



# 214

# CICS

*September 2003*

---

## In this issue

- 3 CICS TS 2.2 EJB Server JVM architecture and tuning tips
  - 14 Displaying task storage use in a CICS region during SOS to support CICS TS 1.3 regions
  - 16 MVS Logger and CICS system logs
  - 30 An EXCI application – enhanced
  - 46 CICS-XMITIP interface for sending e-mails without activating CICS TCP/IP environment – revisited
  - 49 CICS questions and answers
  - 51 CICS news
- 

# update

# ***CICS Update***

---

## **Published by**

Xephon  
27-35 London Road  
Newbury  
Berkshire RG14 1JL  
England  
Telephone: 01635 38342  
From USA: 01144 1635 38342  
E-mail: [trevore@xephon.com](mailto:trevore@xephon.com)

## **North American office**

Xephon  
PO Box 350100  
Westminster, CO 80035-0100  
USA  
Telephone: 303 410 9344

## **Subscriptions and back-issues**

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs £175.00 in the UK; \$270.00 in the USA and Canada; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the December 1999 issue, are available separately to subscribers for £16.00 (\$24.00) each including postage.

## ***CICS Update* on-line**

Code from *CICS Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at <http://www.xephon.com/cics>; you will need to supply a word from the printed issue.

## **Editor**

Trevor Eddolls

## **Disclaimer**

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, and other contents of this journal before making any use of it.

## **Contributions**

When Xephon is given copyright, articles published in *CICS Update* are paid for at the rate of £100 (\$160) per 1000 words and £50 (\$80) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £20 (\$32) per 100 lines. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from [www.xephon.com/nfc](http://www.xephon.com/nfc).

---

© Xephon plc 2003. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

*Printed in England.*

# CICS TS 2.2 EJB Server JVM architecture and tuning tips

## JAVA BACKGROUND UNDER CICS

Most companies today are moving towards using Java under CICS because they see Java as the language of the future, with its 'write once, run anywhere' capability.

CICS TS 1.3 was the first release to support Java code under CICS, and it was further enhanced in V2 to support CICS EJB Server. The Java bytecodes can be interpreted by the standard z/OS JVM or they can be compiled to machine code by the HPJ compiler that is provided with VisualAge for Java Enterprise Edition. The CICS Java programs can access CICS services and LINK to existing CICS legacy programs ( like COBOL or Assembler) using JCICS classes.

## CICS EJB SUPPORT IMPLEMENTATION

CICS EJB support differs from WebSphere EJB functionality. CICS is not intended to be fully J2EE compliant. WebSphere provides a fully J2EE compliant infrastructure. CICS provides EJB support because that is the part of the J2EE programming model that requires a robust, highly-scalable, secure transaction monitor. CICS does not support servlets, JSPs, or JavaMail, because CICS is not optimized for those functions. CICS is optimized for supporting Enterprise beans and, in particular, session beans. CICS currently does not support entity beans; neither does it support persistence of Enterprise beans.

Another big difference between CICS and WebSphere is JVM reuse. WebSphere allocates one JVM per address space and automatically adds more address spaces (up to the limit specified) when volume increases, whereas CICS adds TCBs, one per JVM, up to the limit set by MAXJVMTCBS.

## PERSISTENT REUSABLE JVM

CICS is well known as a high-performance system handling hundreds or thousands of transactions per second. Running Java code under CICS presents new challenges related to its relatively high CPU overhead and memory usage. In CICS TS 2.2, IBM implemented a way to avoid a full JVM initialization every time, by reusing the JVM on subsequent invocations. Multiple transactions can be executed serially in a single long-running JVM by using persistent reusable JVMs. The serial reuse of a JVM for multiple transactions, while resetting the JVM to a known state between each transaction, provides isolation without paying the high cost of a full JVM initialization for every transaction. There is only one transaction using a JVM at any one time. At transaction termination time, the JVM is made available for reuse by another transaction. There is also optimized garbage collection, which is enabled by the clean separation of short-lived application objects from the long-lived classes, native code, and objects.

A sample CICS JVM profile for a resettable JVM can be found in the dataset CICS.XDFHENV, member DFHJVMPR.

## JIT COMPILE

Java programs can execute in a JVM running on any supported platform through the ability of the JVM to interpret Java bytecode. You can create Java bytecode class files using a Java compiler, such as VisualAge for Java, and these classes can be executed by a JVM without the need for any further translation. The JIT-compile function of the JVM produces JIT-compiled versions of frequently-used Java methods. The JIT-compiling process incurs additional CPU time and uses extra LE storage, but provides more-efficient executable code. The IBM reusable JVM performs a JIT-compile after 2000 references to a class, which means that the CPU cost of the Java applications reduces after the JIT-compiled code is produced.

## THREADSAFE AND JVM SERIAL REUSE

Each CICS Java program runs as Threadsafe (new PPT setting – Concurrency : Threadsafe) and runs under its own TCB – J8. The priority of the J8-mode TCBs is set lower than that of the main CICS QR TCB to ensure that J8-mode tasks do not affect the main CICS workload being processed on the CICS QR TCB.

Multiple transactions are executed serially in a single long-running JVM. A JVM is dedicated to each CICS program for the lifetime of the CICS transaction. This is important to remember when planning to do workload balancing between multiple CICS AORs. CICS invokes the routing program for requests for methods that will run under a new OTS transaction, but not for requests for methods that will run under an existing OTS transaction; these it directs automatically to the AOR in which the existing OTS transaction runs. To balance method requests across AORs, you have the following two options:

- 1 Write a customized version of the CICS distributed routing program, DFHDSRP (source in .CICS.SDFHSAMP).
- 2 Use the distributed routing program, EYU9XLOP, provided with CICSplex SM.

## JVM CLASSES – MIDDLEWARE AND SHARED

In order to tune your CICS Java Application for best performance, you need to know that there are different JVM classes, dividing the classes contained in the JVM into distinct parts.

### Middleware

The middleware JVM consists of the system classes and standard Java extension classes, which provide the base services in the JVM. It provides services that access resources like JCICS classes, JNDI classes, CCF, CTG, and JDBC classes. This JVM is known as a Trusted Middleware class path and it is built automatically by CICS from the paths specified on the CICS\_DIRECTORY and JAVA\_HOME parameters. You can add your own JAR files to this directory by using the TMPREFIX

and TMSUFFIX parameters. The Trusted Middleware class should have classes supplied by IBM, and you may also choose to load your common vendor JAR files in order to improve performance. Putting JAR files in this class will prevent static initializers in JAR files being executed with every load of a JVM.

Trusted Middleware classpath should not be used for user applications. The danger is that applications would not be written in a sufficiently general way that they reset the JVM, ready for the next invocation.

Also note that classes defined in Middleware Class are loaded before the Shareable Application class.

Here is an example of how to append vendor JAR files *after* IBM supplied JAR files:

```
XDFHENV(DFHJVMPR)
```

```
TMSUFFIX X=/usr/lpp/cics/cics22/dvl p/eabl i b. jar: \  
/usr/lpp/cics/cics22/dvl p/recjava.jar
```

The Record Framework (recjava.jar) is a collection of predefined classes of objects that work together to handle record-oriented data in Java. The Record Framework calls a user-provided typeset.

The Enterprise Access Builder (EAB) for Transactions (EABlib.jar) feature of VisualAge for Java remains the primary tool for building applications that need to connect to a variety of back ends, including MQSeries, CICS, and IMS. It was originally built to use the VisualAge for Java Common Connector Framework (CCF), but now also supports J2EE connectors. To get the EAB run-time and J2EE Connector Architecture, download the Connector Architecture for WebSphere Application Server from the WebSphere Developer Domain. This free download gives you a ZIP file in which you can find five JAR files – recjava.jar, eablib.jar, j2c.jar, jca.jar, and ccf.jar.

## Shared

The shareable application code, consisting of the Enterprise

beans themselves, is designated for Application code. You should use JVM parameter `ibm.jvm.shareable.application.class.path` to define your JAR file to the shareable application classpath. Classes defined here take precedence over classes from the deployed JAR file. The Enterprise bean classes are also loaded into the application class system heap. They are located by reference to the DJAR definition in CICS.

Classes loaded from `ibm.jvm.shareable.application.class.path` are preserved in the JVM across resets, and, over time, can become compiled and cached by the JIT compiler. This improves performance.

Here is an example of how to define a JAR file in a Shared class in XDFHENV (DFJJVMR):

```
ibm.jvm.shareable.application.class.path=/usr/lpp/cics/cics22/dvip/your.jar
```

## JVM STORAGE MANAGEMENT

The next step to understanding how to tune a Java Application is to become familiar with JVM storage management and how it relates to JVM classes.

The reusable JVM manages run-time storage in several segregated heaps. This is similar to DSA's allocation, like RDSA, UDSA, CDSA, or SDSA.

A JVM in CICS runs as a Unix System Services process in a CEEPIPI enclave, using MVS LE services rather than CICS LE services. As a result, all storage obtained by the JVM is MVS storage, obtained by calls to MVS LE services. This storage resides within the CICS address space but is not included in DSAs.

You can override the LE run-time options and set your own values using the DFHSJJ80 user-replaceable module. You can find the source in CICS.SDFHSAMP.

To get LE statistics, update DFHSJJ80 as follows:

DC	C' RPT0(ON) '	Report LE Options
DC	C' RPTS(ON) '	Report LE Storage

This will generate an LE report when a JVM ends, as a part of CICS output. To force this report to be generated, issue the CEMT S JVM PHASEOUT command.

Please note that running with these options set will increase CPU usage. They should be used in a Test environment only.

Default settings in DFHSJJ8O are:

```

ANYHEAP(4096, 8176, ANYWHERE, FREE)
BELOWHEAP(4096, 2048, FREE)
HEAP(4194304, 1048576, ANYWHERE, FREE, 0, 4080)

```

Here is an example of how you may adjust the defaults to optimize storage usage, based on the LE HEAP storage report below:

```

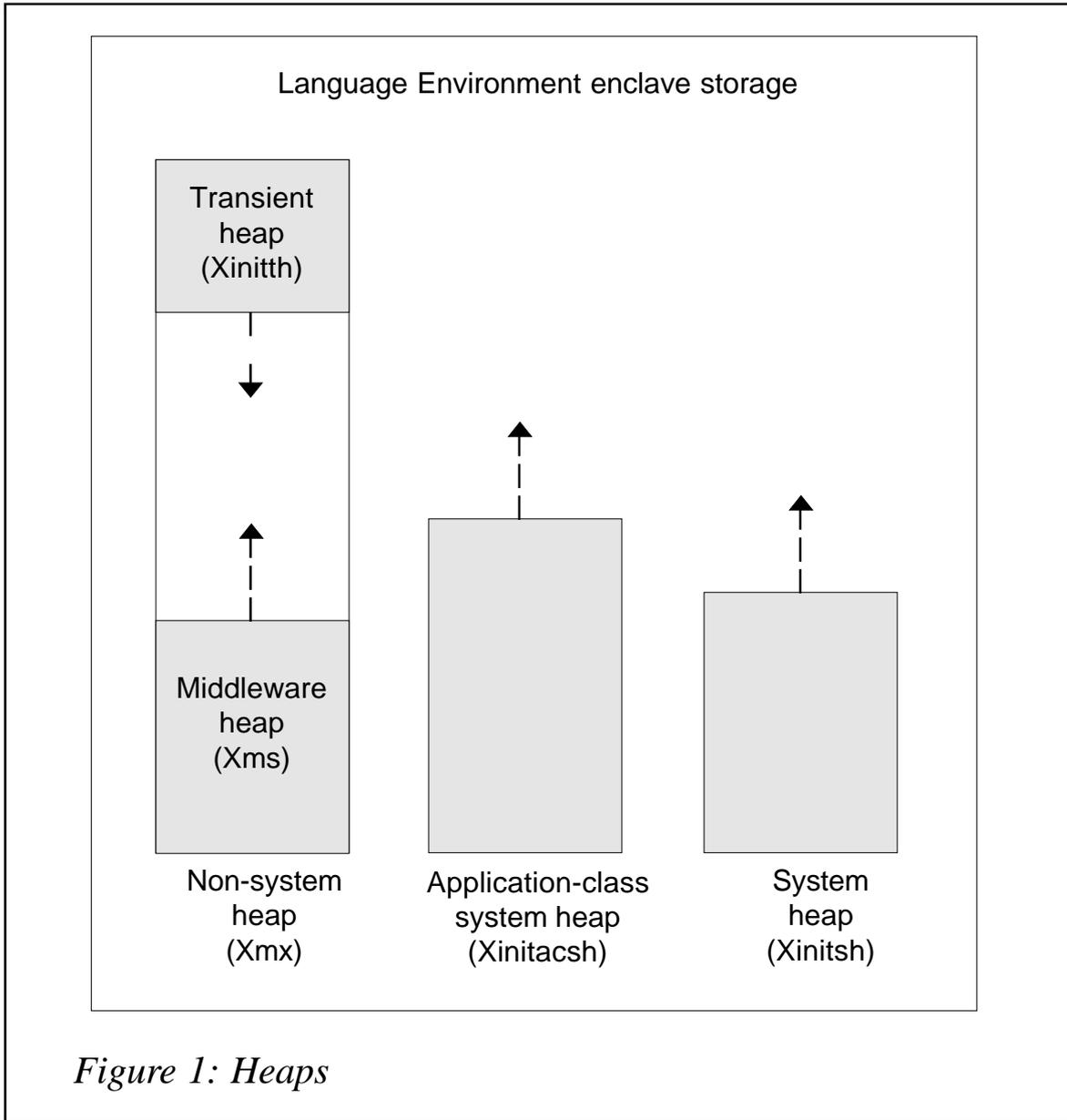
ANYHEAP(633160, 8176, ANYWHERE, FREE)
BELOWHEAP(2528, 2048, FREE)
HEAP(20888944, 1048576, ANYWHERE, FREE, 0, 4080)

```

```

HEAP statistics:
  Initial size:                4194304
  Increment size:              1048576
  Total heap storage used (sugg. initial size): 20888944  <=
  Successful Get Heap requests: 24974
  Successful Free Heap requests: 18652
  Number of segments allocated: 53
  Number of segments freed:    44
HEAP24 statistics:
  Initial size:                0
  Increment size:              4080
  Total heap storage used (sugg. initial size): 0
  Successful Get Heap requests: 0
  Successful Free Heap requests: 0
  Number of segments allocated: 0
  Number of segments freed:    0
ANYHEAP statistics:
  Initial size:                4096
  Increment size:              8176
  Total heap storage used (sugg. initial size): 633160  <=
  Successful Get Heap requests: 453
  Successful Free Heap requests: 169
  Number of segments allocated: 11
  Number of segments freed:    3
BELOWHEAP statistics:

```



Initial size:	4096	
Increment size:	2048	
Total heap storage used (sugg. initial size):	2528	<=
Successful Get Heap requests:	4	
Successful Free Heap requests:	3	
Number of segments allocated:	1	
Number of segments freed:	0	
Additional Heap statistics:		
Successful Create Heap requests:	4	
Successful Discard Heap requests:	0	
Total heap storage used:	34779136	
Successful Get Heap requests:	4	
Successful Free Heap requests:	0	
Number of segments allocated:	4	

Number of segments freed: 0  
Largest number of threads concurrently active: 5

The heaps used by JVM are as shown in Figure 1.

### Transient heap

The transient heap contains objects constructed by application classes. This heap is subject to JVM garbage collection, which involves clearing the heap when the application completes.

### Middleware heap

The middleware heap contains objects constructed by classes defined in the trusted middleware classpath. The objects in this heap are subject to normal garbage collection when no longer used. Middleware classes enable optimization through the caching of state to be used by multiple applications. This does improve performance, but you have to be very selective when determining which classes are eligible for the middleware heap. The heap is responsible for resetting itself correctly at the end of a transaction and, if necessary, reinitializing at the beginning of a new transaction in order to isolate different applications from each other.

### System heap

The system heap never goes through garbage-collection. It is composed of system classes, and classes loaded from the Trusted Middleware classpath, and contains classes loaded from the `ibm.jvm.shareable.application.class.path` and all classes loaded from the deployed JAR file.

Since this is never garbage-collected, applications do not have to be reloaded from the HFS if they are subsequently reused by another CICS transaction that reuses the JVM.

## JVM RESET

After a transaction has utilized a JVM, the JVM is reset, in that

the transient heap is cleared, but middleware and system heaps are maintained. This provides for a clean environment for the execution of the next transaction without the high initialization costs of starting a new JVM.

### **JVM parameters that control heap allocation size**

The middleware heap is allocated from low storage in the non-system heap and expands upwards; the transient heap is allocated from high storage in the non-system heap, and expands down towards low storage. The two heaps can expand only until they meet.

Set the Xmx value (maximum middleware and heap size) high, and set the Xms (initial middleware heap size) value to about 50% of the Xmx value. The JVM requires at least 20MB for the middleware heap to run an Enterprise bean, assuming a small transient heap of 500KB.

I suggest you set Xmx to 60MB (default in CICS is 32MB) and Xms to 30MB (default in CICS is 1MB). These settings will reduce the frequency of garbage-collection and heap size expansion, and will reduce the CPU overhead.

The Xinitth option (initial Transient heap size), is set to half the value of the Xms option (if xms is specified explicitly). Set Xinitth to 1MB, otherwise you will end up with 15MB, based on a 30MB Xms size. Be careful not to overallocate the transient heap size.

Leave the Xinitacsh and Xinitsh options to the default (see the *CICS Performance* guide). You can monitor heap expansions by analysing the data produced by the VERBOSE=gc JVM initialization option, and increase the initial sizes of these heaps only if a JVM requires the heaps to be expanded (indicated by a garbage-collection message).

Set VERBOSE=gc to collect output garbage-collection messages during your testing. Review the output from the last garbage-collection, written to the JVM stderr output file, to determine the high-water marks for the transient and middleware heaps.

## MAXJVMTCBS VALUE

To avoid MVS storage constraints under CICS, do not set the MAXJVMTCBS value too high!

This is a SIT parameter, which controls the maximum number of TCBs that can be allocated for JVMs – the J8 TCBs. The default is 5, which is a little too low for a busy system.

This parameter can be changed dynamically with CEMT S DIS MAXJVMTCBS(##). My recommendation is not to set it higher than 7. CICS statistics show that the average *Max TCB Pool Limit delay time* is 0.03 seconds, which is very low, considering the CPU and memory overhead of creating a new JVM. *Total Max TCB Pool Limit delay time* was also very low, in seconds, even when running high volumes with 2,000-3,000 requests queueing up.

So, the conclusion is: allow JVMs to queue up and do not overspecify JVM TCB pool limit.

CICS EJB code is very stable, but CPU and memory intense. Each JVM uses 55–100MB of Private storage above the line (depending on the Xmx value ) and around 68KB below.

Be aware that if, when you are testing in a single EJB Server region with MAXJVMTCBS set higher than 8, CPU becomes constrained with other workloads, all LE storage above the line can get exhausted, spilling below the line and, for all practical purposes, bringing down the region.

When this condition occurs, CICS statistics show a negative number for the Private area storage available below 16MB, and the following Java error messages are written to CICS dfhjvmerr log on HFS:

```
**Out of memory, aborting**
```

```
*** panic: JVMST017: Cannot allocate memory in  
initializeMarkAndAllLocBits(markbits1)
```

```
...
```

```
Unable to allocate an initial java heap of 1048576 bytes.
```

```
**Out of memory, aborting**
```

\*\*\* panic: JVMST016: Cannot allocate memory for initial java heap

When testing CICS TS 2.2 EJB server in a stand-alone environment under OS/390 V2.10, the following tuning changes resulted in drastic performance improvements, where response time went down from 2.2 seconds to 0.65-0.75 seconds, and CPU usage went down from 0.067–1.20 seconds to 0.016–0.044 seconds per transaction:

- 1 Xmx increased from 30M to 60M – less GC collection, less CPU.
- 2 Xms increased from 15M to 30M – less heap size expansion, less CPU.
- 3 JVM TCB limit decreased from 11 to 7 – fewer concurrent activates, preventing CPU going to 100%.
- 4 GC reporting turned OFF – less CPU, less I/O.
- 5 LE reporting turned OFF – less CPU, less I/O.
- 6 Application level logging turned off – less I/O.

And last, but not least, IBM just closed a PTF, PQ68286, which reduces the amount of I/O and CPU used during JVM initialization. The JVM initialization uses a large amount of CPU and performs a lot of file I/O. Because of this, JVM initialization can take several seconds to complete. This results in a very long response time for a user whose transaction has to initialize a JVM. This fix will make several changes to the CICS JVM support to reduce the impact of JVM initialization on performance.

You can reduce the start-up time for your JVMs by adding the following line to the JVM profiles that are used by your Java programs:

```
Xservi ce="-Xqui ckstart"
```

At JVM start-up, the -Xquickstart option causes the JIT compiler function of the JVM to run with a subset of optimizations – that is, a quick compile. However, when the JVM is subsequently reused by other Java programs, the JIT-compiling process runs

again to compile the code into its optimal form, and this will degrade performance for those programs. So by specifying -Xquickstart, you are reducing the initial start-up time for your JVMs, but causing a minor performance impact during later reuses of the JVMs.

---

*Elena Nanos*

*IBM Certified Solution Expert in CICS Web Enablement  
Zurich NA (USA)*

© Xephon 2003

---

## **Displaying task storage use in a CICS region during SOS to support CICS TS 1.3 regions**

The program MAPDSA, described in the article *Displaying task storage use in a CICS region during SOS*, published in *CICS Update* issue 186, May 2001, works on a CICS V4.1 region only. The essential logic of the program remains mostly unchanged for CICS TS 1.3 regions, but there are several offset changes necessitated by minor control block modifications.

This article shows how to modify program MAPDSA for CICS TS 1.3 and a non-fixed Kernel Anchor block so that it can be used in tandem with the updated version of MAPTCA described in *Displaying task activity in a CICS region under stress to support CICS TS 1.3 regions*, *CICS Update* issue 210, May 2003. No changes are required to the SOSCHECK REXX program also described in the May 2001 article.

The block of code 'INTO ACCESS MODE - R6 FOR CICS ADDRESS SPACE' ends with the following statement:

```
LAM    R6, R6, =F' 1'
```

After this statement, add the following block of code:

```

* -----
* FIND CICS KERNEL ANCHOR BLOCK
* -----
      L      R6, ASCBASXB          ASXB
      L      R6, X' 04' (, R6)    ASXBFTCB
PROCESS_TCB DS 0H
      LR     R8, R6                STORE TCB
      L      R6, X' D0' (, R6)    TCBEXT2
      L      R6, X' 14' (, R6)    SUBSYSTEM FACILITY CB (AFCB)
      LTR    R6, R6                AFCB POINTER?
      BNZ    PROCESS_AFCB         YES, USE IT
      LR     R6, R8                RESTORE TCB
      L      R6, X' 74' (, R6)    TCBTCB
      B      PROCESS_TCB
PROCESS_AFCB DS 0H
      AH     R6, X' 06' (, R6)    VECTOR LIST LENGTH
      LA     R6, X' 10' (, R6)    AFCB PREFIX LENGTH
      L      R6, X' 04' (, R6)    AFCS
      L      R6, X' 08' (, R6)    DFHKEKCB
      L      R6, X' 440' (, R6)   DFHSMANCHOR
* -----
* END OF NEW CODE
* -----

```

Immediately after this, the code between comment 'FOLLOW CICS STORAGE CHAINS' and label PROCESS\_SCA must be REPLACED with the following code:

```

* -----
* FOLLOW CICS STORAGE CHAINS
* -----
      LR     R5, R6                SAVE DFHSMANCHOR
      L      R6, X' 1C' (, R6)    LAST SCA
      LR     R3, R6
      LR     R6, R5                RESTORE DFHSMANCHOR
      L      R6, X' 18' (, R6)    FIRST SCA
PROCESS_SCA DS 0H

```

Optionally, as they are no longer required, the following two lines may be deleted:

```

KEKCB   DS   0F
        DC   XL4' 00006418'

```

---

*Patrick Mullen*  
*Consultant (Canada)*

© Xephon 2003

---

## MVS Logger and CICS system logs

All the information in this document was gathered from the following IBM manuals and reviews of our production systems.

The IBM manuals used are:

- *OS/390 V2R10.0 MVS Setting Up a Sysplex*
- *CICS TS 1.3 Installation Guide*
- *CICS TS 1.3 System Definition Guide*
- *CICS TS 1.3 Restart and Recovery Guide*
- *CICS TS 1.3 Performance Guide.*

### WHAT IS THIS DOCUMENT?

This document was intended to satisfy the following four objectives:

- 1 Provide myself with a better understanding of the MVS Logger since I did not fully understand it.
- 2 Provide some overview and training material for Operations about the MVS Logger.
- 3 Confirm the validity of my assessment of our MVS Logger implementation.
- 4 Provide a context to ask questions about our MVS Logger implementation for CICS system logs.

The format is a question and answer session that addresses operations issues in a progressive overview format. The following questions are posed and answered:

- 1 What is this document?
- 2 What is the MVS Logger?
- 3 What is a Logstream?

- 4 What can use the MVS Logger?
- 5 What functions does the MVS Logger perform?
- 6 How important is the MVS Logger to Operations?
- 7 Can I see the MVS Logger?
- 8 What resources does the MVS Logger manage?
- 9 How does CICS use the MVS Logger?
- 10 Does this affect normal CICS start-ups and shutdowns?
- 11 What happens to CICS if the MVS Logger breaks or CICS loses contact with its logstream?
- 12 Can I monitor logstreams?
- 13 What do you do if a CICS region crashes after a logstream failure?
- 14 How does the MVS Logger attempt to recover when failures occur?
- 15 What are the benefits and drawbacks of Coupling Facility logstreams?
- 16 What are the benefits and drawbacks of DASD-only logstreams?
- 17 What types of logstream do we use?

## WHAT IS THE MVS LOGGER?

The MVS Logger is an application provided by IBM to perform the common generic practice of application logging. The IXGLOGR address space is the MVS Logger.

Many applications (like SYSLOG, LOGREC, SMF, CICS, MQ, and DB2) maintain logs or journals for recovery, accounting, and/or auditing. The logging technique is very similar among applications that perform logging. Therefore, IBM provided a generic facility to perform this function. The MVS Logger also has the ability to coordinate shared logging activities between multiple

applications running on multiple CECs within a Sysplex. In providing the MVS Logger, IBM provided an opportunity to reduce the complexity of each application, reduce the individual points of failure, and provide a high-performance tool that can take advantage of Sysplex facilities (like the Coupling Facility) when appropriate. Applications using the MVS Logger will log their data to logstreams.

## WHAT IS A LOGSTREAM?

A logstream is a chunk of memory dedicated to an application solely for the purpose of logging data. If the MVS Logger implementation uses the Coupling Facility for logstreams, this chunk of memory resides inside a predefined structure in the Coupling Facility. If the MVS Logger implementation uses DASD-only logstreams, this chunk of memory resides in a dataspace attached to the IXGLOGR address space on the local CEC of the application using the MVS Logger. Coupling Facility logstreams are good when there is a need for a 'shared' log and/or Coupling Facility space is plentiful. DASD-only logstreams are good when the logstream cannot be shared between applications and/or Coupling Facility space is limited.

A structure is the name of the 'in memory' containers used within the Coupling Facility. Any product that uses the Coupling Facility must define one or more structures in the Coupling Facility. This is done by the systems programmers by defining a CFRM (Coupling Facility Resource Management) policy. The CFRM policy describes the attributes of the structures (size, location, etc). The policy is installed by running a batch job and activated using a SETXCF command. The CFRM policy is stored in the CFRM sysplex dataset. The name of our CFRM sysplex dataset is SYS1.CSPLEX03.CFRMnn (01 for primary and 02 for alternate) and can be displayed using the D XCF,COUPLE command.

A dataspace is the name of the 'in memory' container used by MVS to allow an address space to access more than 2GB of data in memory at a time within a single MVS image. Dataspaces look like address spaces but contain only data and are not executable.

You can issue a D A,name to see the dataspace attached to an address space. VTAM is a good example of a product that uses dataspace.

### WHAT CAN USE THE MVS LOGGER?

Some products have been modified to use the MVS Logger to perform their logging functions. In some cases we decided to implement these enhancements, in others we have not or the project is still on the drawing board. The products we have that can take advantage of the MVS Logger are:

- SYSLOG – can create a new sysplex-wide SYSLOG called OPERLOG.
- LOGREC – can create a new sysplex-wide LOGREC.
- CICS – can use MVS Logger to replace CICS Journals.

### WHAT FUNCTIONS DOES THE MVS LOGGER PERFORM?

The MVS Logger provides a high-performance, highly-available logging function. It provides the Application Programming Interfaces (APIs) and infrastructure to accomplish this on behalf of all users of the MVS Logger.

The MVS Logger can provide high-performance logging functions to 'interim' storage in either the Coupling Facility using logstreams or DASD-only logging using a dataspace on the local CEC. In both cases, the MVS Logger has the option to duplex the data to specially defined datasets called staging datasets and will always offload the interim storage to the more secure DASD offload datasets when usage exceeds user-defined high-use thresholds. This process of offloading data in memory (which is volatile) to secure DASD datasets is called 'hardening'.

The DASD offload datasets are rotated between similar DB2 active log datasets. These offload datasets are VSAM linear datasets and are managed by SMS. The offload datasets are allocated by the MVS Logger for the offload function when

'interim' storage reaches a high threshold condition and are deallocated when the user-defined low threshold is attained.

MVS supports many different file types (dataset organizations – DSORGs). The most common types are Physical Sequential (PS), Partitioned DataSets (PDS), Virtual Storage Access Method (VSAM), and Hierarchical File System (HFS). Each type was developed to store a different type of data. PS is good for backups and archives. PDSs are good for source code, documentation, and executable programs. VSAM is commonly used to store data. HFS is used for the Unix personality of MVS called Unix System Services (USS). There are four types of VSAM file: the Key Sequenced DataSet (KSDS), Entry Sequenced DataSet (ESDS), Relative Record DataSet (RRDS), and Linear DataSet (LDS). Like the other file types, the variety in VSAM was to handle different types of data more efficiently. We have chosen to use linear datasets to support all our DB2 tablespaces and MQ pagesets. A linear dataset is a high-performance VSAM dataset that MVS will automatically map onto the page datasets. MVS uses page datasets to move data in and out of memory and at a lower level in the I/O subsystem than traditional I/O. The MVS Logger also uses linear datasets for the offload datasets.

## HOW IMPORTANT IS THE MVS LOGGER TO OPERATIONS?

The logging function is very critical to most applications that use logging. CICS in particular will stop working if its logging function becomes unavailable. Therefore, the availability and performance of the MVS Logger is now a critical application to monitor.

## CAN I SEE THE MVS LOGGER?

The IXGLOGR address space is the MVS Logger. An MVS D A,IXGLOGR or SDSF DA;PRE IXGLOGR will display the address space. The MVS D LOGGER command will display its status and the resources it is currently managing. It is governed by a policy in the sysplex LOGR dataset. The MVS D XCF,COUPLE,TYPE=LOGR command will display the current sysplex LOGR dataset.

## WHAT RESOURCES DOES THE MVS LOGGER MANAGE?

The MVS Logger manages logstreams. Logstreams live inside a Coupling Facility structure if Coupling Facility logstreams are used. They live in a dataspace on a local CEC if DASD-only logstreams are used.

To display all structures, use the D XCF,STR command or D XCF,STR,STRNM=... to see a specific structure and its detail. All logstream structures are connected to the IXGLOGR address space. At our shop, the MVS Logger is used primarily for CICS TS 1.3 system logs. The only other thing using logstreams is a homegrown application logger. The structures used by CICS TS 1.3 logstreams are named:

- AOR\_TSxx\_LOG – primary CICS system log for AORs and RORs (one per CEC).
- AOR\_TSxx\_SHUNT – secondary CICS system log for AORs and RORs (one per CEC).
- CPSM\_CMASL\_LOG – primary CPSM system log.
- CPSM\_CMASL\_SHUNT – secondary CPSM system log.
- CPSM\_CMASS\_LOG – primary CPSM system log.
- CPSM\_CMASS\_SHUNT – secondary CPSM system log.
- QOR\_ELECL\_LOG – primary CICS system log for QORs.
- QOR\_ELECL\_SHUNT – secondary CICS system log for QORs.
- QOR\_ELECS\_LOG – primary CICS system log for QORs.
- QOR\_ELECS\_SHUNT – secondary CICS system log for QORs.
- TOR\_3270L\_LOG – primary CICS system log for TORs.
- TOR\_3270L\_SHUNT – secondary CICS system log for TORs.
- TOR\_3270S\_LOG – primary CICS system log for TORs.

- TOR\_3270S\_SHUNT – secondary CICS system log for TORs.

To display which logstreams are allocated to which structures, issue the D LOGGER,STR command. Our logstream names use the following naming convention:

- &CICS\_ID.&APPLID.DFHLOG
- &CICS\_ID.&APPLID.DFHSHUNT.

&CICS\_ID is the security ID for the CICS region:

- CICSRR for RORs
- CICSP0 for TORs and QORs
- CICSP1 for P1 AORs
- CICSP2 for P2 AORs
- CPCMAS for CPSM.

&APPLID is the VTAM APPLID for each region like:

- CIP1A1 – region CICSP1A1
- CIP2A4 – region CICSP2A4
- CIP2C8 – region C76AP2C8
- C01TP4T1 – region C01TP4T1.

The D LOGGER,C,LSN=\*,D command will show you all the details. It clearly shows which logstreams are in use by which address spaces:

```
D LOGGER,C,LSN=*,D
IXG6011 17.14.22  LOGGER DISPLAY 312
CONNECTION INFORMATION BY LOGSTREAM FOR SYSTEM CS01
LOGSTREAM          STRUCTURE          #CONN  STATUS
-----
CICSP0.CIP4Q1.DFHLOG  QOR_ELECL_LOG  000001 IN USE
STG DS: NO
JOBNAME: C01QP4Q1  ASID: 0079
R/W CONN: 000000 / 000001
RES MGR./CONNECTED: *NONE* / NO
IMPORT CONNECT: NO
```

This logstream, CICSPO.CIP4Q1.DFHLOG, is used by QOR C01QP4Q1 on TS01 in structure QOR\_ELECL\_LOG. It is currently IN USE and has one active connection. This logstream is not using Staging Datasets.

The MVS Logger also manages the DASD offload datasets. This is commonly handled by defining a MODEL definition when the logstream is defined. The creation and management of the DASD offload datasets is handled by SMS, and the ACS routines govern those datasets like any other dataset. The naming convention for our DASD offload datasets is:

```
TSPLEX03. &CICS_ACID. &APPLID. DFHLOG. Annnnnnn  
TSPLEX03. &CICS_ACID. &APPLID. DFHSHUNT. Annnnnnn
```

where *nnnnnn* is a sequential number to keep the datasets unique.

## HOW DOES CICS USE THE MVS LOGGER?

CICS uses the MVS Logger to act as the CICS system log. This function used to be performed by the Journal Control Program (JCP) and the datasets defined in the Journal Control Table (JCT). In CICS V4 and earlier, JCP would use the DFHJ01A, DFHJ01B, and DFHJ01X journals to handle the recovery, restart, and dynamic transaction backout activities. Now a new JOURNALMODEL is defined in the CSD to identify the Logstream names to be used and the new CICS Log Manager uses these logstreams to perform all the recovery, restart, and dynamic transaction backout functions. The DFHLOG and DFHSHUNT together make up the CICS system log. The CICS system log cannot be shared. Therefore there is always a one-to-one relationship between the CICS system log logstreams and a CICS region.

The CICS system definition file is the file where all CICS application changes are installed. This is most frequently done in batch (using program DFHCSDUP), but can be done on-line with the CEDA transaction. This is a CICS VSAM file that is the repository for all CICS definitions. The CSD file (DFHCSD), the

batch DFHCSDUP program, and the CEDA transaction make up the CICS Resource Definition Online (RDO) facilities.

As a refresher, CICS acts as a 'coordinator' in the two-phase commit relationship with DB2 (or any external resource manager). CICS uses a coordinated set of commit points (stored as log records) between the resource manager and CICS to determine whether to commit units of work or rollback units of work. When CICS has to rollback a unit of work, this is called Dynamic Transaction Backout (DTB). This activity is crucial to CICS' being able to ensure data integrity.

When CICS successfully takes an activity keypoint (syncpoint), all the prior log records are invalidated and deleted from the logstream. The logstream should be sized such that normal CICS keypoint activity avoids the triggering of the high-use threshold. IBM provides a CICS journal analysis program, DFHLSCU, to help determine the correct size of the logstream based on actual log data.

### DOES THIS AFFECT NORMAL CICS START-UPS AND SHUTDOWNS?

This should not affect normal CICS start-ups or shutdowns. CICS just has a different medium for System Log management. If a CICS region abends, it will reconnect to the logstream when it is restarted. Any in-flight units of work will be detected and the normal CICS recovery process will begin. As always, make sure the CICS region is started with START=AUTO so a CICS Emergency restart will occur. As always, a CICS region should never be cold started after an abend. Recovery data will be lost and data integrity will be in jeopardy.

There are five types of CICS restart with CICS TS 1.3. The first type is a START=WARM. CICS will restart using all information it stored from prior runs and restarts. The second type is START=EMERGENCY. This forces CICS to try to back out any in-flight work. This is what should be done after any abend. The

third type is START=AUTO. This lets CICS determine whether it needs a START=EMERGENCY or a START=WARM. This is how we have all our CICS regions set up. The fourth type is a START=COLD. This causes CICS to reload all the CICS definitions from the CSD (CICS System Definition file). A COLD start also clears any in-flight transaction history, which puts data integrity at risk, so this is inappropriate after an abend. The fifth type is a START=INITIAL. This is new with CICS TS 1.3 and does everything a COLD start does; it reinitializes the CICS Global Catalog, the CICS Local Catalog, and the CICS logstream. This is also inappropriate after an abend. A START=INITIAL will usually be done only under the supervision of a systems programmer.

Even though the MVS Logger may have offloaded logstream data to multiple DASD datasets, the recovery process is automatic. The MVS Logger keeps a directory of all DASD datasets available and in use by each logstream. The MVS Logger will allocate all the required datasets in the correct sequence on behalf of the application without human intervention. If DASD offload datasets are assigned to an SMS MGMTCLAS that allows them to migrate, this could take a while.

The other potential complication is the many-to-one mapping of logstreams to structures when Coupling Facility logstreams are used. A Coupling Facility logstream must reside in a Coupling Facility structure. This can be a one-to-one relationship or a many-to-one relationship (IBM recommends a maximum of 20). The Coupling Facility can hold a maximum of 255 structures so the number of CICS regions we have makes it prohibitive to use a one-to-one mapping. The drawback in this approach is that the Coupling Facility exploits parallelism at the structure level, not the logstream level. This means that each CICS will have to wait its turn during start-ups to serialize on the logstream structure. After this serialization is resolved the structure space is subdivided evenly among all the participants in the logstream structure.

## WHAT HAPPENS TO CICS IF THE MVS LOGGER BREAKS OR CICS LOSES CONTACT WITH ITS LOGSTREAM?

CICS will stop all activity if the MVS Logger becomes unavailable. An individual CICS region will crash if it loses contact with its logstream. Therefore, it is critical that Operations understand how to monitor the MVS Logger and the CICS logstreams.

Watch the console for DFHLG\* messages. These usually relate to the CICS Log Manager and the logstreams.

When it occurred, the MVS Logger attempted to offload the logstream and failed to allocate the DASD offload dataset owing to an HSM enqueue during volume space management. The logstream filled and CICS abended because it could not write any more log records.

It is also possible for the IXGLOGR address space to fail. IXGLOGR is an internally-started address space and is not controlled by automation. If IXGLOGR fails, it can be restarted with the MVS command:

```
S IXGLOGRS
```

The chances are that all users of the MVS Logger (all CICS regions) on the CEC where IXGLOGR failed will abend at that point. If this occurs, the CICS regions should be restarted with normal CICS restart procedures. Automation does a CICS START=AUTO restart unless it is the first CICS start-up on a Sunday morning when automation forces a START=COLD. The Coupling Facility will hold a copy of the logstream if Coupling Facility logstreams are used. The staging datasets will hold a copy of the logstream if DASD-only logstreams are used. Therefore the CICS regions should be able to reconnect to their logstreams and perform a normal Emergency Restart. If not, contact CICS Support before continuing.

## CAN I MONITOR LOGSTREAMS?

There are commands to watch the logstreams and all the underlying pieces. The three best tools with which to see

logstream information for CICS are the console, Omegamon/CICS, and MVS commands.

The console is already showing us CICS DFHLG\* messages. Watch for these and react quickly.

Omegamon/Omegaview can drill down into Omegamon/CICS (CUA version) to see the journals data. This will show the logstreams in use by a CICS region.

Finally, the most useful monitoring can be done with the MVS D LOGGER command (see *Quickref* or *MVS System Commands* manual for more information).

- D LOGGER – display the status of the Logger.
- D LOGGER,STR – display all structures in use by the Logger.
- D LOGGER,C – display logstream connections defined to the Logger.
- D LOGGER,C,LSN=\*,D – display all details of logstreams defined to the Logger.
- ROUTE \*ALL,D LOGGER,C,LSN=,D – display all logstream details across the Plex.

The most useful thing to monitor in these displays is #CONN and STATUS. The #CONN should be 000001 and the STATUS should be IN USE.

These displays also show the structure to logstream mapping for Coupling Facility logstreams.

## WHAT DO YOU DO IF A CICS REGION CRASHES AFTER A LOGSTREAM FAILURE?

If the DFHLG\* message (or Quickref description) tells you to restart the CICS region with START=INITIAL, you must check for INDOUBT DB2 units of work. Issue the following DB2 command on the system that the CICS failure occurred on:

-crc DIS THD(\*) TYPE(INDOUBT)

If there are any INDOUBT units reported, call DB2 support before restarting the region. The CICS region can then be started, but DB2 resources will be locked until DB2 support resolves the INDOUBT condition.

## HOW DOES THE MVS LOGGER ATTEMPT TO RECOVER WHEN FAILURES OCCUR?

The MVS Logger will always attempt to offload the data in 'interim' storage to DASD (if possible) before any recovery actions are taken.

First, in our configuration (two Coupling Facilities) the MVS Logger will duplex updates in the Coupling Facility logstream in an IXGLOGR dataspace on the local CEC. If either fails, it should be recoverable from the copy.

Second, if a Coupling Facility fails, normal Coupling Facility recovery begins and structures are rebuilt in the next available Coupling Facility assuming space permits. MVS Logger activity may be suspended while the rebuild is in progress. This may also be initiated by the Operator during a planned Coupling Facility fail-over for maintenance purposes (SETXCF START,REBUILD...).

Third, if the Coupling Facility survives the failure and a connection exists to the logstream structure from any application on any CEC defined to the logstream structure, the other CEC can initiate logstream recovery. Logstream recovery is simply initiating the offload process to 'harden' the log data.

In all cases (except the operator-initiated rebuild), this should be a 'hands off' event.

## WHAT ARE THE BENEFITS AND DRAWBACKS OF COUPLING FACILITY LOGSTREAMS?

Coupling Facility logstreams are best suited for applications that

require multiple CEC access to a shared log. The Coupling Facility logstream provides a consistent, reliable, interleaving log mechanism. It also provides for logstream recovery by an alternate CEC if any CEC in the Plex with an application connected to the logstream fails. Our logger application is a good example of this requirement.

The drawbacks are that logstreams use Coupling Facility space and the many-to-one mapping of logstreams to logstream structures causes a serialization on the logstream structure.

### WHAT ARE THE BENEFITS AND DRAWBACKS OF DASD-ONLY LOGSTREAMS?

DASD-only logstreams are better suited to applications that cannot share logstreams. Since the logstream is stored in a dataspace on the local CEC, performance should be slightly better than the Coupling Facility logstream.

The drawback is that more memory on the local CEC is used if the MVS Logger has not already duplexed the logstream data to the local CEC. If the environment is considered to be non-volatile (two Coupling Facilities), the MVS Logger automatically makes this decision and the memory is already in use. Since CICS System Logs cannot be shared, there is no operational benefit to Coupling Facility logstreams.

### WHAT TYPES OF LOGSTREAM DO WE USE?

We use both Coupling Facility logstreams and DASD-only logstreams. In DEVO, we use exclusively DASD-only logstreams primarily because of Coupling Facility space constraints. In Production, we currently use both Coupling Facility logstreams and DASD-only logstreams. All AORs use DASD-only logstreams. Only some of the special purpose regions use Coupling Facility logstreams in Production.

---

*Robert Zenuk*  
*Systems Programmer (USA)*

© Xephon 2003

---

## An EXCI application – enhanced

In *Deactivating and reactivating CICS* in *CICS Update* issue 201, August 2002 we found a way to connect to CICS from batch. The application is designed to deactivate and reactivate a running CICS system without a shutdown. EXCI technology is particularly versatile. EXCI could be used, for example, for the communication between CICS and batch in a 24-hour CICS system.

But back to the application from last August. Here are some improvements. Job aborts in batch have been replaced by return codes and in the CICS system transactions (even in the broader sense) have been excluded from the cancel command.

### CSPROJ (CICS PROGRAM)

```
*ASM XOPTS(CICS, SP)
CSPROJ  TITLE '*** Program to make CICS available for the BATCH ***'
        SPACE
*-----*
*                C S P R O J                *
*-----*
* Note:          Program must be linked with AMODE=ANY / RMODE=ANY          *
*                                                        *
* Comment:       This program is designed to make a CICS exclusively          *
*                available for BATCH processing or vice versa                  *
*                CSPROJ must be called via BATCH as an "EXCI" -              *
*                program.                                                      *
*                A COMMAREA with a length of four bytes with the              *
*                content STOP (BATCH - processing) or START (ONLI-            *
*                NE - operation) is mandatory.                                  *
*                If STOP is requested (BATCH - processing):                    *
*                1. VTAM-ACB will be closed.                                   *
*                2. Active (user) transactions will be purged.                *
*                3. User - datasets (VSAM) will be closed                      *
*                   (DD - name starting with "D").                             *
*                If STRT is requested (ONLINE - operation):                    *
*                1. User - datasets (VSAM) will be opened                      *
*                   (DD - name starting with "D").                             *
*                2. User - programs("CI*") will be reloaded.                 *
*                3. VTAM - ACB will be opened.                                 *
*                SOME INFORMATION MESSAGES DURING THIS PROCESS AT              *
*-----*
```

```

*          SYSTEM  CONSOLE ARE ALSO AVAILABLE.          *
*
*-----*
SPACE
* *****
* *****
EJECT
* *****
*          D e f i n i t i o n s          *
* *****
          SPACE
*          INCLUDE++ CSCWAA          INCLUDE CICS CWA ASSEMBLER STRUCTURE
          EJECT
*-----*
*          C I C S          C W A - R E G I O N          *
*          INCLUDE-ELEMENT  FOR ASM  PROGRAM CSCWAA          *
*          ADDRESSING      :  EXEC CICS ADDRESS CWA(CWAPTR)          *
*          WARNING         :  NO CHANGES ALLOWED -          *
*                               CWAPTR MUST NOT BE DEFINED          *
          SPACE 3
          USING CWADSECT, CWAPTR
          SPACE 3
CWADSECT  DSECT
          SPACE 3
CWAAREA   DS  ØCL1536          CWA-BEREICH
          SPACE 1
CWAIEYCA  DC  CL4' '          EYECATCHER
CWAISYS   DC  CL4' '          ZUGEORDNETE VSAM-SYSID (CWACICID)
CWAISYS   DC  CL4' '          ZUGEORDNETE TERM-SYSID (CWACICID)
CWAISYID  DC  CL4' '          ORIGINAL SYSTEM-ID
CWAISYID  DC  CL8' '          ORIGINAL APPLICATION-ID
CWAISYID  DC  AL4(Ø)          POINTER NLV-CVT
CWAISYID  DC  AL4(Ø)          ADRESSE D. CUA-TRANSAKTIONSTABELLE
CWAISYID  DC  ØCL4' '          CICS-LEVEL 'Ø311' OR 'Ø33Ø'
CWAISYID  DC  CL1' '          CICS-LEVEL
CWAISYID  DC  CL1' '          CICS-VERSION
CWAISYID  DC  CL1' '          CICS-RELEASE
CWAISYID  DC  CL1' '          CICS-MODIFICATION
CWAISYID  DC  PL4' Ø'          STOP-TIME FOR CMF-EVENTS HHMSSTC
          DS  XL56Ø          .....  FREI  .....
          DS  XL12Ø          .....  FREI  .....
          DS  XL22          .....  FREI  .....
CWACICTX  DC  CL4' '          CICS-ID-BESCHREIBUNG
CWACICID  DC  CL1' '          CICS-ID
CWA$PROD  EQU  C' P'          ..  PROD
CWA$TEST  EQU  C' T'          ..  TEST
CWA$VPRD  EQU  C' V'          ..  VORPROD
CWA$SYST  EQU  C' S'          ..  SYSTEM-CICS
CWACICNR  DC  CL1' '          CICS-NR
CWA$TERM  EQU  C' T'          ..  TERMINAL

```

CWA\$VSAM	EQU	C' V'	.. DATASET VSAM
CWA\$PAIS	EQU	C' P'	.. PAISY
CWA\$ODM	EQU	C' O'	.. ODM
CWA\$INFO	EQU	C' I'	.. INFO
CWA\$APPL	EQU	C' Ø'	.. APPLICATION ØØ-Ø9
* \$APPL	EQU	????	.. APPLICATION A-C
* \$APPL	EQU	????	.. APPLICATION E-0
* \$APPL	EQU	????	.. APPLICATION Q-S
* \$APPL	EQU	????	.. APPLICATION U-Z
CWADATUM	DC	CL8' '	DATUM FORMAT TT.MM.JJ
CWACTMJ	DC	CL6' '	DATUM TTMMJJ
CWAPTMJ	DC	PL4' Ø'	DATUM ØTTMMJJC
CWACJMT	DC	CL6' '	DATUM JJMMTT
CWAPJMT	DC	PL4' Ø'	DATUM ØJJMMTTC
CWACTMJ4	DC	CL8' '	DATUM TTMMJJJJ
CWAPTMJ4	DC	PL5' Ø'	DATUM ØTTMMJJJJC
CWACJ4MT	DC	CL8' '	DATUM JJJJMMTT
CWAPJ4MT	DC	PL5' Ø'	DATUM ØJJJJMMTTC
CWACMJ	DC	CL4' '	DATUM MMJJ
CWAPMJ	DC	PL3' Ø'	DATUM ØMMJJC
CWACJM	DC	CL4' '	DATUM JJMM
CWAPJM	DC	PL3' Ø'	DATUM ØJJMMC
CWACMJ4	DC	CL6' '	DATUM MMJJJJ
CWAPMJ4	DC	PL4' Ø'	DATUM ØMMJJJJC
CWACJ4M	DC	CL6' '	DATUM JJJJMM
CWAPJ4M	DC	PL4' Ø'	DATUM ØJJJJMMC
CWACT3J	DC	CL5' '	DATUM TTTJJ
CWAPT3J	DC	PL3' Ø'	DATUM TTTJJC
CWACJT3	DC	CL5' '	DATUM JJTTT
CWAPJT3	DC	PL3' Ø'	DATUM JJTTTC
CWACT3J4	DC	CL7' '	DATUM TTTJJJJ
CWAPT3J4	DC	PL4' Ø'	DATUM TTTJJJJC
CWACJ4T3	DC	CL7' '	DATUM JJJJTTT
CWAPJ4T3	DC	PL4' Ø'	DATUM JJJJTTC
CWAZEIT	DC	CL5' '	UHRZEIT SS:MM
*			-----
CWATABLE	DS	ØCL24	, +Ø123456789-, Ø123456789
*			-----
CWATAB1	DS	ØCL13	TABELLE 1 /, /+ /Ø123456789/- /
CWATAB2	DS	ØCL12	TABELLE 2 /, /+ /Ø123456789/
CWACHK01	DC	C' ,'	
CWATAB3	DS	ØCL12	TABELLE 3 /+ /Ø123456789/- /
CWATAB4	DS	ØCL11	TABELLE 4 /+ /Ø123456789/
CWACHARP	DC	C' +'	
CWATAB5	DS	ØCL12	TABELLE 5 /Ø123456789/- /, /
CWATAB6	DS	ØCL11	TABELLE 6 /Ø123456789/- /
CWATAB7	DS	ØCL1Ø	TABELLE 7 /Ø123456789/
CWACHØ9	DC	C' Ø123456789'	
CWACHARM	DC	C' -'	
CWATAB8	DS	ØCL11	TABELLE 8 /, /Ø123456789/

```

CWACHK02      DC    C' , '
CWACH092      DC    C' 0123456789'
CWAZEI TP     DC    PL4' 0'      UHRZEIT  HHMMSSCT
CWADAY        DC    CL10' '      WOCHENTAG
CWAMONTH      DC    CL9' '      MONAT
CWA_PTR_FTT   DC    AL4(0)      ADRESSE D. FUNKTIONSTASTENTABELLE
CWA_PTR_ANT   DC    AL4(0)      ADRESSE DER AKTIONSNAMENTABELLE
CWA_INFOCICS  DC    C' '      INFO-CICS IDENTIFIER
CWA_INFOCICS_Y EQU   C' Y'      INFO-CICS IDENTIFIER -JA-
CWA_INFOCICS_N EQU   C' '      INFO-CICS IDENTIFIER -NEIN-
CWA_DATUM_JJJJ DC   CL10' '      DATUM FORMAT TT.MM.JJJJ
              SPACE 1
CWAAREAE      EQU   *          ENDE CWA DEFINITIONEN
              SPACE 5

```

```

* -----
*           END OF THE CICS CWA_BEREICHES
* -----

```

```

* -----
*           START OF THE DSECT FOR FUNKTIONSTASTENTABELLE
* -----

```

```

*           ADRESSIERUNG UEBER "CWA_PTR_FTT"
* -----

```

```

CWAFTTDSECT   DSECT
CWA_FTT_TASTE DC    XL1' 0'      TASTENIDENTIFIKATION
CWA_FTT_AKTION DC   CL16' '      KURZBEZEICHNUNG DER TASTE
*                                     BSP. : HILFE
CWA_FTT_ANZEIGE DC  CL20' '      TEXT FUER DEN FUNKTIONS-
*                                     TASTENBLOCK EINES BILDES
*                                     BSP. : F1=HILFE
CWA_FTT_PFKY  DC    CL4' '      PF-TASTE Z.B. "PF1 "
CWA_FTT_KURZTEXT DC  CL8' '      TASTENKUERZEL FUER POP-UP-MENUS
*                                     BSP. : F12=ABBR
CWA_FTT_TEXT  DS    CL207        BESCHREIBUNG DER AKTION
CWAFTTDSECTE EQU   *
CWAFTTANZAHL EQU   30          ANZAHL TABELLENEINTRAEGE FTT
              SPACE 2

```

```

* -----
*           END OF DSECT FOR FUNKTIONSTASTENTABELLE
* -----

```

```

* -----
*           START OF DSECT FOR AKTIONSNAMENTABELLE
*           ADRESSIERUNG UEBER "CWA_PTR_ANT"
* -----

```

```

CWAANTDSECT DSECT
*
CWA_ANT_HI LFE      DS CL16      HILFETEXT
CWA_ANT_TASTEN     DS CL16      ANZEIGE DER TASTENBELEGUNG
CWA_ANT_AUSGANG    DS CL16      BEENDEN EINER FUNKTION

```

CWA_ANT_REFRESH	DS CL16	WIEDERHERSTELLEN
CWA_ANT_UPDATE	DS CL16	DATEN SPEICHERN
CWA_ANT RUECKWAERTS	DS CL16	RÜCKWÄRTS BLÄTTERN
CWA_ANT_VORWAERTS	DS CL16	VORWÄRTS BLÄTTERN
CWA_ANT_AKTION	DS CL16	AKTIVIERUNG ACTIONBAR
CWA_ANT_UNTERBRECHEN	DS CL16	VORGANGSUNTERBRECHUNG
CWA_ANT_ABBRUCH	DS CL16	ABBRUCH
CWA_ANT_EINSTIEG	DS CL16	ZURÜCK ZUM EINSTIEGSBILD
CWA_ANT_AUSWAHL	DS CL16	ZURÜCK ZUM AUSWAHLBILD
CWA_ANT_SICHERN	DS CL16	EINFRIEREN VON DATEN
CWA_ANT_LINKS	DS CL16	BLÄTTERN NACH LINKS
CWA_ANT_RECHTS	DS CL16	BLÄTTERN NACH RECHTS
CWA_ANT_ANFANG	DS CL16	ANZEIGE DER ERSTEN SEITE
CWA_ANT_SCHLUSS	DS CL16	ANZEIGE DER LETZTEN SEITE
CWA_ANT_ABMELDEN	DS CL16	ZSS-ABMELDUNG
CWA_ANT_DRUCKEN	DS CL16	DRUCKEN (PA2)
CWA_ANT_LOESCHEN	DS CL16	LÖSCHEN BILDSCHIRM
CWA_ANT_DATENFREIGABE	DS CL16	DATENFREIGABE
CWA_ANT_HILFE_ANLEGEN	DS CL16	BOSHELP HILFE ANLEGEN
CWA_ANT_SUCHEN	DS CL16	SUCHEN
CWA_ANT_EURODM	DS CL16	UMRECHNEN EURO/DM
CWAANTDSECTE	EQU *	

\*-----  
\*                   END OF DSECT FOR AKTIONSNAMENTABELLE                   \*  
\*-----

\*                   END INCLUDE++

SPACE		
DFHEI STG	DSECT	
NBR	DS    F	NBR OF TASKS
RESP	DS    F	RESPONSE-CODE
LENGTH	DS    H	OUTPUT LENGTH
PROGRAM	DS    CL8	RELOADED PROGRAMS
FILE	DS    CL8	FILES TO OPEN OR CLOSE
USRID	DS    CL8	USERID
TRAN	DS    CL4	TRANSACTION NAME
TASKNO	DS    F	TRANSACTIONS TO PURGE
ABCODE	DS    CL4	ABEND CODE
INAREA	DS    CL4	COMMAREA INPUT
OUTAREA	DS    CL7Ø	TERMINAL-OUTPUT-AREA

EJECT  
\* \*\*\*\*\*  
\*                   M a i n - p r o g r a m                   \*  
\* \*\*\*\*\*

SPACE		
CSPROJ	DFHEI ENT	CODEREG=(R3, R4), DATAREG=R12
CSPROJ	AMODE	ANY
CSPROJ	RMODE	ANY
EXEC	CICS ADDRESS	CWA                   (CWAPTR)                   *
	RESP	(RESP)
CLC	RESP, DFHRESP	(NORMAL)

	BE	CSPR0200	
	MVC	ABCODE, =C' CAAA'	SYSID can't be assigned
	BAS	R6, ABEND	
	B	ERRORWA	
	SPACE		
CSPR0200	EQU	*	
	EXEC	CICS ASSIGN USERID (USRID)	*
		RESP(RESP)	
	CLC	RESP, DFHRESP(NORMAL)	
	BE	CSPR0500	
	MVC	ABCODE, =C' USID'	USERID can't be assigned
	BAS	R6, ABEND	
	B	ERRORWA	
	SPACE		
CSPR0500	EQU	*	
	CLI	CWACICID, CWA\$SYST	Is it SYSTEM-CICS?
	BE	CSPR0700	YES: Check whether it's a
	CLI	CWACICID, CWA\$TEST	Is it SYSTEM-CICS?
	BE	CSPR0700	YES: Check whether it's a
	SPACE		
CSPR0550	EQU	*	
	MVC	OUTAREA(50), MSG018	CICS address space not valid
	BAS	R7, SENDMSG	
	B	ERRORWA	
	SPACE		
CSPR0700	EQU	*	
	CLI	CWACICNR, CWA\$TERM	Is it a TOR?
	BE	CSPR0550	YES: Invalid CICS
	CLI	CWACICNR, CWA\$VSAM	Is it a FOR?
	BE	CSPR0550	YES: Invalid CICS
	CLC	CWASYSID(2), =C' SS'	Is it a SYSTEM-AOR's?
	BE	CSPR0800	YES: Can running in each AOR
	CLC	CWASYSID, =C' ST09'	Is it a TEST-AOR's and a CICS
*			namend "Project-CICS"?
	BNH	CSPR0550	No: Invalid CICS
	SPACE		
CSPR0800	EQU	*	
	EXEC	CICS ADDRESS COMMAREA(R2)	*
		RESP(RESP)	
	CLC	RESP, DFHRESP(NORMAL)	
	BE	CSPR1000	
	MVC	OUTAREA(50), MSG000	No access to the COMMAREA
	BAS	R7, SENDMSG	
	B	ERRORWA	
	SPACE		
CSPR1000	EQU	*	
	CLC	EI BCALEN, =H' 4'	
	BE	CSPR2000	
	MVC	OUTAREA(50), MSG001	Invalid COMMAREA detected
	BAS	R7, SENDMSG	

```

B      ERRORWA
SPACE
CSPR2000 EQU  *
MVC    I NAREA, 0(R2)           Move COMMAREA - content
CLC    I NAREA, =C' STOP'
BNE    CSPR3000
BAS    R6, CLVTAACB           BATCH - processing requested
BAS    R6, DELAY              Delay processing for a second
BAS    R6, PURGTRAN
BAS    R6, DELAY              Delay processing for a second
BAS    R6, CLOSEVSA
B      RETURN
SPACE
CSPR3000 EQU  *
CLC    I NAREA, =C' STRT'
BNE    CSPR9000
BAS    R6, OPENVSAM           ONLINE - operation requested
BAS    R6, DELAY              Delay processing for a second
BAS    R6, RELOADPG
BAS    R6, DELAY              Delay processing for a second
BAS    R6, OPVTAACB
B      RETURN
SPACE
CSPR9000 EQU  *
MVC    OUTAREA(50), MSG002     Wrong processing characters
MVC    OUTAREA+36(L' I NAREA), I NAREA
BAS    R7, SENDMSG
B      ERRORWA
SPACE
RETURN EQU  *
EXEC   CICS RETURN
EJECT
* ***** *
*      S u b r o u t i n e s      *
* ***** *
SPACE
* ***** *
*      V T A M - A C B w i l l b e c l o s e d      *
* ***** *
SPACE
CLVTAACB EQU  *
MVC    OUTAREA(50), MSG003
BAS    R7, SENDMSG
EXEC   CICS SET VTAM CLOSED      *
      RESP(RESP)
CLC    RESP, DFHRESP(NORMAL)
BNE    ERRORWA
MVC    OUTAREA(50), MSG004
BAS    R7, SENDMSG
BR     R6

```

```

SPACE
* *****
* Purge all active transactions
* *****
SPACE
PURGTRAN EQU *
MVC OUTAREA(50), MSG005
BAS R7, SENDMSG
*
EXEC CICS INQUIRE TASK LIST *
LISTSIZE(NBR) *
SET(R9) *
RESP(RESP)
*
CLC RESP, DFHRESP(NORMAL) Get the task list.
BNE ERRORWA Was there a problem?
L R5, NBR Yes. Take the error exit.
LTR R5, R5 No. Get the number of tasks.
BZ PUR230 Are there any tasks?
SPACE No, so start the next phase.
PUR210 EQU *
MVC TASKNO, 0(R9) Yes. Store the task number.
CLC EIBTASKN, 0(R9) If its this task..
BE PUR220 ..ignore it
* get the task details.
EXEC CICS INQUIRE *
TASK(TASKNO) *
TRANSACTION(TRAN) *
RESP(RESP)
*
CLC RESP, DFHRESP(NORMAL) Go and force purge the task.
BNE PUR220 Is the task still there?
CLC TRAN, =C'CSKL' No. So nothing to force purge.
BE PUR220 Is it the listener from TCP/IP?
CLC TRAN, =C'ISER' Yes: Don't purge
BE PUR220 Is it the ISER transaction?
CLC TRAN(2), =C'CK' Yes: Don't purge
BE PUR220 Is it the MQSeries?
CLC TRAN(2), =C'TM' Yes: Don't purge
BE PUR220 Is it the TMON/CICS?
* Go and force purge the task.
EXEC CICS SET TASK(TASKNO) *
FORCEPURGE *
RESP(RESP)
*
CLC RESP, DFHRESP(NORMAL) ..say that force purge failed.
BE PUR220 If everthing is ok..
MVC OUTAREA(50), MSG019 carry on,
MVC OUTAREA+16(L' TASKNO), TASKNO Move Tasknumber
MVC OUTAREA+27(L' TRAN), TRAN Move Transactionname

```

```

      BAS  R7, SENDMSG
      SPACE
PUR220 EQU  *
      AH  R9, =H' 4'           Move on to the next task
      BCT R5, PUR210          Loop back if there is more to do
      SPACE
PUR230 EQU  *
      MVC  OUTAREA(50), MSG006
      BAS  R7, SENDMSG
      BR   R6
      SPACE
* *****
*   U s e r - d a t a s e t s   w i l l   b e   c l o s e d   *
* *****
      SPACE
CLOSEVSA EQU  *
      MVC  OUTAREA(50), MSG007
      BAS  R7, SENDMSG
      EXEC CICS INQUIRE FILE START          *
           RESP(RESP)
      CLC  RESP, DFHRESP(NORMAL)
      BE   CLO1000
      MVC  ABCODE, =C' CINS'
      BAS  R6, ABEND
      B    ERRORWA
      SPACE
CLO1000 EQU  *
      EXEC CICS INQUIRE FILE(FILE) NEXT      *
           RESP(RESP)
      CLC  RESP, DFHRESP(END)                Is it "End-Condition" ?
      BE   CLO8000                          Yes: Stop processing
      CLC  RESP, DFHRESP(NORMAL)            Is it "Normal-Condition" ?
      BE   CLO2000                          No: Abend taks
      MVC  ABCODE, =C' CNEX'
      BAS  R6, ABEND
      B    ERRORWA
      SPACE
CLO2000 EQU  *
      CLI  FILE, C' D'                      Is it a User-Dataset?
      BNE  CLO1000                          NO: No action
      CLC  FILE(3), =C' DAI'                Is it a PAISY-Dataset?
      BE   CLO1000                          YES: No action
      CLC  FILE(3), =C' DFH'                Is it a CICS-Dataset?
      BE   CLO1000                          YES: No action
      EXEC CICS SET FILE(FILE) CLOSED      *
           RESP(RESP)
      CLC  RESP, DFHRESP(NORMAL)            Is it "Normal-Condition" ?
      BE   CLO1000                          Yes: Next file
      MVC  OUTAREA(50), MSG016             Move message
      MVC  OUTAREA+26(L' FILE), FILE       Move DD-Name

```

```

      BAS  R7, SENDMSG
      B    CLO1000
      SPACE
CLO8000 EQU  *
      EXEC CICS INQUIRE FILE END          *
              RESP(RESP)
      CLC  RESP, DFHRESP(NORMAL)
      BE   CLO9000
      MVC  ABCODE, =C' CI NE'
      BAS  R6, ABEND
      B    ERRORWA
      SPACE
CLO9000 EQU  *
      MVC  OUTAREA(50), MSG008
      BAS  R7, SENDMSG
      BR   R6
      SPACE
* ***** *
*   U s e r - d a t a s e t s   w i l l   b e   o p e n e d   *
* ***** *
      SPACE
OPENVSAM EQU  *
      MVC  OUTAREA(50), MSG009
      BAS  R7, SENDMSG
      EXEC CICS INQUIRE FILE START      *
              RESP(RESP)
      CLC  RESP, DFHRESP(NORMAL)
      BE   OPE1000
      MVC  ABCODE, =C' OI NS'
      BAS  R6, ABEND
      B    ERRORWA
      SPACE
OPE1000 EQU  *
      EXEC CICS INQUIRE FILE(FILE) NEXT *
              RESP(RESP)
      CLC  RESP, DFHRESP(END)           Is it "End-Condition" ?
      BE   OPE8000                       Yes: Stop processing
      CLC  RESP, DFHRESP(NORMAL)       Is it "Normal-Condition" ?
      BE   OPE2000                       No: Abend taks
      MVC  ABCODE, =C' ONEX'
      BAS  R6, ABEND
      B    ERRORWA
      SPACE
OPE2000 EQU  *
      CLI  FILE, C' D'                   Is it a User-Dataset?
      BNE  OPE1000
      CLC  FILE(3), =C' DAI '           Is it a PAISY-Dataset?
      BE   OPE1000                       YES: No action
      CLC  FILE(3), =C' DFH'          Is it a CICS-Dataset?
      BE   OPE1000                       YES: No action

```

```

EXEC CICS SET FILE(FILE) OPEN                                *
      RESP(RESP)
CLC  RESP,DFHRESP(NORMAL)  Is it "Normal-Condition" ?
BE   OPE1000               Yes: Next file
MVC  OUTAREA(50),MSG015    Move message
MVC  OUTAREA+25(L'FILE),FILE Move DD-Name
BAS  R7,SENDMSG
B    OPE1000
SPACE
OPE8000 EQU *
EXEC  CICS INQUIRE FILE END                                *
      RESP(RESP)
CLC  RESP,DFHRESP(NORMAL)
BE   OPE9000
MVC  ABCODE,=C' OI NE'
BAS  R6,ABEND
B    ERRORWA
SPACE
OPE9000 EQU *
MVC  OUTAREA(50),MSG010
BAS  R7,SENDMSG
BR   R6
SPACE
* ***** *
*   U s e r - p r o g r a m s   w i l l   b e   r e l o a d e d   *
* ***** *
SPACE
RELOADPG EQU *
MVC  OUTAREA(50),MSG011
BAS  R7,SENDMSG
EXEC  CICS INQUIRE PROGRAM START AT('CI')                *
      RESP(RESP)
CLC  RESP,DFHRESP(NORMAL)
BE   RELOA000
MVC  ABCODE,=C' STAR'
BAS  R6,ABEND
B    ERRORWA
SPACE
RELOA000 EQU *
EXEC  CICS INQUIRE PROGRAM(PROGRAM) NEXT                  *
      RESP(RESP)
CLC  RESP,DFHRESP(NORMAL)
BE   RELOA500
CLC  RESP,DFHRESP(END)
BE   RELOA600
MVC  ABCODE,=C' NEXT'
BAS  R6,ABEND
B    ERRORWA
SPACE
RELOA500 EQU *

```

```

CLC    PROGRAM(2), =C' CI '
BNE    RELOA6000
EXEC   CICS SET PROGRAM(PROGRAM) PHASE IN                *
      RESP(RESP)
CLC    RESP, DFHRESP(NORMAL)
BE     RELOA0000
MVC    OUTAREA(50), MSG017                               Move message
MVC    OUTAREA+19(L' PROGRAM), PROGRAM Move program - name
BAS    R7, SENDMSG
B      RELOA0000
SPACE
RELOA6000 EQU *
EXEC   CICS INQUIRE PROGRAM END                          *
      RESP(RESP)
CLC    RESP, DFHRESP(NORMAL)
BE     RELOA9000
MVC    ABCODE, =C' ENDE'
BAS    R6, ABEND
B      ERRORWA
SPACE
RELOA9000 EQU *
MVC    OUTAREA(50), MSG012
BAS    R7, SENDMSG
BR     R6
SPACE
* *****
*      V T A M - A C B w i l l b e o p e n e d                *
* *****
SPACE
OPVTAACB EQU *
MVC    OUTAREA(50), MSG013
BAS    R7, SENDMSG
EXEC   CICS SET VTAM OPEN                                *
      RESP(RESP)
CLC    RESP, DFHRESP(NORMAL)
BNE    ERRORWA
MVC    OUTAREA(50), MSG014
BAS    R7, SENDMSG
BR     R6
SPACE
* *****
*      S e n d m e s s a g e t o c o n s o l e                *
* *****
SPACE
SENDMSG EQU *
MVC    LENGTH, =H' 70'
MVC    OUTAREA+52(L' USRI D), USRI D
MVC    OUTAREA+61(L' CWAPPLI D), CWAPPLI D
EXEC   CICS WRITE OPERATOR TEXT(OUTAREA)                *
      RESP(RESP)

```

```

CLC    RESP, DFHRESP(NORMAL)
BNE    ERRORWA
BR     R7
SPACE
* *****
*      E r r o r   w i t h   a b e n d   0C1 (operation exception)
* *****
SPACE
ERRORWA EQU    *                Error with abend 0C1
        DC     D' 0'           Never come back statement
        BR     R6
SPACE
* *****
*      A b e n d   w i t h   a b e n d   c o d e   "ABCODE"
* *****
SPACE
ABEND   DS     0H
        EXEC   CICS ABEND ABCODE(ABCODE)
        BR     R6
SPACE
* *****
*      D e l a y   p r o c e s s i n g   f o r   o n e   s e c o n d
* *****
SPACE
DELAY   DS     0H
        EXEC   CICS DELAY INTERVAL(1)
        BR     R6
        EJECT
* *****
*      R e g i s t e r   e q u a t e s   a n d   c o n s t a n t s
* *****
SPACE
EQUIREG
CWAPTR EQU    R8
SPACE
MSG000 DC     CL50' CSPROJ-000 Access to commarea impossible
MSG001 DC     CL50' CSPROJ-001 Invalid commarea length detected
MSG002 DC     CL50' CSPROJ-002 Invalid commarea content xxxx
MSG003 DC     CL50' CSPROJ-003 VTAM-ACB will be closed
MSG004 DC     CL50' CSPROJ-004 VTAM-ACB successfully closed
MSG005 DC     CL50' CSPROJ-005 Active transactions will be purged
MSG006 DC     CL50' CSPROJ-006 Active transactions successfully purged
MSG007 DC     CL50' CSPROJ-007 User - datasets will be closed
MSG008 DC     CL50' CSPROJ-008 User - datasets successfully closed
MSG009 DC     CL50' CSPROJ-009 User - datasets will be opened
MSG010 DC     CL50' CSPROJ-010 User - datasets successfully opened
MSG011 DC     CL50' CSPROJ-011 User - programs will be reloaded
MSG012 DC     CL50' CSPROJ-012 User - programs successfully reloaded
MSG013 DC     CL50' CSPROJ-013 VTAM-ACB will be opened
MSG014 DC     CL50' CSPROJ-014 VTAM-ACB successfully opened

```

```

MSGØ15  DC    CL5Ø' CSPROJ-Ø15 Open for file xxxxxxxx not possible  '
MSGØ16  DC    CL5Ø' CSPROJ-Ø16 Close for file xxxxxxxx not possible  '
MSGØ17  DC    CL5Ø' CSPROJ-Ø17 Program xxxxxxxx can not be reloaded  '
MSGØ18  DC    CL5Ø' CSPROJ-Ø18 CICS address space not valid         '
MSGØ19  DC    CL5Ø' CSPROJ-Ø19 Task nnnn (name xxxx) can not purged  '
EJECT
* *****
*           L i t e r a l s
* *****
          SPACE
          LTORG
          SPACE
          DC      C' '
          END

```

## CSPROJEX (BATCH PROGRAM)

```

*ASM XOPTS(EXCI)
*****
* MODULE NAME = CSPROJEX
* PROGRAM MUST BE LINKED WITH AMODE(31) AND RMODE(24) !
* DESCRIPTIVE NAME = Assembler Batch client program
* -----
* Author:          Claus Reis  Nuernberger Lebensversicherung AG
* Sample:          Use IBM's SDFHSAMP - library (member DFH$AXCC)
* STATUS = 5.3.Ø
* -----
* FUNCTION =      Connect a CICS via the Batch and run transaction PROJ
*                  with program CSPROJ.
*                  Go back to the Batch.
* -----
* ENTRY POINT = CSPROJEX
* INPUT PARAMETERS:
*   The only input to this program is supplied by the JCL PARM
*   parameter. This must contain the applid of the target CICS
*   system, a comma and a parm like STOP or STRT.
*****
          EJECT
*****
*   Include the standard copybook for EXCI return codes and DSECTS.
*****
          SPACE
          COPY DFHXCRCO
          EJECT
*****
*   Include the standard copybook for EXCI Parameter list equates.
*****
          SPACE
          COPY DFHXCPLD

```



```

L      R15, EXEC_RESP
SPACE
*****
*      Terminate program
*****

SPACE
GOBACK EQU   *
        DFHEI RET RCREG=(R15)
SPACE
*****
*      Terminate program after processing characters aren't valid
*****

SPACE
INVPRCHA EQU   *
*      ABEND 2222, DUMP
*      DS    D' 0'
        LH    R15, =H' 2222'
        B     GOBACK
SPACE
*****
*      Terminate program after call failed
*****

SPACE
LINKFAIL EQU   *
*      ABEND 3333, DUMP
*      DS    D' 0'
        LH    R15, =H' 3333'
        B     GOBACK
        EJECT
*****
*      Constants.
*****

SPACE
TARGET_PROGRAM DC CL8' CSPROJ  '
TARGET_TRANSID DC CL4' PROJ'
EYESYS         DC C' ###EYESYS###'
TARGET_SYSTEM  DC CL8'         '
TARGET_USERID  DC CL8'         '
SPACE
EXEC_INFO      DS 0F              EXEC level specific information
EXEC_DAT_LEN   DC AL2(4)          Outbound length
EXEC_COM_LEN   DC AL2(4)          Inbound length
EXEC_RETAREA   DS CL(EXCI_EXEC_RETURN_CODE__LEN)
SPACE
ZERO           DC F' 0'
SPACE
EYECOM         DC C' ###EYECOM###'
COMMAREA       DS 0CL4
PROCHAR        DS CL4
ORG            COMMAREA+4

```

```
SPACE
DFHREGS
SPACE
END      CSPROJEX
```

---

*Claus Reis*  
*CICS Systems Programmer*  
*Nuernberger Lebensversicherung AG (Germany)*

---

© Xephon 2003

## **CICS-XMITIP interface for sending e-mails without activating CICS TCP/IP environment – revisited**

*Editor's note: the original article, CICS-XMITIP interface for sending e-mails without activating CICS TCP/IP environment, was published in the May 2003 edition (issue 210) of CICS Update.*

While running XFHSMAIL in my production environment recently, I discovered a bug. Sometimes the program calling XFHSMAIL abends at the next EXEC instruction after linking to XFHSMAIL. This bug is caused by erroneous memory management performed by XFHSMAIL, which overwrites the instruction operand of the next EXEC instruction. I've corrected the program with some slight changes.

Summary of corrections:

- I've added DSECT for MSG\_BUFFER and inserted some changes to variable definitions in MSG\_BUFFER and JOB\_DSECT DSECT:

```
*****
* Message declarations.
*****
MSG_BUFFER      DSECT ,
MSG_BUFFER_R1  DS      ØCL128           Message buffer: 1st row
[Left unchanged until ...]
                ORG    MSG_TEXT
OK_MSG_AUTO    DS      ØCL46
                DS      CLØ4           Job
```

```

MSG_JOBNAME      DS      CL08      Job name
                  DS      CL34
*
                  ORG      MSG_TEXT
ERROR_MSG_AUTO  DS      0CL46
                  DS      CL09
ERROR_TEXT       DS      CL37
*
MSG_BUFFER_R2   DS      0CL128      Message buffer: 2dn row
                  DS      CL07
MSG_MAILMSG     DS      CL80      Message text
                  DS      CL41
MSG_BUFFER_LEN  EQU      *-MSG_BUFFER
*
*****
* Job DSECT.
*****
JOB_DSECT        DSECT ,
JOB_NAME         DS      CL02      //
                  DS      0CL08      Job name
                  DS      CL02
JOB_NAME_SYSID  DS      CL04
                  DS      CL02
                  DS      CL06
                  DS      CL13      Account
                  DS      CL08      ), CLASS=
JOB_CLASS        DS      CL01
                  DS      CL10      , MSGCLASS=
JOB_MSGCLASS     DS      CL01
                  DS      CL31
                  DS      CL02      ,
                  DS      CL04      //
                  DS      CL07      4 BLANKS
                  DS      CL05      REGION=
JOB_REGION       DS      CL06      Region value
                  DS      CL08      , USER=
JOB_USER         DS      CL08      User
                  DS      CL48
                  DS      CL400
                  DS      CL12      . / ADD NAME=
JOB_I EBUPDTE_M DS      CL08      Message member name
                  DS      CL60      , LIST=ALL
JOB_I EBUPDTE_X DS      CL80      Message
                  DS      CL80      ><
                  DS      CL80      /**
                  DS      CL560
                  DS      CL12      %CICSMAIL
JOB_CICSMAIL_A  DS      CL08
                  DS      CL01
JOB_CICSMAIL_F  DS      CL08

```

```

                DS      CL01
JOB_CICSMAIL_S DS      CL08
                DS      CL01
JOB_CICSMAIL_M DS      CL08
                DS      CL33
                DS      CL80
JOB_DSECT_END EQU     *
*

```

- I've added a GETMAIN and register addressability for MSG\_BUFFER DSECT between the 'Addressing commarea' step and LBASSIGN label and deleted instructions for initializing MSG\_BUFFER\_R1 and MSG\_BUFFER\_R2 in LBASSIGN step:

```

                USING COMMAREA, R2
*
*****
* Addressing MSG_BUFFER.
*****
                XR      R5, R5
                EXEC    CICS GETMAIN                                *
                        SET(R5)                                    *
                        LENGTH(=Y(MSG_BUFFER_LEN))                *
                        INITIMG(SPACE)
                USING  MSG_BUFFER, R5
*
*****
* Obtain the program name and CICS applid for messages.
* In addition, obtain the startcode type.
*****
LBASSIGN EQU     *
                MVC    FIRST_MSG_AUTO, FIRST_MSG
                MVC    MSG_BUFFER_R2, SECOND_MSG
                MVC    MSG_MAILMSG, CMESSAGE

```

- I've added a FREEMAIN for MSG\_BUFFER DSECT at LBRETURN label:

```

*****
* End the program and return to CICS.
*****
LBRETURN EQU     *
                EXEC    CICS FREEMAIN DATA(MSG_BUFFER)
                EXEC    CICS RETURN
*

```

- I've modified labels for error management because of the

new length of the ERROR\_TXT variable. All MVC instructions in LBE00x labels must be changed in

```
MVC ERROR_TXT, =CL37' . . . '
```

ie for LBE001 label:

```
LBE001 EQU *  
MVC ERROR_TXT, =CL37' Assi gn. '  
B LBEMSG
```

- I've changed some constant definitions:

```
OK_MSG DC CL46' Job JOB_NAME submitted successfully. '  
*  
SECOND_MSG DC CL128' MSG: '  
*  
ERROR_MSG DC CL46' Error on '
```

---

*Gianluca Bonzano*  
*Cedacri Ovest SpA( Italy)*

© Xephon 2003

---

## CICS questions and answers

Q We're using CICS Web Services. Any recommendations for what I should specify for SOCKETCLOSE?

A SOCKETCLOSE controls the length of time CICS will support HTTP persistent sockets. With persistent sockets, the TCP/IP connection remains open, which saves CPU both in connecting and in starting/ending a task. In CICS terms, you'll see a CWXN task suspended while waiting for a new request. This task, and the underlying socket, is for one specific user and won't be shared. In addition, if you're using SSL then this persistence saves a lot of CPU-intensive handshaking.

If you specify SOCKETCLOSE=NO, then tasks will wait until the underlying TCP/IP socket is closed by the client, which for Internet Explorer seems to be about a minute. If you

specify 000000, then the CWXN task will end as soon as the alias transaction is started.

It is probably best to pick a figure slightly greater than the average user think time, something between 10 seconds and 2 minutes (?); this needs to be balanced against MAXTASKs and the arrival rate of Web requests.

*If you have any CICS-related questions, please send them in and we will do our best to find answers. Alternatively, e-mail them directly to [cicsq@xephon.net](mailto:cicsq@xephon.net).*

---

© Xephon 2003

---

# CICS news

---

Jacada has released Version 4.0 of its Jacada Integrator, featuring J2EE Connector Architecture (JCA) support and new visual data modelling capabilities.

It includes capabilities for direct integration to IBM mainframe CICS and IMS transactions and for non-intrusive, real-time integration to existing host legacy environments such as zSeries and iSeries plus Unisys, VAX, Data General, and Unix.

The addition of the Visual Data Modeler is said to eliminate the need for custom coding and allow Jacada Integrator to conform to the data standards and requirements defined by the user, instead of the user having to conform to a data standard dictated by a tool. Using Jacada Integrator's Visual Data Modeler allows users to create or import data structures into the Jacada Integrator development environment and to perform data mapping and transformation operations between the various data structures involved in the integration process, without, it's claimed, writing any code.

Eliminating the need for XSLT processing, says the vendor, reduces the overhead of processing the XML document.

In addition, there's new support for XPath, a standard for working with XML documents, allowing users to query and transform the hierarchical data structures created within the development environment.

For further information contact:  
Jacada, 400 Perimeter Center Terrace, Suite 100, Atlanta, GA 30346, USA.

Tel: (770) 352 1300.

URL: <http://www.jacada.com/news/PR173.htm>.

\* \* \*

Responsive has announced Cache Xtender (CX), which is designed to optimize the use of DASD Control Unit cache memory for datasets accessed by subsystems such as DB2, IMS/DB, IMS/DBCT, and CICS VSAM.

CX allows system administrators to specify entire performance critical objects, such as DB2 or IMS indices or a partition of an object, to be kept in DASD cache memory. CX is said to be particularly useful when dealing with very large critical indices or data that cannot fit into the host memory pools, but can fit into the DASD cache.

When using CX with DB2 and IMS/DB in a sysplex environment with data sharing, the host buffer pools and CF GBPs may be kept at a more reasonable size by keeping large critical objects fully-cached in the DASD memory.

CX is designed to integrate with any automation product available on z/OS.

For further information contact:  
Responsive Systems, 281 Highway 79,  
Morganville, NJ 07751, USA.

Tel; (732) 972 1261.

URL: <http://www.responsivesystems.com/announcements/CX1%20Press%20Release.pdf>.

\* \* \*



**xephon**