August 2002

In this issue

3  CICS Open Transaction Environment enhancements
10  Deactivating and reactivating CICS
25  Displaying transaction information
38  Interpreting CICS interval control behaviour
44  CLER – a new CICS transaction for displaying and changing some LE options
47  CICS questions and answers
48  CICS news

© Xephon plc 2002
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

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 £170 ($260) per 1000 words and £100 ($160) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £50 ($80) per 100 lines. In addition, there is a flat fee of £30 ($50) per article. 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.

© Xephon plc 2002. 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 Open Transaction Environment enhancements

INTRODUCTION
CICS Transaction Server for z/OS Version 2 Release 2 became generally available in January 2002. One of the many enhancements included in this new release of CICS was the continued evolution of the Open Transaction Environment (OTE). This article discusses the exploitation of OTE within CICS TS 2.2, and describes how OTE support has been extended in the new release.

OTE IN CICS TRANSACTION SERVER 1.3
Support for OTE was introduced in CICS Transaction Server 1.3. It provides the ability to run specific types of program environments under a separate TCB to the CICS quasi-reentrant (QR) TCB, used for subdispatching traditional CICS workloads. Such OTE-managed TCBs are referred to as open TCBs and run independently of the CICS QR TCB; that is, they are dispatched separately by the MVS dispatcher and may well run truly in parallel with the QR TCB, being dispatched on different Central Processors (CPs) at the same time.

By running programs under their own TCB, issues such as the use of MVS services that would lead to TCB suspension (‘blocking’) have been addressed. Such services are documented as being restricted for traditional CICS applications, since they would cause the QR TCB to be suspended whilst they were satisfied by the operating system. Such a suspension would mean that the CICS dispatcher service was unable to subdispatch other tasks on the QR TCB during this time, and so have a detrimental effect on a CICS transaction workload throughput. By limiting the effect of blocking to a particular application running under its own TCB, the impact does not affect other applications running on the CICS system.

When CICS Transaction Server 1.3 was made generally available, OTE supported only JVM programs that interpreted Java class files of
bytecodes. Such JVM programs were dispatched on OTE open TCBs referred to as J8 TCBs. The naming convention of each two-character open TCB name relates to the type of OTE open TCB mode. For J8 TCBs, the letter J denotes JVM programs and the number 8 denotes that the program environment will run in storage protection key 8. Subsequent to CICS Transaction Server 1.3 being released, CICS Java support was enhanced via PTF UQ44003 to support ‘hot-pooling’ for compiled Java applications processed using the Enterprise Toolkit for OS/390 Compiler and Binder. This type of Java program environment is also referred to as the High Performance Java (HPJ) compiler and run-time. Such HPJ-compiled Java applications are known as Java program objects, to differentiate them from Java class files interpreted in a JVM environment. Hot-pooling support exploits the Language Environment (LE) preinitialization services (known as PIPI) to construct a preinitialized and reusable enclave for use by Java program objects running under CICS TS 1.3. This enclave utilizes batch-LE services. Prior to hot-pooling support, execution of HPJ-compiled program objects was handled under the QR TCB. With hot-pooling, a new OTE open TCB mode was introduced. This is the H8 mode.

The JVM environment provided within CICS Transaction Server 1.3 was not reusable by subsequent transactions. As such, every instance of a JVM program would need to reload and reestablish the Java runtime environment for the JVM to execute in. This was also true for HPJ-compiled Java program objects running under the QR TCB. With Java hot-pooling, however, the LE enclave environment was reusable by subsequent Java programs, subject to certain restrictions. As such, hot-pooling offered a performance improvement for Java programs running under CICS Transaction Server 1.3. (Note: CICS Transaction Server 2.2 provides support for a reusable JVM environment, which offers a significant performance improvement over the original JVM support as provided within CICS Transaction Server 1.3. Applications using Java hot-pooling and HPJ are still supported within CICS Transaction Server 2.2, but these environments are provided for application migration purposes. The JVM environment (running under J8 open OTE TCBs) is the strategic platform for use by Java programs running within CICS Transaction Server 2.2).
The different types of open TCB mode are specific to their type of program environment and OTE therefore handles them in different ways. The type of open TCB mode assigned to a particular program (eg J8 for a JVM, H8 for a Java hot-pooled program object) is determined by CICS, and is not specified by the system programmer nor by the application program itself. Also, it is not possible to exploit OTE for other types of program environment. As such, only JVMs and Java hot-pooled programs could utilize OTE in CICS Transaction Server 1.3.

In CICS Transaction Server 1.3, control of the total number of OTE open TCBs (for all types of mode) was by means of the MAXOPENTCBS system initialization parameter in the SIT, or as a SIT override. The value could also be changed dynamically using the CICS SPI command EXEC CICS SET SYSTEM, or via CEMT or CECI. There could be many open TCBs concurrently allocated to the CICS address space, up to the limiting value of MAXOPENTCBS, subject to the proviso of there being enough virtual storage available to satisfy the TCBs. Again, CEMT, CECI, or the SPI command EXEC CICS INQ SYSTEM could be used to show the current number of OTE open TCBs allocated to tasks (ACTOPENTCBS). A given CICS task can have only one TCB of each OTE mode. The TCB is retained by that task from the time that it is allocated through to the end of the task, at which time it may be allocated to another task or destroyed. All such TCB management is controlled by CICS OTE logic; again, the application program has no control over this activity.

OTE IN CICS TRANSACTION SERVER 2.2

Support for OTE has been extended with CICS Transaction Server 2.2. In addition to providing unique TCB environments for Java programs to exploit, a new open TCB mode has been introduced. This is the L8 mode, for OPENAPI Task-Related User Exit programs. CICS Transaction Server 2.2 has made changes to the CICS DB2 Attachment facility so that this can now exploit OTE and use L8 open TCBs to process DB2 requests from CICS to DB2 6.1 systems (and above). For connections to DB2 5.1 systems (and below), the CICS Transaction Server 2.2 DB2 Adapter will utilize privately managed TCBs, as were used in earlier releases of CICS. (In either case, whether L8 or privately...
managed TCBs are used by the Adapter, their selection and control is fully automated and managed by CICS. Both techniques have the same requirement of shielding the QR TCB from the potential for blocking during calls to the DB2 subsystem.)

The advantage of utilizing open TCBs for DB2 requests, instead of the privately managed ones, may be seen when CICS DB2 application programs are defined with a CONCURRENCY of THREADSAFE instead of the default value of QUASIRENT. Programs that are quasi-reentrant rely upon the serialization provided by CICS when accessing shared resources – this serialization is achieved by the use of the single QR TCB to dispatch and execute such programs on, and by the use of CICS internal locking mechanisms when other internally-managed CICS TCBs are used to process certain specific aspects of such program’s operations. Quasi-reentrancy also means that such programs must be serially reusable between entry and exit points, so that other tasks executing the same program under the QR TCB (but at different times) are presented with the same view of the program. Conversely, threadsafety indicates that applications do not have an affinity towards the QR TCB for their dispatching. A threadsafe program is written to certain standards, and when it accesses shared resources such a program takes into account the possibility that other programs may be executing concurrently (under different TCBs) and may be attempting to modify the same resources at the same time. A threadsafe application uses appropriate techniques, such as locking or assembly language compare and swap instructions, when accessing any such shared resources. The subject of threadsafety is discussed in more detail in the section Multithreading in Chapter 11 (Application design) of the CICS Transaction Server 2.2 Application Programming Guide.

If a CICS application has been defined as being threadsafe, the CICS DB2 Adapter in CICS Transaction Server 2.2 can return control to the application under the L8 TCB used for the DB2 request, rather than having to switch back to the QR TCB as was the case in earlier releases of CICS. Such an approach means that the need for paired TCB switches (from QR TCB to CICS DB2 Adapter TCB, then back again) can be avoided for calls to DB2 from the application. In an ideal case, the application could remain running under its L8 TCB for the duration
of its work, until its eventual syncpoint. At this point, CICS would have need to use the QR TCB for some of the syncpointing work, and so TCB switches would be seen during the commitment of the Unit Of Work (UOW). In reality, switches back to QR from the L8 TCB will be dictated by what work the application has to perform upon return from DB2. Not all the EXEC CICS API or SPI functionality is itself threadsafe, and so CICS will have to perform automatic switches back to the QR TCB in order to process such requests.

Also, a holistic approach must be taken when reviewing whether a given program is threadsafe or not. Whilst executing within CICS, the flow of control can pass from an application to CICS, and then through a number of different environments such as User Replaceable Modules (URMs), Global User Exits (GLUEs) and Task-Related User Exits (TRUEs). Each of these environments has a potential requirement to be treated as quasi-reentrant. Unless they can be deemed to be threadsafe programs and so eligible for execution under an open TCB, such URMs, GLUEs, and TRUEs will force CICS to switch from an open TCB to the QR TCB before invoking them. Therefore, careful consideration of the nature of such program environments must be made before the benefits of the new CICS DB2 Attachment OTE exploitation may be seen. Analysis of the definitions and characteristics of all programs invoked during the execution of CICS DB2 applications should be performed, when reviewing the use of threadsafety on application definitions. Such programs may be home-grown, or else supplied by vendors. The use of CICS statistics will be useful in determining further information on both the program and dispatcher environments within CICS. Also, an examination of CICS trace during the execution of a threadsafe application will reveal places where the task was switched from its L8 TCB to the QR TCB. The trace is a useful way of eyeballing such switches very quickly, and so identifying paths within the execution of the task that may require further investigation when trying to improve the exploitation of open TCBs within CICS.

Below is an edited example of a trace from a CICS Transaction Server 2.2 system, where an application has issued an EXEC SQL call to DB2:

```
00055 QR  AP 2520 ERM  ENTRY COBOL-APPLICATION-CALL-TO-TRUE(DSNCSQNL) =000335=
00055 QR  US 0401 USXM  ENTRY INQUIRE_TRANSACTION_USER =000336=
```
The program is running under task number 00055. When CICS invokes the External Resource Manager (DFHERM), control is switched from the QR TCB to an OTE-managed open TCB; in this case it is number L8000. Note the TCB number is given in the second column of the trace data.

Clearly, any CICS application benefits in reduction of CPU utilization by the use of L8 open OTE TCBs for DB2 requests will be dependent on the number of calls to DB2 from within a threadsafe application, and upon the other non-DB2 work that the application has to perform, which may itself dictate a TCB switch back to the QR TCB. Some metrics from CICS Transaction Server 2.2 performance testing were taken from an example threadsafe CICS application that issued 100 DB2 calls. The same program was run on both CICS Transaction Server 2.1 and CICS Transaction Server 2.2, on a 9672 X37 processor. When run on CICS Transaction Server 2.1, the application used 11.27ms of CPU. When run on CICS Transaction Server 2.2, the same application used 8.7ms of CPU. This shows an approximate saving of 22% CPU time when this particular application was migrated to CICS Transaction Server 2.2. However, as stated above, such results are dependent upon the threadsafety of any other CICS work performed by the program and upon the number of calls to DB2. Each application would be expected to reveal differing results.

SYSTEM DEFINITIONAL CHANGES
In CICS Transaction Server 1.3, the single system definition parameter
MAXOPENTCBS was used to control the maximum number of open TCBs that could coexist within a CICS region. With the introduction of a third mode of open TCB in CICS Transaction Server 2.2 (the L8 mode), this was no longer granular enough for system administration purposes. For example, a larger number of OPENAPI L8 TCBs for DB2 use might be required than the number of J8 TCBs for JVM programs to use. Therefore, separate system definition parameters have been introduced for the different open TCB modes. The maximum number of J8 TCBs is controlled by the new MAXJVMTCBS parameter; similarly, H8 TCBs are controlled by the new MAXHPTCBS parameter. MAXOPENTCBS has been retained, but is now used to solely control the maximum number of L8 TCBs available for use by OPENAPI TRUEs (such as DB2). Each of these parameters can be set in the SIT (or via a SIT override), and can be inquired upon and varied by the CICS SPI, or via the CECI and CEMT transactions. It should be noted that their visibility is now under a new system administration component; that is, they are accessed via inquire or set commands against the DISPATCHER component of CICS rather than via the SYSTEM component as was the case in CICS Transaction Server 1.3. As well as providing these parameters for governing the maximum number of open TCBs of each OTE mode, corresponding parameters exist to return the actual current number of each mode of OTE TCB - these are ACTJVMTCBS, ACTHPTCBS and ACTOPENTCBS.

FURTHER READING AND CONTACT INFORMATION

The **CICS Application Programming Guide** and **Resource Definition Guide** give further information on threadsafety and quasi-reentrancy for programs running within CICS. The CICS DB2 Guide discusses the interface between CICS and DB2 in great detail.

I hope that this article has helped explain the enhancements to the Open Transaction Environment within CICS TS 2.2.

*Andy Wright (andy_wright@uk.ibm.com)*  
*CICS Change Team*  
*IBM (UK)*  

© IBM 2002
Deactivating and reactivating CICS

This application was designed to make the resources from a CICS region available for batch processing without a CICS stop or restart. Also it’s possible to reactivate the resources for the CICS (online) processing with this application.

It’s important to copy the CSMI transaction from the DFHISC group as the PROJ transaction with the parameter:

```plaintext
PROgram : DFHMIIRS
```

otherwise you can’t work with the requested COMMAREA!

Actions:

- Close all VSAM user datasets (starting with ‘D’) and stop the running transactions and lock the selected CICS for online processing.

Or:

- (Re) open all VSAM user datasets (starting with ‘D’) and reload all user programs (starting with ‘CI’), and make the selected CICS ready for online processing again.

JCL to deactivate CICS via batch for online processing:

```plaintext
//B001475X JOB 002665,'REIS CLAUS',NOTIFY=B001475,
//*---------------------------------------------------------------------
//** JOB SUBMITTED FROM ?????????????????????????????
//** DOC: DEACTIVATE A CICS
//** GRP: ????????
//** DATE: ??:??:??, TIME: ??:??
//*---------------------------------------------------------------------
// CLASS=T,MSGCLASS=X,REGION=4M,RESTART=*                
//---------------------------------------------------------------------
//STOP EXEC PGM=CSPROJEX,PARM='C13ST??,STOP'
//*---------------------------------------------------------------------
//STEPLIB DD DSN=MVS.NLV.LINKLIB,DISP=SHR
// DD DSN=CICSTS13.NLV.SDFHEXCI,DISP=SHR
//SYSABEND DD SYSOUT=*                                   
//SYSPRINT DD SYSOUT=*                                  
//SYSOUT DD SYSOUT=*                                    
/*
//
```
The parameter C13ST?? must be replaced with the application ID from a CICS.

Valid return codes are:
- 0 = OK
- 2222 = invalid parameter
- 3333 = link failed (eg invalid application ID).

JCL to reactivate CICS via batch for online processing:

```plaintext
//B01475X JOB 002665,'REIS CLAUS',NOTIFY=B01475,
//*****************************************************************************************/
/// JOB SUBMITTED FROM ?????????????????????????????????????????
/// DOC: DEACTIVATE A CICS
/// GRP: ???????
/// DATE: ??.,??., TIME: ??:??
///*****************************************************************************************/
/// CLASS=T,MSGCLASS=X,REGION=4M,RESTART=*  
///*****************************************************************************************/
///STOP EXEC PGM=CSPROJEX,PARM='C13ST??,STRT'
///*****************************************************************************************/
///STEPLIB DD DSN=MVS.NLV.LINKLIB,DISP=SHR
/// DD DSN=CICSTS13.NLV.SDFHEXCI,DISP=SHR
///SYSABEND DD SYSOUT=*  
///SYSPRINT DD SYSOUT=*  
///SYSOUT DD SYSOUT=*  
///*****************************************************************************************/

Again, the parameter C13ST?? must be replaced by the application ID from a CICS.

Valid return codes are:
- 0 = OK
- 2222 = invalid parameter
- 3333 = link failed (eg invalid application ID).

You can find a message from each activity from the server program and in the CICS log.

CSPROJEX
*ASM X0PTS(EXCI)
* MODULE NAME = CSPROJEX  
* PROGRAM MUST BE LINKED WITH AMODE(31) AND RMODE(24) !  
* DESCRIPTIVE NAME = Assembler batch client program  
* Author: Claus Reis Nuenberger Lebensversicherung AG  
*              February, 2002  
* Sample: Use IBM's SDFSAMP - 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 START.  
* EJECT  
* Include the standard copybook for EXCI return codes and DSECTS.  
* SPACE  
COPY DFHXCRCD  
EJECT  
* Include the standard copybook for EXCI Parameter list equates.  
* SPACE  
COPY DFHXCPLD  
EJECT  
* >>>>>>>>>>>>>>> MAINLINE CODE <<<<<<<<<<<<<<<<<<<  
* SPACE  
CSPROJEX AMODE 31  
CSPROJEX RMODE 24  
CSPROJEX DFHEIENT CODEREG=(R3,R4)  
SPACE  
* Retrieve the target CICS Applid and Userid, passed in PARM.  
* SPACE  
L R1,Ø,(R1) R1-->JCL Plist.  
LA R1,2,(R1) Chars at parm start  
MVC TARGET_SYSTEM(7),Ø(R1) MOVE TARGET CICS  
MVC PROCHAR,8(R1) MOVE STRT or STOP  
CLC PROCHAR,='STRT' IS  
BE PROCESS the  
CLC PROCHAR,='STOP' parm  
BNE INVPRCHA valid? No: Abend222  
SPACE  
PROCESS EQU *  
LA R9,EXEC_RETAREA R9--> EXEC returns  
USING EXCI_EXEC RETURN_CODE,R9  
SPACE  
* Perform LINK request.  
* SPACE  
EXEC CICS LINK PROGRAM(TARGET_PROGRAM)  
   TRANSID(TARGET_TRANSID)  
   APPLID(TARGET_SYSTEM)  
   C

© 2002. Xphon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.
COMMAREA(COMMAREA)
LENGTH(EXEC_COM_LEN)
DATALength(EXEC_DAT_LEN)
RETCODE(EXEC_RETAREA)
SYNCONRETURN

SPACE
* Did the call fail? Check the Return Codes.
* SPACE
CLC EXEC_RESP,ZERO
BNE LINKFAIL Abend with rc = 3333
SPACE
* Terminate program correctly
* SPACE
L R15,EXEC_RESP
DFHEIRET RCREG=(R15)
SPACE
* Terminate program after processing characters aren't valid
* SPACE
INVPRCHA EQU *
   ABEND 2222,DUMP
   DS D'O'
   SPACE
* Terminate program after call failed
* SPACE
LINKFAIL EQU *
   ABEND 3333,DUMP
   DS D'O'
   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' '

EXEC_INFO DS ØF 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)

ZERO DC F'O'
SPACE
EYEcom DC C'#####EYEcom#####'
COMMAREA DS ØCL4
PROCHAR DS CL4
ORG COMMAREA+4
SPACE
DFHREGS
SPACE
END CSPROJEX
CS PROJ

*ASM XOPTS(CICS,SP)
CSPROJ    TITLE '*** Program to make CICS available for the BATCH ***'

*    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 for the reverse case.
* CSPROJ must be called via the BATCH as an "EXCI" - program.
* A COMMAREA with a length of four bytes with the content STOP (BATCH - processing) or START (ONLINE - 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 - AGB will be opened.
* SOME INFORMATION - MESSAGES DURING THIS PROCESS AT SYSTEM - CONSOLE ARE ALSO AVAILABLE.

SPACE
* Author: Claus Reis / February, 2002
EJECT
* Definitions
SPACE
*  INCLUDE++ CSWA
*  INCLUDE CICS CWA ASSEMBLER STRUCTURE
*  DATA SET CSWA AT LEVEL 047 AS OF 26/10/01
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
CWAECSSA DC AL4(0) POINTER ECSA-CTL-AREA
CWAENQA DC AL4(0) POINTER ECSA ENQ-AREA
CWATSYS DC CL4' ' TERMINAL-SYSID FÜR CWACICID
CWASYSID DC CL4' ' ORIGINAL SYSTEM-ID
CWAPPLID DC CL8' ' ORIGINAL APPLICATION-ID
<table>
<thead>
<tr>
<th>Prozedur</th>
<th>DC</th>
<th>Feldbezeichner</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWANCVT</td>
<td>DC</td>
<td>AL4(0)</td>
<td>POINTER NLV-CVT</td>
</tr>
<tr>
<td>CWE_PTR_CUATR</td>
<td>DC</td>
<td>AL4(0)</td>
<td>ADRESSE D.CUA-TRANSAKTIONSTABELLE</td>
</tr>
<tr>
<td>CWA_CICSLEVEL</td>
<td>DC</td>
<td>ØCL4'</td>
<td>CICS-LEVEL '0311' OR '0330'</td>
</tr>
<tr>
<td>CWA_CICSLV</td>
<td>DC</td>
<td>CL1'</td>
<td>CICS-LEVEL</td>
</tr>
<tr>
<td>CWA_CICSVR</td>
<td>DC</td>
<td>CL1'</td>
<td>CICS-VERSION</td>
</tr>
<tr>
<td>CWA_CICSREL</td>
<td>DC</td>
<td>CL1'</td>
<td>CICS-RELEASE</td>
</tr>
<tr>
<td>CWA_CICSMOD</td>
<td>DC</td>
<td>CL1'</td>
<td>CICS-MODIFICATION</td>
</tr>
<tr>
<td>CWA_CMFSTOP</td>
<td>DC</td>
<td>PL4'Ø'</td>
<td>STOP-TIME FOR CMF EVENTS HHMMSS STC</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>XL56Ø</td>
<td>FREI</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>XLI2Ø</td>
<td>FREI</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>XLI22</td>
<td>FREI</td>
</tr>
<tr>
<td>CWACICTX</td>
<td>DC</td>
<td>CL4'</td>
<td>CICS-ID-BESCHREIBUNG</td>
</tr>
<tr>
<td>CWACICID</td>
<td>DC</td>
<td>CL1'</td>
<td>CICS-ID</td>
</tr>
<tr>
<td>CWASPROD</td>
<td>EQU</td>
<td>C'P'</td>
<td>.. PROD</td>
</tr>
<tr>
<td>CWASTEST</td>
<td>EQU</td>
<td>C'T'</td>
<td>.. TEST</td>
</tr>
<tr>
<td>CWASVPRD</td>
<td>EQU</td>
<td>C'V'</td>
<td>.. VORPROD</td>
</tr>
<tr>
<td>CWASBOST</td>
<td>EQU</td>
<td>C'S'</td>
<td>.. SYSTEM-CICS</td>
</tr>
<tr>
<td>CWACICNR</td>
<td>DC</td>
<td>CL1'</td>
<td>CICS-NR</td>
</tr>
<tr>
<td>CWASTERM</td>
<td>EQU</td>
<td>C'T'</td>
<td>.. TERMINAL</td>
</tr>
<tr>
<td>CWASVSAM</td>
<td>EQU</td>
<td>C'V'</td>
<td>.. DATASET VSAM</td>
</tr>
<tr>
<td>CWASPAIS</td>
<td>EQU</td>
<td>C'P'</td>
<td>.. PAISY</td>
</tr>
<tr>
<td>CWASODM</td>
<td>EQU</td>
<td>C'O'</td>
<td>.. ODM</td>
</tr>
<tr>
<td>CWASPROB</td>
<td>EQU</td>
<td>C'9'</td>
<td>.. APPLICATION 9 / PROBLEM CICS</td>
</tr>
<tr>
<td>CWASAPPL</td>
<td>EQU</td>
<td>C'Ø'</td>
<td>.. APPLICATION ØØ-Ø9</td>
</tr>
<tr>
<td>* $APPL</td>
<td>EQU</td>
<td>???</td>
<td>.. APPLICATION A-C</td>
</tr>
<tr>
<td>* $APPL</td>
<td>EQU</td>
<td>???</td>
<td>.. APPLICATION E-O</td>
</tr>
<tr>
<td>* $APPL</td>
<td>EQU</td>
<td>???</td>
<td>.. APPLICATION Q-S</td>
</tr>
<tr>
<td>* $APPL</td>
<td>EQU</td>
<td>???</td>
<td>.. APPLICATION U-Z</td>
</tr>
<tr>
<td>CWADATUM</td>
<td>DC</td>
<td>CL8'</td>
<td>DATUM FORMAT TT.MM.JJ</td>
</tr>
<tr>
<td>CWACTMJ</td>
<td>DC</td>
<td>CL6'</td>
<td>DATUM TTMMJ</td>
</tr>
<tr>
<td>CWAPTMJ</td>
<td>DC</td>
<td>PL4'Ø'</td>
<td>DATUM ØTTMMJJC</td>
</tr>
<tr>
<td>CWACJMT</td>
<td>DC</td>
<td>CL6'</td>
<td>DATUM JJMMM</td>
</tr>
<tr>
<td>CWAPJMT</td>
<td>DC</td>
<td>PL4'Ø'</td>
<td>DATUM ØJJMMTTC</td>
</tr>
<tr>
<td>CWACTM4J</td>
<td>DC</td>
<td>CL8'</td>
<td>DATUM TTMMJ4</td>
</tr>
<tr>
<td>CWAPTM4J</td>
<td>DC</td>
<td>PL5'Ø'</td>
<td>DATUM ØTMMJ4J4</td>
</tr>
<tr>
<td>CWACJ4MT</td>
<td>DC</td>
<td>CL8'</td>
<td>DATUM JJJMMT</td>
</tr>
<tr>
<td>CWAPJ4MT</td>
<td>DC</td>
<td>PL5'Ø'</td>
<td>DATUM ØJJJMMT4</td>
</tr>
<tr>
<td>CWACMJ</td>
<td>DC</td>
<td>CL4'</td>
<td>DATUM MMJJ</td>
</tr>
<tr>
<td>CWAPMJ</td>
<td>DC</td>
<td>PL3'Ø'</td>
<td>DATUM ØMMJJC</td>
</tr>
<tr>
<td>CWACJM</td>
<td>DC</td>
<td>CL4'</td>
<td>DATUM JJJM</td>
</tr>
<tr>
<td>CWAPJM</td>
<td>DC</td>
<td>PL3'Ø'</td>
<td>DATUM ØJJMMC</td>
</tr>
<tr>
<td>CWACJM4J</td>
<td>DC</td>
<td>CL6'</td>
<td>DATUM MMJ4J</td>
</tr>
<tr>
<td>CWAPMJ4J</td>
<td>DC</td>
<td>PL4'Ø'</td>
<td>DATUM ØMMJ4J4</td>
</tr>
<tr>
<td>CWACJ4MJ</td>
<td>DC</td>
<td>CL6'</td>
<td>DATUM JJJMM</td>
</tr>
<tr>
<td>CWAPJ4MJ</td>
<td>DC</td>
<td>PL4'Ø'</td>
<td>DATUM ØJJJMMC</td>
</tr>
<tr>
<td>CWACT3J</td>
<td>DC</td>
<td>CL5'</td>
<td>DATUM TTTJ</td>
</tr>
<tr>
<td>CWAP3T3J</td>
<td>DC</td>
<td>PL3'Ø'</td>
<td>DATUM TTTJ3</td>
</tr>
<tr>
<td>CWACJ3T</td>
<td>DC</td>
<td>CL5'</td>
<td>DATUM JJTJ</td>
</tr>
<tr>
<td>CWAP3J3T</td>
<td>DC</td>
<td>PL3'Ø'</td>
<td>DATUM JTTTC</td>
</tr>
<tr>
<td>CWACT3J4</td>
<td>DC</td>
<td>CL7'</td>
<td>DATUM TTTJ3J</td>
</tr>
<tr>
<td>CWAP3J4T</td>
<td>DC</td>
<td>PL4'Ø'</td>
<td>DATUM TTTJ3J4</td>
</tr>
</tbody>
</table>
CWACJ4T3 DC CL7' ' DATUM JJJJTTC
CWAPJ4T3 DC PL4'Ø' DATUM JJJJTTTC
CWAZEIT DC CL5' ' UHRZEIT SS:MM
CWATABLE DS 0CL24 ,+0123456789,-0123456789
CWATAB1 DS 0CL13 TABELLE 1 /,+/0123456789/-/
CWATAB2 DS 0CL12 TABELLE 2 /,+/0123456789/
CWACHK01 DC C','
CWATAB3 DS 0CL12 TABELLE 3 /,+/0123456789/-/
CWATAB4 DS 0CL11 TABELLE 4 /,+/0123456789/
CWACHARP DC C'+'
CWATAB5 DS 0CL12 TABELLE 5 /0123456789/-/,
CWATAB6 DS 0CL11 TABELLE 6 /0123456789/-/,
CWATAB7 DS 0CL10 TABELLE 7 /0123456789/
CWACH09 DC C'0123456789'
CWACHARM DC C'-'
CWATAB8 DS 0CL11 TABELLE 8 /,0123456789/
CWACHK02 DC C','
CWACH092 DC C'0123456789'
CWAZEITP DC PL4'Ø' UHRZEIT HHMMSTTC
CWADAY DC CL10' ' WOCHENTAG
CWAMONTH DC CL9' ' MONAT
CWA_PTR_FTT DC AL4(Ø) ADRESSE D. FUNKTIONSTASTENTABELLE
CWA_PTR_ANT DC AL4(Ø) ADRESSE DER AKTIONSMONTABLLE
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_REGION *
* BEGIN THE DSECT FOR FUNKTIONSTASTENTABELLE *
* ADDRESSING OVER "CWA_PTR_FTT" *
CWAFTDSECT DSECT
CWA_FTT_TASTE DC XL1'Ø' TASTENIDENTIFIKATION
CWA_FTT_AKTION DC CL16' ' KURZBEZEICHNUNG DER TASTE *
* BESP. : HILFE
CWA_FTT_ANZEIGE DC CL20' ' TEXT FUR DEN FUNKTIONS-
* TASTENBLOCK EINES BILDES *
* BESP. : F1=HILFE
CWA_FTT_PFKEY DC CL4' ' PF-TASTE Z.B. "PF1"
CWA_FTT_KURZTEXT DC CL8' ' TASTENKUERZEL FUR POP-UP-MENUS *
* BESP. : F12=ABBR
CWA_FTT_TEXT DS CL207 BESCHREIBUNG DER AKTION
CWAFTDSECTE EQU *
CWAFTTANZAHL EQU 30 ANZAHL TABELLENINTRAEGE FTT
SPACE 2
* END OF THE DSECT FOR FUNKTIONSTASTENTABELLE *
* BEGIN THE DSECT FOR AKTIONSMONTABLLE *
* ADDRESSING OVER "CWA_PTR_ANT" *
CWAANTDSECT DSECT
CWA_ANT_HILFE   DS CL16   HELP TEXT
CWA_ANT_TASTEN  DS CL16   SHOW THE TASTENBELEGUNG
CWA_ANT_AUSGANG DS CL16   COMPLETE A FUNCTION
CWA_ANT_REFRESH DS CL16   RESTORE
CWA_ANT_UPDATE  DS CL16   DATE STORE
CWA_ANT_RUECKWAERTS DS CL16   BACKWARDS BROWSE
CWA_ANT_VORWAERTS DS CL16   FORWARDS BROWSE
CWA_ANT_AKTION  DS CL16   ACTIVATE ACTION BAR
CWA_ANT_UNTERBRECHEN DS CL16   EVENTS UNDER VIEW
CWA_ANT_ABBRUCH DS CL16   ABORT
CWA_ANT_EINSTIEG DS CL16   BACK TO EINSTIEGSBILD
CWA_ANT_AUSWAHL DS CL16   BACK TO AUSWAHLBILD
CWA_ANT_SICHERN DS CL16   FREEZE THE DTA
CWA_ANT_LINKS   DS CL16   LEFT-SIDE PAGES
CWA_ANT_RECHTS  DS CL16   RIGHT-SIDE PAGE
CWA_ANT_ANFANG  DS CL16   SHOW THE FIRST SIDE
CWA_ANT_SCHLUSS DS CL16   SHOW THE OTHER SIDE
CWA_ANT_ABMELENDE DS CL16   ZSS-ABMELENDE
CWA_ANT_DRUCKEN DS CL16   PRINT (PA2)
CWA_ANT_LOESCHEN DS CL16   OUTPUT TO SCREEN
CWA_ANT_DATENFREIGABE DS CL16   DATENFREIGABE
CWA_ANT_HILFE_ANLEGEN DS CL16   BOSHELP HELP START
CWA_ANT_SUCHEN  DS CL16   SEARCH
CWA_ANT_EURODM  DS CL16   CONVERT EURO/DM
CWAANTDSECTE    EQU *
*       END THE DSECT FOR AKTIONSNAMEN TABELLE     *
*       END INCLUDE++                                *
SPACE
DFHEISTG 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
TRAN   DS CL4  TRANSACTION NAME
TASKNO DS F   TRANSACTIONS TO PURGE
ABCODE DS CL4  ABEND CODE
INAREA DS CL4  COMMAREA INPUT
OUTAREA DS CL50 TERMINAL-OUTPUT-AREA
EJECT  
*   Main - program                                     *
SPACE
CSPROJ DFHEIENT CODEREG=(R3,R4),DATAREG=R12
CSPROJ AMODE ANY
CSPROJ RMODE ANY
EXEC  CICS ADDRESS CWA (CWAPTR) C
       RESP(RESP)
       CLC RESP,DFHRESP(NORMAL)
       BE  CSPR0500
MVC  ABCODE,=C'CWAA'   SYSID can't be assigned
BAS R6,ABEND
B ERRORWA

SPACE

CSPR0500 EQU *
CLI CWACICID,CWA$BOST Is it SYSTEM-CICS?
BE CSPR0700 YES: Check whether it's a
TOR/ FOR or an AOR

* CLI CWACICID,CWA$TEST Is it SYSTEM-CICS?
BE CSPR0700 YES: Check whether it's a
TOR/ FOR or an AOR

SPACE

CSPR0550 EQU *
MVC OUTAREA,MSG018 No access to the COMMAREA
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"?"
B LH CSPR0550 No: Invalid CICS

SPACE

CSPR0800 EQU *
EXEC CICS ADDRESS COMMAREA(R2) C
RESP(Resp)
CLC RESP,DFHRESP(NORMAL)
BE CSPR1000
MVC OUTAREA,MSG000 No access to the COMMAREA
BAS R7,SENDMSG
B ERRORWA

SPACE

CSPR1000 EQU *
CLC EIBCALEN,=H'4'
BE CSPR2000
MVC OUTAREA,MSG001 Invalid COMMAREA detected
BAS R7,SENDMSG
B ERRORWA

SPACE

CSPR2000 EQU *
MVC INAREA,0(R2) Move COMMAREA - content
CLC INAREA,=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 INAREA,=C'STRT'
BNE CSPR9000
BAS R6,OPENVSA 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,MSG002 Wrong processing characters
MVC OUTAREA+36(L'INAREA),INAREA
BAS R7,SENDMSG
B ERRORWA
SPACE
RETURN EQU *
EXEC CICS RETURN
EJECT
* Sub routines *
SPACE
* VTAM - ACB will be closed *
SPACE
CLVTAACB EQU *
MVC OUTAREA,MSG003
BAS R7,SENDMSG
*
EXEC CICS SET VTAM CLOSED C
RESP(RESP)
*
CLC RESP,DFHRESP(NORMAL)
BNE ERRORWA
MVC OUTAREA,MSG004
BAS R7,SENDMSG
BR R6
SPACE
* Purge all active transactions *
SPACE
PURGTRAN EQU *
MVC OUTAREA,MSG005
BAS R7,SENDMSG
*
EXEC CICS INQUIRE TASK LIST C
LISTSIZE(NBR) C
SET(R9) C
RESP(RESP) C
*
CLC RESP,DFHRESP(NORMAL) Was there a problem?
BNE ERRORWA Yes. Take the error exit.
L R5,NBR No. Get the number of tasks.
LTR R5,R5 Are there any tasks?
BZ PUR230 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
* EXEC CICS INQUIRE C
   TASK(TASKNO) C
   TRANSACTION(TAN) C
   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(2),=C'TM' Yes: Don't purge
BE PUR220 Is it the TMON/CICS?

* EXEC CICS SET TASK(TASKNO) C
   FORCEPURGE C
   RESP(RESP)
* CLC RESP,DFHRESP(NORMAL) Go and force purge the task.
BE PUR220 ..say that force purge failed.
MVC OUTAREA,MSG019 carry on,
MVC OUTAREA+15(L'TASKNO),TASKNO Move Tasknumber
MVC OUTAREA+26(L'TRAN),TRAN Move Transactionname
BAS R7,SENDMSG

PUR220 EQU *
AH R9,=H'4' Move on to the next task
BCT R5,PUR210 Loop back if there is more to do

PUR230 EQU *
MVC OUTAREA,MSG006 SPACE
BAS R7,SENDMSG SPACE
BR R6 SPACE

* User datasets will be closed *

CLOSEVS A EQU *
MVC OUTAREA,MSG07 SPACE
BAS R7,SENDMSG SPACE
EXEC CICS INQUIRE FILE START C
   RESP(RESP)
CLC RESP,DFHRESP(NORMAL)
BE CLO10000
MVC  ABCODE,=C'CINS'
BAS  R6,ABEND
B   ERRORWA
SPACE

CL01000  EQU  *
EXEC  CICS INQUIRE FILE(FILE) NEXT
      RESP(RESP)
      CLC  RESP,DFHRESP(END)  Is it "End-Condition" ?
      BE  CL08000  Yes: Stop processing
      CLC  RESP,DFHRESP(NORMAL)  Is it "Normal-Condition" ?
      BE  CL02000  No: Abend taks
      MVC  ABCODE,=C'CNEX'
      BAS  R6,ABEND
      B   ERRORWA
SPACE

CL02000  EQU  *
CLI  FILE,C'D'  Is it a User-Dataset?
      BNE  CL01000  NO: No action
      CLC  FILE(3),=C'DAI'  Is it a PAISY-Dataset?
      BE  CL01000  YES: No action
      CLC  FILE(3),=C'DFH'  Is it a CICS-Dataset?
      BE  CL01000  YES: No action
EXEC  CICS SET FILE(FILE) CLOSED
      RESP(RESP)
      CLC  RESP,DFHRESP(NORMAL)  Is it "Normal-Condition" ?
      BE  CL01000  Yes: Next file
      MVC  OUTAREA,MSG016  Move message
      MVC  OUTAREA+26(L'FILE),FILE  Move DD-Name
      BAS  R7,SENDMSG
      B   CL01000
SPACE

CL08000  EQU  *
EXEC  CICS INQUIRE FILE END
      RESP(RESP)
      CLC  RESP,DFHRESP(NORMAL)
      BE  CL09000
      MVC  ABCODE,=C'CINE'
      BAS  R6,ABEND
      B   ERRORWA
SPACE

CL09000  EQU  *
MVC  OUTAREA,MSG008
      BAS  R7,SENDMSG
      BR  R6
SPACE

*  User - datasets will be opened  *
SPACE
OPENVSAM  EQU  *
MVC  OUTAREA,MSG009
      BAS  R7,SENDMSG
      EXEC  CICS INQUIRE FILE START

RESPP(RESP)
CLC  RESP,DFHRESP(NORMAL)
BE  OPE1000
MVC  ABCODE,=C'0INS'
BAS  R6,ABEND
B  ERRORWA
SPACE

OPE1000  EQU *
EXEC  CICS INQUIRE FILE(FILE) NEXT
   C  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
   C  RESP(RESP)
   CLC  RESP,DFHRESP(NORMAL)  Is it "Normal-Condition"?
   BE  OPE1000  Yes: Next file
   MVC  OUTAREA,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
   C  RESP(RESP)
   CLC  RESP,DFHRESP(NORMAL)
   BE  OPE9000
   MVC  ABCODE,=C'OINE'
   BAS  R6,ABEND
   B  ERRORWA
   SPACE

OPE9000  EQU *
MVC  OUTAREA,MSG010
BAS  R7,SENDMSG
BR  R6
SPACE

*  User - programs will be reloaded  *
SPACE
RELOADPG  EQU *
MVC OUTAREA,MSG011
BAS R7,SENDMSG
EXEC CICS INQUIRE PROGRAM START AT('CI')
   C
   RESP(RESP)
   CLC RESP,DFHRESP(NORMAL)
   BE RELOA000
   MVC ABCODE,='STAR'
   BAS R6,ABEND
   B ERRORWA

SPACE
RELOA000 EQU *
   EXEC CICS INQUIRE PROGRAM(PROGRAM) NEXT
   C
   RESP(RESP)
   CLC RESP,DFHRESP(NORMAL)
   BE RELOA500
   CLC RESP,DFHRESP(END)
   BE RELOA600
   MVC ABCODE,='NEXT'
   BAS R6,ABEND
   B ERRORWA

SPACE
RELOA500 EQU *
   CLC PROGRAM(2),='CI'
   BNE RELOA600
   EXEC CICS SET PROGRAM(PROGRAM) PHASEIN
   C
   RESP(RESP)
   CLC RESP,DFHRESP(NORMAL)
   BE RELOA000
   MVC OUTAREA,MSG017 Move message
   MVC OUTAREA+19(L'PROGRAM),PROGRAM Move program - name
   BAS R7,SENDMSG
   B RELOA000

SPACE
RELOA600 EQU *
   EXEC CICS INQUIRE PROGRAM END
   C
   RESP(RESP)
   CLC RESP,DFHRESP(NORMAL)
   BE RELOA900
   MVC ABCODE,='ENDE'
   BAS R6,ABEND
   B ERRORWA

SPACE
RELOA900 EQU *
   MVC OUTAREA,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,MSG013

BAS R7,SENDMSG
EXEC CICS SET VTAM OPEN C
  RESP(FAIL)
  CLC
  RESP,DFHRESP(NORMAL)
  BNE ERRORWA
  MVC OUTAREA,MSG014
  BAS R7,SENDMSG
  BR R6
  SPACE
  * Send message to console *
  SPACE
SEMDMSG EQU *
  MVC LENGTH,=H'50'
  EXEC CICS WRITE OPERATOR TEXT(OUTAREA) C
  RESP(FAIL)
  CLC
  RESP,DFHRESP(NORMAL)
  BNE ERRORWA
  BR R7
  SPACE
  * Error with abend 0c1 (operation exception) *
  SPACE
ERRORWA EQU *
  Error with abend 0c1
  DC D'0'
  Never come back statement
  BR R6
  SPACE
  * Abend with abend code 'ABCODE' *
  SPACE
  ABEND DS 0H
  *
  EXEC CICS ABEND ABCODE(ABCODE)
  *
  BR R6
  SPACE
  * Delay processing for one second *
  SPACE
DELAY DS 0H
  *
  EXEC CICS DELAY INTERVAL(1)
  *
  BR R6
  EJECT
  *
  Register equates and constants *
  SPACE
  EQUREG
  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'
Displaying transaction information

The following program was created to allow for a quick overview of the transactions defined in a CICS region and their most common characteristics. The display produced is illustrated below:

```
+-------------------------------+-----------------------+------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+
| Tran Program Profile Tclass Sta Pur Prt Twas Rname Rsy dloc Dkey |                   |
|                  DFH   DFHCICS        |                   |
| CBRC DFHBRCP DFHCICST DFHTCL00 Ena 001 0000 Below Cics |
| CECI DFHECIP DFHCICST DFHTCL00 Ena Pur 001 0000 Below User |
| CEHP DFHCHS DFHCICST DFHTCL00 Ena 001 0000 Below Cics |
| CEHS DFHCHS DFHCICST DFHTCL00 Ena 001 0000 Below Cics |
| CEMT DFHEMTP DFHCICST DFHTCL00 Ena 255 0000 Below Cics |
| CETO DFHEOTP DFHCICST DFHTCL00 Ena Pur 255 0000 Below Cics |
| CESF DFHSFP DFHCICST DFHTCL00 Ena Pur 001 0000 Below Cics |
| CESN DFHNSP DFHCICST DFHTCL00 Ena Pur 001 0000 Below Cics |
| CEST DFHESTP DFHCICST DFHTCL00 Ena Pur 255 0000 Below Cics |
```
On the third line of the screen, under the field names, you can input the initial characters of the displayed fields. In the example given, the display is limited to transactions associated with programs whose names start with DFH, who belong to DFHCICS profiles, and whose datalocation is BELOW.

If you do not enter any restriction, all the transactions in the system will be browsed. The columns displayed are: Transaction name, Program associated, Profile, Transaction class, Status (Enabled or Disabled), Purge status, Priority, Twasize, Remotename, Remotesystem, Datalocation (Below or Any) and Datakey (Cics or User).

This application consists of a COBOL program (Vitermp) and a BMS map (Viterms). I chose to call the associated transaction VTRA, but you can choose any name you like. All you need to do is to adjust the transaction name in the last level 77 variable of the program.

VITRANP COBOL SOURCE CODE

IDENTIFICATION DIVISION.
PROGRAM-ID. VITRANP.
*
ENVIRONMENT DIVISION.
DATA DIVISION.
******************************************************************************
WORKING-STORAGE SECTION.
******************************************************************************

77 X PIC S9(4) COMP VALUE +0.
77 Z PIC S9(4)  COMP VALUE +0.
77 SLENL  PIC S9(4) COMP VALUE +0.
77 W-RESP PIC S9(8)  COMP VALUE +0.
77 W-RESP2 PIC S9(8)  COMP VALUE +0.
77 COUNT-I PIC S9(4)  COMP VALUE +0.
77 ABTIME PIC S9(15) COMP-3.
77 NUM-AUX   PIC 9(8) VALUE 0.
77 END-MESSAGE PIC X(3) VALUE 'END'.
77 TRANS-NAMES PIC X(4) VALUE 'VTRA'.

*Ø1 SEARCH-FIELDS.
  Ø2 SEARCH-RESULT PIC X(10).
  Ø2 SEARCH1 PIC X(10).
  Ø2 SEARCH1 REDEFINES SEARCH1 PIC X OCCURS 10.
  Ø2 SEARCH2 PIC X(10).

*Ø1 COMMAREA.

*Ø2 COUNT-LAST PIC S9(4).
Ø2 W-LINE.
  Ø5 W-TRAN PIC X(4).
  Ø5 W-PROG PIC X(8).
  Ø5 W-PROF PIC X(8).
  Ø5 W-TCLA PIC X(8).
  Ø5 W-STAT COMP PIC S9(8).
  Ø5 W-PURG COMP PIC S9(8).
  Ø5 W-PTY COMP PIC S9(8).
  Ø5 W-TWAS COMP PIC S9(8).
  Ø5 W-RNAM PIC X(8).
  Ø5 W-RSYS PIC X(4).
  Ø5 W-DLOC COMP PIC S9(8).
  Ø5 W-DKEY COMP PIC S9(8).
Ø2 LINE-ACT.
  Ø5 TRAN-ACT PIC X(4).
  Ø5 PROG-ACT PIC X(8).
  Ø5 PROF-ACT PIC X(8).
  Ø5 TCLA-ACT PIC X(8).
  Ø5 STAT-ACT PIC X(4).
  Ø5 RNAM-ACT PIC X(8).
  Ø5 SYS-ACT PIC X(4).
  Ø5 DLOC-ACT PIC X(5).
  Ø5 DKEY-ACT PIC X(4).
Ø2 LINE-PRV.
  Ø5 FILLER PIC X(53).

Ø2 VITRANSI.
  Ø4 FILLER PIC X(12).
  Ø4 CICSNL COMP PIC S9(4).
  Ø4 CICSNF PIC X.
  Ø4 CICSN PIC X(8).
  Ø4 DDATEL COMP PIC S9(4).
  Ø4 DDATEF PIC X.
  Ø4 DDATEI PIC X(10).
  Ø4 DTIMEL COMP PIC S9(4).
  Ø4 DTIMEF PIC X.
  Ø4 DTIMEI PIC X(8).
  Ø4 TRANX-L COMP PIC S9(4).
<table>
<thead>
<tr>
<th>Register</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
<td>TRANX-F</td>
<td>PIC X</td>
</tr>
<tr>
<td>04</td>
<td>TRANX-I</td>
<td>PIC X(4)</td>
</tr>
<tr>
<td>04</td>
<td>PROGX-L</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>04</td>
<td>PROGX-F</td>
<td>PIC X</td>
</tr>
<tr>
<td>04</td>
<td>PROGX-I</td>
<td>PIC X(8)</td>
</tr>
<tr>
<td>04</td>
<td>PROFX-L</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>04</td>
<td>PROFX-F</td>
<td>PIC X</td>
</tr>
<tr>
<td>04</td>
<td>PROFX-I</td>
<td>PIC X(8)</td>
</tr>
<tr>
<td>04</td>
<td>TCLAX-L</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>04</td>
<td>TCLAX-F</td>
<td>PIC X</td>
</tr>
<tr>
<td>04</td>
<td>TCLAX-I</td>
<td>PIC X(8)</td>
</tr>
<tr>
<td>04</td>
<td>STATX-L</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>04</td>
<td>STATX-F</td>
<td>PIC X</td>
</tr>
<tr>
<td>04</td>
<td>STATX-I</td>
<td>PIC X(3)</td>
</tr>
<tr>
<td>04</td>
<td>RNAMX-L</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>04</td>
<td>RNAMX-F</td>
<td>PIC X</td>
</tr>
<tr>
<td>04</td>
<td>RNAMX-I</td>
<td>PIC X(8)</td>
</tr>
<tr>
<td>04</td>
<td>RSYSX-L</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>04</td>
<td>RSYSX-F</td>
<td>PIC X</td>
</tr>
<tr>
<td>04</td>
<td>RSYSX-I</td>
<td>PIC X(4)</td>
</tr>
<tr>
<td>04</td>
<td>DLOCX-L</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>04</td>
<td>DLOCX-F</td>
<td>PIC X</td>
</tr>
<tr>
<td>04</td>
<td>DLOCX-I</td>
<td>PIC X(5)</td>
</tr>
<tr>
<td>04</td>
<td>DKEYX-L</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>04</td>
<td>DKEYX-F</td>
<td>PIC X</td>
</tr>
<tr>
<td>04</td>
<td>DKEYX-I</td>
<td>PIC X(4)</td>
</tr>
<tr>
<td>04</td>
<td>SCREEN-LINES</td>
<td>PIC X(1764)</td>
</tr>
</tbody>
</table>

**04 LINEI REDEFINES SCREEN-LINES OCCURS 18.**

<table>
<thead>
<tr>
<th>Register</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>06</td>
<td>TRANL</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>06</td>
<td>TRANF</td>
<td>PIC X</td>
</tr>
<tr>
<td>06</td>
<td>TRANI</td>
<td>PIC X(4)</td>
</tr>
<tr>
<td>06</td>
<td>PROGL</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>06</td>
<td>PROGF</td>
<td>PIC X</td>
</tr>
<tr>
<td>06</td>
<td>PROGI</td>
<td>PIC X(8)</td>
</tr>
<tr>
<td>06</td>
<td>PROFL</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>06</td>
<td>PROFF</td>
<td>PIC X</td>
</tr>
<tr>
<td>06</td>
<td>PROFI</td>
<td>PIC X(8)</td>
</tr>
<tr>
<td>06</td>
<td>TCLAL</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>06</td>
<td>TCLAF</td>
<td>PIC X</td>
</tr>
<tr>
<td>06</td>
<td>TCLAI</td>
<td>PIC X(8)</td>
</tr>
<tr>
<td>06</td>
<td>STATL</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>06</td>
<td>STATF</td>
<td>PIC X</td>
</tr>
<tr>
<td>06</td>
<td>STATI</td>
<td>PIC X(3)</td>
</tr>
<tr>
<td>06</td>
<td>PURGL</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>06</td>
<td>PURGF</td>
<td>PIC X</td>
</tr>
<tr>
<td>06</td>
<td>PURGI</td>
<td>PIC X(3)</td>
</tr>
<tr>
<td>06</td>
<td>PRTYL</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>06</td>
<td>PRTYF</td>
<td>PIC X</td>
</tr>
<tr>
<td>06</td>
<td>PRTYI</td>
<td>PIC X(3)</td>
</tr>
<tr>
<td>06</td>
<td>TWASL</td>
<td>COMP PIC S9(4)</td>
</tr>
<tr>
<td>06</td>
<td>TWASF</td>
<td>PIC X</td>
</tr>
</tbody>
</table>
06 TWASI PIC X(4).
06 RNAML COMP PIC S9(4).
06 RNAMF PIC X.
06 RNAMI PIC X(8).
06 RSYSL COMP PIC S9(4).
06 RSYSF PIC X.
06 RSYSI PIC X(4).
06 DLOCL COMP PIC S9(4).
06 DLOCF PIC X.
06 DLOCi PIC X(5).
06 DKEYl COMP PIC S9(4).
06 DKEYf PIC X.
06 DKEYI PIC X(4).

02 VITRANSO REDEFINES VITRANSI PIC X(1890).

* **************************************************
** LINKAGE SECTION.                         ******************
* **************************************************
01 DFHCOMMAREA.
02 FILLER PIC X(4000).

* **************************************************
** PROCEDURE DIVISION.                    ******************
* **************************************************
* FIRST-TIME-ONLY.                     ******************
IF EIBCALEN = 0
   MOVE LOW-VALUES TO COMMAREA
   MOVE 2064 TO EIBCALEN
   PERFORM INITIATE-SCREEN
   PERFORM INPUT-MODIFIED
   PERFORM INQUIRE-CICS
   PERFORM SEND-SCREEN-ERASE
   GO TO RETURN-TRANSID
END-IF.
*
OTHER-TIMES.
****************
   MOVE DFHCOMMAREA TO COMMAREA
   PERFORM RECEIVE-SCREEN
   PERFORM INPUT-MODIFIED
   PERFORM INQUIRE-CICS
   PERFORM SEND-SCREEN
   GO TO RETURN-TRANSID.
****************
* Subroutines                     ******************
* **************************************************
* INPUT-MODIFIED.                   ******************
MOVE  TRANX-I  TO  TRAN-ACT
MOVE  PROGX-I  TO  PROG-ACT
MOVE  PROFX-I  TO  PROF-ACT
MOVE  TCLAX-I  TO  TCLA-ACT
MOVE  STATX-I  TO  STAT-ACT
MOVE  RNAMX-I  TO  RNAM-ACT
MOVE  RSYSX-I  TO  RSYS-ACT
MOVE  DLOCX-I  TO  DLOC-ACT
MOVE  DKEYX-I  TO  DKEY-ACT
IF  LINE-PRV  NOT  EQUAL  LINE-ACT
   MOVE  LOW-VALUES  TO  W-TRAN
   MOVE  Ø  TO  COUNT-LAST
END-IF
MOVE  LINE-ACT  TO  LINE-PRV.
*
INQUIRE-CICS.
*********************
MOVE  Ø  TO  X.
EXEC  CICS  INQUIRE  TRANSACTION  START
END-EXEC
PERFORM  INQUIRE-CICS-LOOP  THRU
   INQUIRE-CICS-LOOP-EXIT  UNTIL  X  >  18.
MOVE  COUNT-I  TO  COUNT-LAST
EXEC  CICS  INQUIRE  TRANSACTION  END
END-EXEC.
*
INQUIRE-CICS-LOOP.
*********************
EXEC  CICS  INQUIRE  TRANSACTION  (W-TRAN)
   PROGRAM  (W-PROG)
   PROFILE  (W-PROF)
   TRANCLASS  (W-TCLA)
   STATUS  (W-STAT)
   PURGEABILITY  (W-PURG)
   PRIORITY  (W-PRTY)
   TWASIZE  (W-TWAS)
   REMOTENAME  (W-RNAM)
   REMOTESYSTEM  (W-RSYS)
   TASKDATALOC  (W-DLOC)
   TASKDATAKEY  (W-DKEY)
   RESP  (W-RESP)
   RESP2  (W-RESP2)
   NEXT
END-EXEC

IF  W-RESP2  >  Ø
   ADD  1  TO  X
   PERFORM  CLEAN-SCREEN-LINES
   MOVE  99  TO  X
   MOVE  Ø  TO  COUNT-I
   GO  TO  INQUIRE-CICS-LOOP-EXIT
END-IF

ADD 1 TO COUNT-I
IF COUNT-LAST GREATER COUNT-I
   GO TO INQUIRE-CICS-LOOP-EXIT
END-IF

IF NOT (TRANX-I = SPACES OR = LOW-VALUES)
   MOVE 4 TO SLENG
   MOVE TRAXN-I TO SEARCH1
   MOVE W-TRAN TO SEARCH2
   PERFORM COMPARE-STRINGS
   IF SEARCH-RESULT = Ø
      GO TO INQUIRE-CICS-LOOP-EXIT
   END-IF
END-IF

IF NOT (PROGX-I = SPACES OR = LOW-VALUES)
   MOVE 8 TO SLENG
   MOVE PROGX-I TO SEARCH1
   MOVE W-PROG TO SEARCH2
   PERFORM COMPARE-STRINGS
   IF SEARCH-RESULT = Ø
      GO TO INQUIRE-CICS-LOOP-EXIT
   END-IF
END-IF

IF NOT (PROFX-I = SPACES OR = LOW-VALUES)
   MOVE 8 TO SLENG
   MOVE PROFX-I TO SEARCH1
   MOVE W-PROF TO SEARCH2
   PERFORM COMPARE-STRINGS
   IF SEARCH-RESULT = Ø
      GO TO INQUIRE-CICS-LOOP-EXIT
   END-IF
END-IF

IF NOT (TCLAX-I = SPACES OR = LOW-VALUES)
   MOVE 8 TO SLENG
   MOVE TCLAX-I TO SEARCH1
   MOVE W-TCLA TO SEARCH2
   PERFORM COMPARE-STRINGS
   IF SEARCH-RESULT = Ø
      GO TO INQUIRE-CICS-LOOP-EXIT
   END-IF
END-IF

IF NOT (STATX-I = SPACES OR = LOW-VALUES)
   MOVE 3 TO SLENG
   MOVE STATX-I TO SEARCH1
   PERFORM TRANSLATE-STATUS
PERFORM COMPARE-STRINGS
IF SEARCH-RESULT = Ø
  GO TO INQUIRE-CICS-LOOP-EXIT
END-IF
END-IF

IF NOT (RNAMX-I = SPACES OR = LOW-VALUES)
  MOVE 8 TO SLEN
  MOVE RNAMX-I TO SEARCH1
  MOVE W-RNAME TO SEARCH2
  PERFORM COMPARE-STRINGS
  IF SEARCH-RESULT = Ø
    GO TO INQUIRE-CICS-LOOP-EXIT
  END-IF
END-IF

IF NOT (RSYSX-I = SPACES OR = LOW-VALUES)
  MOVE 4 TO SLEN
  MOVE RSYSX-I TO SEARCH1
  MOVE W-RSYSTO SEARCH2
  PERFORM COMPARE-STRINGS
  IF SEARCH-RESULT = Ø
    GO TO INQUIRE-CICS-LOOP-EXIT
  END-IF
END-IF

IF NOT (DLOCX-I = SPACES OR = LOW-VALUES)
  MOVE 5 TO SLEN
  MOVE DLOCX-I TO SEARCH1
  PERFORM TRANSLATE-DLOC
  PERFORM COMPARE-STRINGS
  IF SEARCH-RESULT = Ø
    GO TO INQUIRE-CICS-LOOP-EXIT
  END-IF
END-IF

IF NOT (DKEYX-I = SPACES OR = LOW-VALUES)
  MOVE 4 TO SLEN
  MOVE DKEYX-I TO SEARCH1
  PERFORM TRANSLATE-DKEY
  PERFORM COMPARE-STRINGS
  IF SEARCH-RESULT = Ø
    GO TO INQUIRE-CICS-LOOP-EXIT
  END-IF
END-IF

ADD 1 TO X

MOVE W-TRAN TO TRANI(X)
MOVE W-PROG TO PROGI(X)
MOVE W-PROF TO PROFI(X)
MOVE W-TCLA TO TCLAI(X)
MOVE W-RNAM TO RNAMI(X)
MOVE W-RSYS TO RSYSI(X)

MOVE W-PRTY TO NUM-AUX
MOVE NUM-AUX(6:3) TO PRTYI(X)
MOVE W-TWAS TO NUM-AUX
MOVE NUM-AUX(5:4) TO TWASI(X)

PERFORM DISPLAY-STAT
PERFORM DISPLAY-PURG
PERFORM DISPLAY-DLOC
PERFORM DISPLAY-DKEY.

* INQUIRE-CICS-LOOP-EXIT.
******************
EXIT.
*

COMPARE-STRINGS.
***************
IF SEARCH-1(SLENG) = SPACE OR = '*'
    SUBTRACT 1 FROM SLENG
    GO TO COMPARE-STRINGS
END-IF
IF SEARCH1(1:SLENG) = SEARCH2(1:SLENG)
    MOVE 1 TO SEARCH-RESULT
ELSE
    MOVE 0 TO SEARCH-RESULT
END-IF.

* TRANSLATE-STATUS.
***************
IF W-STAT = DFHVALUE(ENABLED)
    MOVE 'ENA' TO SEARCH2
ELSE
    IF W-STAT = DFHVALUE(DISABLED)
        MOVE 'DIS' TO SEARCH2
END-IF.
*

TRANSLATE-DLOC.
***************
IF W-DLOC = DFHVALUE(BELOW)
    MOVE 'BELOW' TO SEARCH2
ELSE
    IF W-DLOC = DFHVALUE(ANY)
        MOVE 'ANY ' TO SEARCH2
END-IF.
*

TRANSLATE-DKEY.
***************
IF W-DKEY = DFHVALUE(USERDATAKEY)
MOVE 'USER' TO SEARCH2
ELSE
  IF W-DKEY = DFHVALUE(CICSDATAKEY)
    MOVE 'CICS' TO SEARCH2
  END-IF.
*
DISPLAY-STAT.
************************
  IF W-STAT = DFHVALUE(ENABLED)
    MOVE 'Ena' To STATI(X)
  ELSE
    IF W-STAT = DFHVALUE(DISABLED)
      MOVE 'Dis' TO STATI(X)
    END-IF.
*
DISPLAY-PURG.
****************
  IF W-PURG = DFHVALUE(PURGEABLE)
    MOVE 'Pur' TO PURGI(X)
  ELSE
    IF W-PURG = DFHVALUE(NOTPURGEABLE)
      MOVE ' ' TO PURGI(X)
    END-IF.
*
DISPLAY-DLOC.
****************
  IF W-DLOC = DFHVALUE(BELOW)
    MOVE 'Below' TO DLOCI(X)
  ELSE
    IF W-DLOC = DFHVALUE(ANY)
      MOVE 'Any ' TO DLOCI(X)
    END-IF.
*
DISPLAY-DKEY.
****************
  IF W-DKEY = DFHVALUE(USERDATAKEY)
    MOVE 'User' TO DKEYI(X)
  ELSE
    IF W-DKEY = DFHVALUE(CICSDATAKEY)
      MOVE 'Cics' TO DKEYI(X)
    END-IF.
*
CLEAN-SCREEN-LINES.
***********************
  PERFORM CLEAN-SCREEN-LINES-LOOP
      VARYING Z FROM X BY 1 UNTIL Z > 18.
*
CLEAN-SCREEN-LINES-LOOP.
***********************
  MOVE SPACES TO TRANI(Z) PROGI(Z) PROFI(Z)
      TCLAI(Z) STATI(Z) PURGI(Z)
* INITIATE-SCREEN.
***************
EXEC CICS ASSIGN APPLID (CICSNI)
END-EXEC
EXEC CICS ASKTIME ABSTIME (ABSTIME)
END-EXEC
EXEC CICS FOMATTIME
    ABSTIME (ABSTIME)
    DATE (DDATEI)
    DATESEP ('/')
    TIME (DTIMEI)
    TIMESSEP (':')
END-EXEC.
*
* RECEIVE-SCREEN.
***************
EXEC CICS HANDLE CONDITION MAPFAIL(RETURN-EXIT)
END-EXEC
EXEC CICS HANDLE AID PF3 (RETURN-EXIT)
    PF15 (RETURN-EXIT)
END-EXEC
EXEC CICS RECEIVE MAP('VITRANS')
END-EXEC.
*
SEND-SCREEN.
*************
EXEC CICS SEND MAP('VITRANS') DATAONLY
END-EXEC.
*
SEND-SCREEN-ERASE.
***************
EXEC CICS SEND MAP('VITRANS') ERASE
END-EXEC.
*
RETURN-TRANSID.
*************
EXEC CICS RETURN
    TRANSID (TRANS-NAME)
    COMMAREA (COMMAREA)
    LENGTH (EIBCALEN)
END-EXEC.
*
RETURN-EXIT.
*************
EXEC CICS SEND
    FROM (END-MESSAGE)
    LENGTH (3)
    ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC
GOBACK.

VITRANS BMS SOURCE CODE

MAPSET  DFHMDS TYPE=&SYSPARM,MODE=INOUT,CTRL=(FREEKB),
        LANG=COBOL,TIOAPFX=YES,EXTATT=MAPONLY

* VITRANS DFHMDI SIZE=(24,80)

* CICSN  DFHMDF POS=(01,04),LENGTH=08,ATTRB=(ASKIP,PROT),
        COLOR=yellow

* DDATE  DFHMDF POS=(01,57),LENGTH=10,ATTRB=(ASKIP,PROT),
        COLOR=yellow

* DTIME  DFHMDF POS=(01,68),LENGTH=08,ATTRB=(ASKIP,PROT),
        COLOR=yellow

* DFHMDF POS=(02,02),LENGTH=04,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Tran'

* DFHMDF POS=(02,07),LENGTH=07,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Program'

* DFHMDF POS=(02,16),LENGTH=07,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Profile'

* DFHMDF POS=(02,25),LENGTH=06,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Tclass'

* DFHMDF POS=(02,34),LENGTH=03,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Sta'

* DFHMDF POS=(02,38),LENGTH=03,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Pur'

* DFHMDF POS=(02,42),LENGTH=03,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Prt'

* DFHMDF POS=(02,46),LENGTH=04,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Twas'

* DFHMDF POS=(02,51),LENGTH=05,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Rname'

* DFHMDF POS=(02,60),LENGTH=04,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Rsyst'

* DFHMDF POS=(02,65),LENGTH=04,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Dloc'

* DFHMDF POS=(02,71),LENGTH=04,ATTRB=(ASKIP,PROT),
        COLOR=pink,INITIAL='Dkey'

* TRAN-X  DFHMDF POS=(03,02),LENGTH=04,ATTRB=(UNPROT,FSET,BRT,IC),
        COLOR=yellow

* PROG-X  DFHMDF POS=(03,07),LENGTH=08,ATTRB=(UNPROT,FSET,BRT),
        COLOR=yellow

* PROF-X  DFHMDF POS=(03,16),LENGTH=08,ATTRB=(UNPROT,FSET,BRT),
        COLOR=yellow
TCLA-X DFHMDF POS=(03,25),LENGTH=08,ATTRB=(UNPROT,FSET,BRT), COLOR=yellow
STAT-X DFHMDF POS=(03,34),LENGTH=03,ATTRB=(UNPROT,FSET,BRT), COLOR=yellow
RNAM-X DFHMDF POS=(03,51),LENGTH=08,ATTRB=(UNPROT,FSET,BRT), COLOR=yellow
RSYS-X DFHMDF POS=(03,60),LENGTH=04,ATTRB=(UNPROT,FSET,BRT), COLOR=yellow
DLOC-X DFHMDF POS=(03,65),LENGTH=05,ATTRB=(UNPROT,FSET,BRT), COLOR=yellow
DKEY-X DFHMDF POS=(03,71),LENGTH=04,ATTRB=(UNPROT,FSET,BRT), COLOR=yellow

* DFHMDF POS=(04,02),LENGTH=76,ATTRB=(ASKIP,PROT), COLOR=red,
INITIAL='-----------------------------------------------*
*-----------------------------------------------*
* TRAN-01 DFHMDF POS=(05,02),LENGTH=04,ATTRB=(ASKIP,PROT),COLOR=yellow
PROG-01 DFHMDF POS=(05,07),LENGTH=08,ATTRB=(ASKIP,PROT)
PROF-01 DFHMDF POS=(05,16),LENGTH=08,ATTRB=(ASKIP,PROT)
TCLA-01 DFHMDF POS=(05,25),LENGTH=08,ATTRB=(ASKIP,PROT)
STAT-01 DFHMDF POS=(05,34),LENGTH=03,ATTRB=(ASKIP,PROT)
PURG-01 DFHMDF POS=(05,38),LENGTH=03,ATTRB=(ASKIP,PROT)
PRTY-01 DFHMDF POS=(05,42),LENGTH=03,ATTRB=(ASKIP,PROT)
TWAS-01 DFHMDF POS=(05,46),LENGTH=04,ATTRB=(ASKIP,PROT)
RNAM-01 DFHMDF POS=(05,51),LENGTH=08,ATTRB=(ASKIP,PROT)
RSYS-01 DFHMDF POS=(05,60),LENGTH=04,ATTRB=(ASKIP,PROT)
DLOC-01 DFHMDF POS=(05,65),LENGTH=05,ATTRB=(ASKIP,PROT)
DKEY-01 DFHMDF POS=(05,71),LENGTH=04,ATTRB=(ASKIP,PROT)

* TRAN-02 DFHMDF POS=(06,02),LENGTH=04,ATTRB=(ASKIP,PROT),COLOR=yellow
PROG-02 DFHMDF POS=(06,07),LENGTH=08,ATTRB=(ASKIP,PROT)
PROF-02 DFHMDF POS=(06,16),LENGTH=08,ATTRB=(ASKIP,PROT)
TCLA-02 DFHMDF POS=(06,25),LENGTH=08,ATTRB=(ASKIP,PROT)
STAT-02 DFHMDF POS=(06,34),LENGTH=03,ATTRB=(ASKIP,PROT)
PURG-02 DFHMDF POS=(06,38),LENGTH=03,ATTRB=(ASKIP,PROT)
PRTY-02 DFHMDF POS=(06,42),LENGTH=03,ATTRB=(ASKIP,PROT)
TWAS-02 DFHMDF POS=(06,46),LENGTH=04,ATTRB=(ASKIP,PROT)
RNAM-02 DFHMDF POS=(06,51),LENGTH=08,ATTRB=(ASKIP,PROT)
RSYS-02 DFHMDF POS=(06,60),LENGTH=04,ATTRB=(ASKIP,PROT)
DLOC-02 DFHMDF POS=(06,65),LENGTH=05,ATTRB=(ASKIP,PROT)
DKEY-02 DFHMDF POS=(06,71),LENGTH=04,ATTRB=(ASKIP,PROT)

* TRAN-03 DFHMDF POS=(07,02),LENGTH=04,ATTRB=(ASKIP,PROT),COLOR=yellow
PROG-03 DFHMDF POS=(07,07),LENGTH=08,ATTRB=(ASKIP,PROT)
PROF-03 DFHMDF POS=(07,16),LENGTH=08,ATTRB=(ASKIP,PROT)
TCLA-03 DFHMDF POS=(07,25),LENGTH=08,ATTRB=(ASKIP,PROT)
STAT-03 DFHMDF POS=(07,34),LENGTH=03,ATTRB=(ASKIP,PROT)
PURG-03 DFHMDF POS=(07,38),LENGTH=03,ATTRB=(ASKIP,PROT)
PRTY-03 DFHMDF POS=(07,42),LENGTH=03,ATTRB=(ASKIP,PROT)
Editor's note: the code here is repeated (with appropriate modifications) for 04 to 17.

```
TRAN-18 DFHMDF POS=(22,02),LENGTH=04,ATTRB=(ASKIP,PROT),COLOR=.YELLOW
PROG-18 DFHMDF POS=(22,07),LENGTH=08,ATTRB=(ASKIP,PROT)
PROF-18 DFHMDF POS=(22,16),LENGTH=08,ATTRB=(ASKIP,PROT)
TCLA-18 DFHMDF POS=(22,25),LENGTH=08,ATTRB=(ASKIP,PROT)
STAT-18 DFHMDF POS=(22,34),LENGTH=03,ATTRB=(ASKIP,PROT)
PRTY-18 DFHMDF POS=(22,38),LENGTH=03,ATTRB=(ASKIP,PROT)
TWAS-18 DFHMDF POS=(22,42),LENGTH=03,ATTRB=(ASKIP,PROT)
RNAM-18 DFHMDF POS=(22,51),LENGTH=08,ATTRB=(ASKIP,PROT)
RSYS-18 DFHMDF POS=(22,60),LENGTH=04,ATTRB=(ASKIP,PROT)
DLOC-18 DFHMDF POS=(22,65),LENGTH=05,ATTRB=(ASKIP,PROT)
DKEY-18 DFHMDF POS=(22,71),LENGTH=04,ATTRB=(ASKIP,PROT)

DFHMDF POS=(23,02),LENGTH=76,ATTRB=(ASKIP,PROT), COLOR=RED, INITIAL='---------------------------*

DFHMDF POS=(24,06),LENGTH=16,ATTRB=(ASKIP,PROT), COLOR=NEUTRAL,INITIAL='ENTER Next page'

DFHMDF POS=(24,60),LENGTH=13,ATTRB=(ASKIP,PROT), COLOR=NEUTRAL,INITIAL='PF3/PF15 End'

DFHMSD TYPE=FINAL
END
```

Systems Programmer
(Portugal) © Xephon 2002

Interpreting CICS interval control behaviour

INTRODUCTION

CICS interval control processing provides support for the CICS API to perform time-related operations. These include the asynchronous starting of CICS transactions at some point in the future, the delaying of
programs by set periods of time, and the cancelling of outstanding start operations. Such operations are supported by the EXEC CICS START, EXEC CICS DELAY, and EXEC CICS CANCEL commands. Support for interval control processing within CICS Transaction Server has been enhanced in a number of ways when compared with CICS/ESA 4.1.0. This article discusses the relationship between interval control and temporary storage processing within CICS, and reviews the various changes made to this area of the product.

OVERVIEW OF TEMPORARY STORAGE USE BY INTERVAL CONTROL

Before discussing the enhancements to interval control within CICS Transaction Server, it is beneficial to understand the relationship between interval control and CICS temporary storage processing, since this has a bearing on the changes that have been made. Interval control represents an outstanding EXEC CICS START command by means of an Interval Control Element (ICE) control block within CICS. There is a relationship between ICEs and CICS temporary storage records. CICS utilizes temporary storage to hold FROM area data associated with EXEC CICS START commands. There are a number of subtleties to be aware of when considering this relationship.

If an EXEC CICS START command does not specify a REQID value or the PROTECT option, CICS will store the data passed by the FROM area into a system-generated temporary storage queue. The name is unique, and begins with the characters ‘DF’. The remaining suffix of the name is a hexadecimal alphanumeric counter, generated by CICS, which CICS increments for every new START request.

If an EXEC CICS START command specifies a REQID value, CICS uses this as the name of the temporary storage queue to be used to hold the data. Use of a REQID value allows a subsequent EXEC CICS CANCEL command to delete a previous EXEC CICS START request that has yet to expire; the REQID is used as the token to identify the corresponding start command to be cancelled.

If an EXEC CICS START command specifies the PROTECT option, this means that the start request is not to be scheduled by CICS until the task that issues the command has successfully syncpointed and
committed its Unit Of work (UOW) forwards. Therefore, CICS will defer the scheduling of the ICE associated with the EXEC CICS START command until the second phase of syncpoint has been reached. EXEC CICS START commands with the PROTECT option will have their FROM data stored by CICS in a temporary storage queue. If the command does not specify REQID, a unique queue name will be generated for use by the command. This will begin with an X'FC' character and be suffixed with a unique hexadecimal alphanumeric counter value. If REQID is specified on an EXEC CICS START command with the PROTECT option, the REQID value will be used as the name of the temporary storage queue to store the FROM data in.

Tasks that are started as the result of an EXEC CICS START command can refer to the FROM data passed on the command. An EXEC CICS RETRIEVE command will read the FROM data from the started task’s associated temporary storage queue and return it to the application. This will have the effect of deleting the temporary storage record (and its associated queue if now empty). If the started task does not issue an EXEC CICS RETRIEVE command, CICS will automatically delete the temporary storage data associated with the request as part of the implicit syncpoint during task detach. This tidy-up processing ensures that FROM data for previous START commands is not retained by the CICS system after started tasks have completed.

Information pertaining to unexpired ICEs (outstanding EXEC CICS START commands) is preserved by CICS across a warm or emergency restart of the system. On a warm restarted CICS system, ICEs are rebuilt by CICS restart processing and re-associated with their FROM data, as restored from temporary storage. On an emergency-restarted CICS system, ICEs are rebuilt only if their FROM data was written to a recoverable temporary storage destination. CICS can define specific temporary storage queues as being recoverable by providing RDO TSMODEL entries with associated prefix values for such queuenames, or by using the old-style DFHTST Temporary Storage Table and providing TYPE=RECOVERY entries. Recoverable temporary storage destinations provide the ability to undo changes made to such queues, and to be recovered across a CICS emergency restart.

THE START PROTECT OPTION AND THE TST

There is a subtle difference between CICS/ESA 4.1.0 and CICS
Transaction Server in the way in which CICS handles the recovery of temporary storage queues used to hold the FROM data associated with EXEC CICS START PROTECT commands.

In CICS/ESA 4.1.0, CICS would determine whether a temporary storage queue was recoverable or not by checking for a matching queue prefix in the DFHTST table. If a corresponding prefix was found in a TYPE=RECOVERY entry within the table, CICS would treat the queue as recoverable and so log any changes made to the queue, to enable both task dynamic transaction backout and system emergency restart operations. The logic was optimized to check for certain queuenames before interrogating the DFHTST. It recognized queuenames starting with an X'FC' character as being queues generated by EXEC CICS START PROTECT queues, and therefore recoverable by default. Therefore, X'FC' was not required as a TYPE=RECOVERY prefix within the table. However, before any checks were made on the recoverability of a queue, CICS/ESA 4.1.0 verified whether a DFHTST table existed at all. If no table was provided for that run of CICS, all temporary storage queues (including those prefixed by x ‘FC’ for EXEC CICS START PROTECT commands) were deemed non-recoverable. This was a known consequence of not needing to provide DFHTST table entries on a CICS/ESA 4.1.0 system. A solution was to provide a dummy DFHTST with no entries. This ensured that CICS/ESA 4.1.0 treated queues for EXEC CICS START PROTECT commands as recoverable temporary storage destinations.

With the introduction of a new object-oriented temporary storage domain in CICS Transaction Server, part of the redesign involved the creation of RDO-managed TSMODELs to define queue attributes (as provided in CICS Transaction Server 1.3). Recoverability is now tested for by checking for a corresponding queuname prefix match in the TSMODEL definitions (if RDO for temporary storage attributes is being used). This logic ensures that queues starting with an X'FC' character are treated as recoverable. The only time when they would not be is if CICS has been started with no auxiliary temporary storage support, and so only the unrecoverable main temporary storage environment is supported.

This change means that a dummy DFHTST is no longer required to ensure EXEC CICS START PROTECT queues are treated as recoverable.
when using CICS Transaction Server and exploiting RDO for temporary storage attribute definitions.

MULTIPLE START REQUESTS FOR THE SAME REQID

Prior to CICS Transaction Server, it was possible to issue multiple EXEC CICS START commands with the same non-recoverable REQID value specified. That is, several EXEC CICS START commands could write their FROM data to the same temporary storage queue, as specified by the REQID parameter. This was supported provided the queue in question was non-recoverable. If the REQID mapped to a recoverable temporary storage queue, only one such EXEC CICS START request for that queue was allowed to be outstanding at any one time. That is, another EXEC CICS START command could not be issued for the same REQID until the task to be started had retrieved its FROM data and then syncpointed. If subsequent EXEC CICS START commands were issued before this point, an IOERR condition was returned by CICS interval control processing which, if unhandled, resulted in an AEIQ abend.

The possibility of having several outstanding EXEC CICS START commands using the same non-recoverable temporary storage queue to hold their FROM data meant that it was possible for the data items to be retrieved from the queue in a different order to that which had been intended. This could have been the case if start operations were delayed for some reason (if, for example, the associated terminal for a started task was not available at the time that the ICE for that START operation had expired). Therefore, to avoid this problem, and to make interval control processing treat such queues in the same way it handled recoverable temporary storage destinations, CICS Transaction Server has restricted non-recoverable REQIDs to be used by only one outstanding EXEC CICS START command at any time. This ensures that problems with incorrect FROM data record retrieval are now avoided.

One consequence of this change is that applications using EXEC CICS START commands with FROM data need to ensure that their started tasks retrieve their data before issuing a further EXEC CICS START command with the same REQID, assuming they use such a means of propagating task activity within CICS.
PROTECTED STARTS AND RECOVERABLE QUEUES

The use of a recoverable temporary storage queue as the repository for FROM data written by an EXEC CICS START command means that there are some implications to consider regarding the task that is to be started. For example, if the task was started before the originating task had completed its UOW and syncpointed, an EXEC CICS RETRIEVE command issued from the started task would have to wait for release of the queue lock by the originating task. This could happen if the expiry interval period on the EXEC CICS START command was omitted, or was of short enough duration to expire before the originating task had completed. Expiry intervals are denoted by the use of INTERVAL on the command, by an explicit time value specified by TIME on the command, or by the use of the command option AFTER, with expiry details given in hours, minutes and/or seconds.

Another possible problem is if the task which had issued the EXEC CICS START command (without the PROTECT option) were to subsequently abend or issue an EXEC CICS SYNCPOINT ROLLBACK command, after the START command. This would back-out the write to the recoverable temporary storage queue. Therefore, a subsequent EXEC CICS RETRIEVE command issued by the started task would fail with a NOTFND condition (abend AEIM).

As part of general CICS redesign and improvement, CICS Transaction Server 1.3 now avoids these problems by treating EXEC CICS START requests with FROM data directed to recoverable temporary storage queues as being protected starts. This ensures that the start request will not be scheduled by CICS until after the originating task has completed its UOW and syncpointed. By committing a start request at syncpoint time, the temporary storage queue lock is released and so the task to be started will no longer have the potential to wait for the originating task to have freed its lock upon the queue. Also, had the originating task abended or rolled-back after the EXEC CICS START command was issued, the fact that it was deemed to be a protected start means that the start would not yet have been scheduled, so avoiding the potential for NOTFND conditions on subsequent EXEC CICS RETRIEVE commands.
SUMMARY AND CONTACT INFORMATION

The changes to CICS interval control processing outlined above are to help provide a consistent approach to the execution of EXEC CICS START commands. I hope that this article has helped explain the way in which CICS interval control processing has been enhanced and works in CICS Transaction Server 1.3 and above.

Andy Wright (andy_wright@uk.ibm.com)
CICS Change Team
IBM (UK) © IBM 2002

CLER – a new CICS transaction for displaying and changing some LE options

A new transaction, CLER, was introduced and shipped with Language Environment (LE) z/OS V1.1 and retrofitted to OS/390 V2.8, V2.9, and V2.10. It allows users to:

- Display the LE Region Runtime options.
- Write the LE Runtime options to the CESE transient data queue for printing.
- Modify a subset of the LE Runtime Options.

The transaction is conversational, presenting the user with a standard 3270 Model 2 terminal display. The Runtime options that can be modified with this transaction will only be in effect for the duration of the running of the CICS address space.

The resources for CLER transaction must be defined in the CICS CSD (CICS System Definition file). There are new entries in the LE CEECCSD job in the SCEESAMP dataset for this support. The four definitions that are required if you want to use the CEDA transaction are:

```
DEFINE TRANS(CLER) PROG(CEL4RTO) GROUP(CEE)
DEFINE PROGRAM(CEL4RTO) GROUP(CEE) LANGUAGE(ASSEMBLER) EXECKEY(CICS)
```
On invoking the CLER transaction the screen below is displayed showing the options that can be changed and their current values:

---Top of screen---

CLER

Language Environment Region Level Runtime Options

Type in your Choices.

<table>
<thead>
<tr>
<th>Runtime option</th>
<th>Choice</th>
<th>Possible choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAP</td>
<td>==&gt; ON</td>
<td>ON, OFF</td>
</tr>
<tr>
<td>RPTOPTS</td>
<td>==&gt; OFF</td>
<td>ON, OFF</td>
</tr>
<tr>
<td>RPTSTG</td>
<td>==&gt; OFF</td>
<td>ON, OFF</td>
</tr>
<tr>
<td>ALL31</td>
<td>==&gt; ON</td>
<td>ON, OFF</td>
</tr>
<tr>
<td>CBLPSHPOP</td>
<td>==&gt; ON</td>
<td>ON, OFF</td>
</tr>
<tr>
<td>TERMTHDACT</td>
<td>==&gt; TRACE</td>
<td></td>
</tr>
</tbody>
</table>

QUIET, MSG, TRACE, DUMP, UAONLY, UATRACE, UADUMP, UAIMM

When finished, press ENTER.

---Bottom of screen---

PFI=Help  3=Quit  5=Current Settings  9=Error List

The meaning of these options are:

- **TRAP** – whether Language Environment should handle error conditions and program interrupts.
- **RPTOPTS** – whether a report of the run-time options in use by the program should be generated.
- **RPTSTG** – whether a report of the storage used by the program should be generated.
- **ALL31** – whether the entire application is running AMODE 31.
- **CBLPSHPOP** – whether CICS PUSH and POP HANDLE are issued when a COBOL subroutine is called.
• TERMTTHDACT – sets the level of information produced because of an error of severity 2 or greater.

For more information on the other Runtime Options refer to the Language Environment manuals.

Note: if the Runtime Option ALL31 is modified to OFF, the stack will be forced below the 16MB line. A warning message will be issued on the screen for you to indicate whether you want to continue or not. Once the stack is modified to below, it will remain below for the duration of the region (even if you set ALL31 back to on).

For detailed documentation on use of the transaction select PF1=Help from the main menu that is displayed when the CLER transaction is invoked.

Language Environment Region Level Runtime Options.

Current Settings

<table>
<thead>
<tr>
<th>LAST WHERE SET</th>
<th>OPTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation default</td>
<td>PROFILE(OFF,&quot;&quot;)</td>
</tr>
<tr>
<td>Installation default</td>
<td>PRTUNIT(6)</td>
</tr>
<tr>
<td>Installation default</td>
<td>PUNUNIT(7)</td>
</tr>
<tr>
<td>Installation default</td>
<td>RDRUNIT(5)</td>
</tr>
<tr>
<td>Installation default</td>
<td>RECPAD(OFF)</td>
</tr>
<tr>
<td>CICS CLER Trans</td>
<td>RPTOPTS(OFF)</td>
</tr>
<tr>
<td>Region default</td>
<td>RPTSTG(OFF)</td>
</tr>
<tr>
<td>Installation default</td>
<td>NORTEREUS</td>
</tr>
<tr>
<td>Installation default</td>
<td>RTLS(OFF)</td>
</tr>
<tr>
<td>Installation default</td>
<td>NOSIMVRD</td>
</tr>
<tr>
<td>Storage user exit</td>
<td>STACK(4096,4080,ANYWHERE,KEEP,4096,4080)</td>
</tr>
<tr>
<td>Installation default</td>
<td>STORAGE(NONE,NONE,NONE,0)</td>
</tr>
</tbody>
</table>

PF: 1=Help 3=Quit 7=Back 8=Forward 10=RPT->CES

This set of six screens are extremely useful to see where an LE option was set from. The example above shows options being set by the Installation and Region defaults, the CLER transaction, and the newly-
introduced Storage user exit. To send the Runtime option report to the CESE queue for output display or printing, press the PF10 key when viewing the Runtime option screen above.

The list below shows the PTF numbers for the releases where this function has been retrofitted:

- UQ48504 – V2.R8
- UQ48505 – V2.R9

---

**CICS questions and answers**

**Q** We are new to Sockets in CICS. We seem to have found that when our Sockets programs run and we transmit data, all CICS transactions stop and wait until the transmit completes. Is this common? Have we missed some installation step?

**A** Sounds like you have the Batch version of EZASOCKET which causes all of CICS to wait on transmissions. You need to INCLUDE EZACICAL in your linkedit of the Sockets program, or if you’re writing in C you need to INCLUDE EZACIC17 (the new(ish) Re-entrant version – see APAR: PQ28963 for details of the re-entrant C CICS Sockets interface – this replaces EZACIC07, the NORENT version!).

*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.*

---

© Xephon 2002
Data21 has announced the availability of Release 2 of ZIP/CICS, its CICS compression tool.

The latest version provides: a built-in FTP client, which simplifies the compression and transmission of files; a built-in e-mail client, allowing files to be compressed and e-mailed in a single step; and a Windows self-extractor, for older Windows versions that may not have a separate unzip tool.

For further information contact:
Data 21, 3510 Torrance Blvd, Suite 300, Torrance, CA 90503, USA.
Tel: (310) 792 1771.

* * *

MacKinney Systems has announced VTAM Virtual Printer (VVP) Release 1.3, which is used to route print from CICS TS to the JES queue for OS/390 and z/OS or to the POWER Queue for VSE.

VVP is said to require no programming changes and it runs as a VTAM application. All printer maintenance may be done through on-line panels.

For further information contact:
MacKinney Systems, 2740 South Glenstone, Suite 103, Springfield, MI 65804, USA.
Tel: (417) 882 8012.

* * *

ClientSoft has announced Tanit Objects (CTO) a CICS solution for Web Services, which is, they claim, the only product to offer a direct, non-screen-scraping approach to building and integrating legacy components as Web services.

Application developers can create Web services for their core CICS transactions. Using one of three access methods, CTO creates customized snap-ins or components for the application server in the form of Enterprise JavaBeans (EJBs), Java Beans, COM, or Active X controls. These snap-ins or components enable the application server to interact directly with the mainframe without the need for data scraping or extra coding. This allows Web services to be implemented faster, at lower cost, and without opening up potential failure points or holes in the existing security system.

For further information contact:
ClientSoft, 8323 Northwest 12 Street, Suite 216, Miami, FL 33126, USA.
Tel: (305) 716 1007.

* * *

Compuware has announced general availability of STROBE 2.5 and introduced iSTROBE, a new browser-based tool for analysing application performance information.

iSTROBE enables interactive analysis of enterprise applications based on WebSphere MQ, CICS/TS, Visual Age for Java, DB2, IMS, and Unix System Services.

For further information contact:
Compuware, 31440 Northwestern Highway, Farmington Hills, MI 48334-2564, USA.
Tel: (248) 737 7300.