group was established as a way for individuals using IBM's CICS TS systems to exchange information, learn new techniques, and advance their skills with the product.

The Web site at www.fundi.com/virtualcics provides a central point for coordinating periodic meetings (which contain technically-oriented topics presented in a webinar format), and provides articles, discussions, links, and other resources of interest to IBM CICS practitioners. Anyone with an interest in CICS is welcome to join the Virtual CICS user group and share in the knowledge exchange.

To share ideas, and for further information, contact trevor@itech-ed.com.

The Virtual CICS user group is free to its members.