167

CICS update

October 1999

In this issue

CICS Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38030
From USA: 01144 1635 38030
E-mail: info@xephon.com

North American office

Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75077-2150
USA
Telephone: 940 455 7050

Contributions

Articles published in *CICS Update* are paid for at the rate of £170 ($250) per 1000 words and £90 ($140) per 100 lines of code for original material. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*.

CICS Update on-line

Code from *CICS Update* can be downloaded from our Web site at http://www.xephon.com/cicsupdate.html; you will need the user-id shown on your address label.

Editor

Robert Burgess

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.

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 January 1994 issue, are available separately to subscribers for £16.00 ($23.50) each including postage.

Printed in England.

Cold start next time

Many sites would like the ability to set CICS to cold start the next time it loads. This requirement comes from implementing changes that need a cold start to be ‘picked up’, for example new product installs, some SIT changes, RDO changes, etc. After the systems programmer has made the changes, he needs to coordinate a cold start with the operations or operations automation areas. This could involve a CLIST change, a simple note for the operator to remember (hopefully), or some other complex system! It would be far easier if the systems programmer could simply set a switch so that CICS will cold start the next time it is loaded.

Here is a very simple system to provide just this facility. It’s simple because it must not fail – a very complex system could have been designed and created, but this would be more risky. Simple is best!

This ‘Next Cold’ system works using JCL only. No programs are needed, so it can easily be ‘enabled’ and ‘disabled’ should it ever prove to be causing a problem (again, this is very simple).

The cold start itself is achieved by REPROing the initialization record into the CICS DFHGCD (global catalog), but using the REUSE option on the REPRO (the REUSE option means that the existing data in the file is cleared and the new records being REPROed will be the only ones in the file). A delete/redefine could have been used, but this carries the risk of deleting the GCD and then not being able to recreate it (because of space problems, perhaps).

In addition, the redefine would need to be set-up so that it takes the attributes from the existing GCD, and even then there could be problems because of extents not taken into consideration – again not as simple as a REPRO REUSE.

The rest of the system is simple JCL to wrap around the IDCAMS REPRO to cold start CICS. I have used a ‘flag’ system, but other systems could be used instead. We run CICS for MVS/ESA Version 4.1 running on OS/390 Version 2.6, but this system could be altered to run with other CICS versions and operating systems.

The system works as follows:

- Did CICS shut OK? This is done by attempting to delete a flag dataset that is created at CICS shutdown – a simple IEFBR14 step after the DFHSIP step to create a ‘SHUTOK’ dataset (see below). The ‘flag’ for CICS shutting down OK is done this way because, should there be a problem in creating this dataset, the following process is skipped:
  - If no, then skip all of the following steps.
  - If yes, is cold required? This is again done by attempting to delete a ‘flag’ dataset that is created by the systems programmer to flag a ‘Cold Next’. If the dataset exists, it is deleted and RC=0 is set, if it doesn’t exist then RC=8 and all of the following steps are skipped. The delete of this dataset also ensures that the cold start is performed only once.
- If cold is required (RC=0 from the previous step) then the REPRO is performed against the DFHGCD.
- CICS (DFHSIP step) starts cold – the following messages are issued by CICS during start-up:
  - DFHSI1518 <region>, no control record on global catalog dataset. Cold start forced.
  - DFHSI1502I <region>, CICS start-up is cold.
- If the DFHSIP step completes RC=0, then create the ‘SHUTOK’ dataset.

Here is the JCL:

```
//*
//*===================================================================*
//*          CHECK/DELETE SHUT OK FLAG                                *
//*===================================================================*
//*
//SHUTCHK  EXEC PGM=IKJEFTØ1,
//          PARM=('DELETE ',
//           ' ''&REGION..SHUTOK'' ')
//*------------------------------------------------------------------ *
//*          IKJEFTØ1 DATASETS                                        *
//*------------------------------------------------------------------ *
//SYSTSPRT DD  SYSOUT=*
```

```
//SYSPRINT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//*
// IF (SHUTCHK.RC = Ø) THEN
//*
//*
//*====================================================================*
//*          CHECK WHETHER COLD START REQUIRED?                        *
//*====================================================================*
//*
//ARCCHK   EXEC PGM=IKJEFTØ1,
//          PARM=('DELETE ',
//           ' ''&REGION..COLDNEXT'' ')
//*------------------------------------------------------------------- *
//*          IKJEFTØ1 DATASETS                                         *
//*------------------------------------------------------------------- *
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//*
// ENDIF
//*
//*
// IF (SHUTCHK.RC = Ø & ARCCHK.RC = Ø) THEN
//*
//*
//*====================================================================*
//*          RE-INIT DFHGCD TO FORCE ARCTIC START                      *
//*====================================================================*
//*
//ARCGCD   EXEC PGM=IDCAMS
//*------------------------------------------------------------------- *
//*          IDCAMS DATASETS                                           *
//*------------------------------------------------------------------- *
//SYSIN    DD  DSN=CICS.COLDINIT(REPRO),     <— REPRO INFILE(DFHINSTØ)
//             DISP=SHR,                         OUTFILE(DFHGCD) REUSE
//             FREE=CLOSE
//DFHINSTØ DD  DSN=CICS.COLDINIT(DFHINSTØ),  <— The Standard IBM init
//             DISP=SHR,
//             FREE=CLOSE
//DFHGCD   DD  DSN=&REGION..DFHGCD,
//             DISP=SHR
//SYSPRINT DD  SYSOUT=*
//*
// ENDIF
//*
.
.
CICS Step (DFHSIP)
.
```

```
.
//*
// IF (CICSSTEP.RC = Ø & ABEND = FALSE) THEN
//*
//*
//*===================================================================*
//*          FLAG CICS SHUTDOWN OK                                     *
//*===================================================================*
//*
//SHUTOK   EXEC PGM=IEFBR14
//*------------------------------------------------------------------ *
//*          SHUTOK DATASETS                                           *
//*------------------------------------------------------------------ *
//SHUTOK   DD  DSN=&REGION..SHUTOK,
//             DISP=(NEW,CATLG,DELETE),
//             DCB=(RECFM=FB,LRECL=8Ø,BLKSIZE=Ø),
//             UNIT=SYSALLDA,
//*            VOL=SER=??????,
//             SPACE=(8Ø,(1,1),RLSE),
//             FREE=CLOSE
//*
// ENDIF
//*
```

So, what else is required? A REXX EXEC or job to create the ‘Next Cold’ datasets for all the CICS regions requiring a cold start – perhaps a REXX that can scan SYS2.PROCLIB for members beginning CICST, CICSU, CICSP, etc, and then create ‘Next Cold’ datasets for each system found. Whatever it is, all that is needed is a ‘Next Cold’ dataset to cold start CICS at the next start-up (that is, if the previous shut down was OK).

Because each site will have its own way to identify CICS regions, and those needing a cold start, I have not included a system to create ‘Next Cold’ datasets – a simple IEFBR14 or ISPF create dataset will do.

The ‘side effects’ of this system mean that, by looking at the creation dates of the ‘flag’ datasets, it is possible to tell when a CICS cold start was requested but not yet done, and when a system was last shut-down (and therefore used).

John Lemmon
Systems Programming
Norwich Union (UK)

CICS allocation problems and SMF exit IEFUSI

INTRODUCTION

In our shop we run several CICS 4.1s in production, test, and development modes. The CICSs are connected to IMS/DBCTL and DB2. Since we migrated from MVS Version 5.1 to OS/390 Version 1.3, we have experienced virtual storage abends 80A/878 in the production CICS. The fetch of modules failed because of insufficient LSQA storage.

When the CICS attachment facility to DB2 starts, it allocates storage for the maximum number of threads that can be created for each transaction in the RCT. 5.5KB are allocated per active thread in the CICS high private area (LSQA, and subpools 229 and 230). The CICS–DBCTL connection also requires CICS high private area virtual storage. CICS allocates 9KB of LSQA storage and 3KB of subpool 230 storage per active thread.

To circumvent 80A/878 system abends for a JES2 environment, IBM recommends the following:

- Allocate the SWA above 16MB with the SWA=ABOVE parameter in the STCCLASS/JOBCLASS statement of the JES2PARM member (this was already done).
- Decrease the user region size for an STC or a Job, to make more space available to the system control blocks in the high private area (LSQA, SWA, and subpools 229 and 230). See *CICS/ESA Performance Guide S488ff* for more details of the CICS high private area.

There are two ways to limit program storage below 16MB in virtual storage – IEALIMIT and IEFUSI. IEFUSI is the preferred exit routine, having several advantages over IEALIMIT:

- IEFUSI is a separate load module in the link pack area. However, IEALIMIT must reside in the nucleus and so the routine must be link-edited again each time the system is IPLed with a different version of the nucleus.

- IEALIMIT can only set region values below the 16MB line, leaving values for the extended private area to be determined by the system.
- IEFUSI is easier to write and less susceptible to system changes.

To install the IEFUSI exit, it must be specified on the EXITS option of the SYS or SUBSYS parameter of the SMF PARMLIB member (SMFPRMXX). It must be re-enterable and refreshable.

The following IEFUSI exit, based on an IBM sample, is used to control the region size below and above the 16MB line and to set installation-defined data space defaults and limits.

The details of the exit are described in the source code.

IEFUSI

```
USI      TITLE 'IEFUSI - SMF Step Initialization Installation Exit'
IEFUSI   CSECT
IEFUSI   AMODE 31
IEFUSI   RMODE ANY
         SPACE 1
***********************************************************************
***********************************************************************
**                                                                   **
** IEFUSI: SMF Step Initialization Installation Exit Sample          **
**                                                                   **
***********************************************************************
***********************************************************************
**                                                                   **
** Function:                                                         **
**                                                                   **
**    IEFUSI receives control from the initiator before each job     **
**    step is started (prior to allocation).                         **
**                                                                   **
**    This exit is used to control the installation limits for       **
**    region, GETMAIN, and data spaces.                              **
**                                                                   **
**    The following installation values are defined at the end of    **
**    this module and should be tailored to the installation's needs:**
**                                                                   **
**    . MINBELOW: Minimum region size below 16MB.                    **
**    . MAXBELOW: Maximum region size below 16MB.                    **
**    . DEFABOVE: Default and minimum region size above 16MB.        **
**    . TSODFABV: Default and minimum region size above 16MB for TSO.**
**    . MAXABOVE: Maximum region size above 16MB.                    **
**    . DEFDSSIZ: Default data space size.                           **
**    . MAXDSSIZ: Maximum combined data space size for Address Space.**
**    . MAXDSNUM: Maximum number of data spaces for Address Space.   **
**                                                                   **
** Operation:                                                        **
**                                                                   **
**   1. Control region size and limits.                              **
**                                                                   **
**       This exit enforces installation-defined minimum, default,   **
**       and maximum values for region size and GETMAIN limits for   **
**       both the area below and the area above 16 megabytes.        **
**                                                                   **
**       a) Handle region size and limit below 16 megabytes.         **
**                                                                   **
**         -  Requested region is zero.                              **
**                                                                   **
**                                                                   **
**            Let system handle zero region size for started tasks,  **
**            otherwise:                                             **
**                                                                   **
**            Set region size   = (private area size) - 64KB.        **
**            Set GETMAIN limit = private area size.                 **
**                                                                   **
**         -  Requested region < installation minimum.               **
**                                                                   **
**                                                                   **
**            Set region size   = minimum region below 16MB.         **
**            Set GETMAIN limit = region size + 64KB.                **
**                                                                   **
**         -  Requested region < ((private area size) - 64KB).       **
**                                                                   **
**                                                                   **
**            Set region size   = requested region size.             **
**            Set GETMAIN limit = region size + 64KB.                **
**                                                                   **
**         -  Requested region > ((private area size) - 64KB).       **
**                                                                   **
**                                                                   **
**            Set region size   = (private area size) - 64KB.        **
**            Set GETMAIN limit = private area size.                 **
**                                                                   **
**         -  Final check: if region size > installation maximum.    **
**                                                                   **
**                                                                   **
**            Accept region size > installation maximum for started  **
**            tasks, otherwise:                                      **
**                                                                   **
**            Set region size   = installation maximum below 16MB.   **
**            Set GETMAIN limit = region size + 64KB.                **
**                                                                   **
**       b) Handle region size and limit above 16 megabytes.          **
**                                                                    **
**         -  Set default extended region size.                       **
**            --------------------------------                        **
**                                                                    **
**            Set default region size above 16MB to installation-     **
**            defined default.                                        **
**                                                                    **
**            Set special (reduced) installation-defined default      **
**            for TSO.                                                **
**                                                                    **
**         -  Set minimum extended region size.                       **
**            --------------------------------                        **
**                                                                    **
**            Use the installation-defined default extended region    **
**            size also as the minimum extended region size.          **
**                                                                    **
**         -  Set maximum extended region size.                       **
**            --------------------------------                        **
**                                                                    **
**            Allow any region size above the installation maximum    **
**            for started tasks.                                      **
**                                                                    **
**            Otherwise, enforce the installation-defined maximum     **
**            value for extended region size.                         **
**                                                                    **
**            No special TSO maximum value is required, as TSO users **
**            cannot request a higher size than that defined in the  **
**            UADS or RACF dataset.                                   **
**                                                                    **
**         -  Set maximum extended GETMAIN limit.                     **
**            ----------------------------------                      **
**                                                                    **
**            Set extended GETMAIN limit equal to extended region     **
**            size.                                                   **
**                                                                    **
**    2. Control data space defaults and limits.                      **
**                                                                    **
**       This exit sets the installation-defined values for           **
**                                                                    **
**       - Default data space size.                                   **
**                                                                    **
**       - Maximum combined size of all data spaces owned within an  **
**         Address Space.                                             **
**                                                                    **
**       - Maximum number of data spaces that may exist at any given **
**         time for an Address Space.                                 **
**                                                                    **
** Recovery-operation:                                                **
**                                                                    **
**    An ESTAE is set up by the calling module IEFSMFIE.             **
**                                                                   **
** Register-Conventions:                                             **
**    Register      13 - Save area provided by caller. (No own save  **
**                       area is required by this sample exit).      **
**    Register      12 - Module base.                                **
**    Register      11 - Job Management Record JMR.                  **
**    Register       8 - Exit parameter list.                        **
**    Register       7 - VSM parameter area (VSMPARM).               **
**    Register       6 - RSM parameter area (RSMPARM).               **
**                                                                   **
** Attributes:                                                       **
**    State:         Supervisor                                      **
**    Key:           Ø                                               **
**    Mode:          Task                                            **
**    Amode:         31                                              **
**    Rmode:         Any                                             **
**    Type:          Reentrant                                       **
**    Serialization: Enabled / Unlocked                              **
**    ASC Mode:      Primary (HASID=PASID=SASID)                     **
**                                                                   **
** Entry-Registers:                                                  **
**    Register       Ø - Irrelevant                                  **
**    Register       1 - Address of the parameter list (EXTPARM)     **
**    Registers   2-12 - Irrelevant                                  **
**    Register      13 - Address of standard 72 byte                 **
**                       save area                                   **
**    Register      14 - Return address                              **
**    Register      15 - Entry address of IEFUSI                     **
**                                                                   **
** Input:                                                            **
**    Register one points to the following list of addresses:        **
**    1. Address of the common exit parameter area mapped by IEFJMR. **
**    2. Address of an eight-byte area containing the job step name. **
**    3. Address of an eight-byte area containing the program name   **
**       from the EXEC statement.                                    **
**    4. Address of an area containing the accounting information of **
**       the EXEC statement. The first byte of the accounting infor- **
**       mation contains the number of following accounting fields.  **
**       Each entry for an accounting field contains the length of   **
**       the field (one-byte, binary) followed by the field. A zero  **
**       in the length byte indicates an omitted field.              **
**    5. Address of the VSM parameter area that IEFUSI can use to    **
**       communicate the region and GETMAIN limits below and above   **
**       the 16 megabyte line to VSM.                                **
**    6. Address of the SRM information fullword. The first bit in   **
**       the first byte of this fullword indicates a V=R job.        **
**    7. Address of the RSM parameter area containing default and    **
**       limiting values for data spaces.                            **
**                                                                   **
** Output:                                                           **
**                                                                   **
**    . A return code indicating whether job processing should       **
**      continue or should be cancelled.                             **
**                                                                   **
**    . Updated VSM parameter area.                                  **
**                                                                   **
**    . Updated RSM parameter area.                                  **
**                                                                   **
** Exit-Registers:                                                   **
**    Registers   Ø-14 - Restored to contents at entry               **
**    Register      15 - Return code                                 **
**                                                                   **
** Return-Codes:                                                     **
**    Ø - Job processing should continue.                            **
**    4 - Job processing should be cancelled.                        **
**                                                                   **
**                                                                   **
** Control-Blocks:                                                   **
**                                                                   **
**    Common Name/Use  Macro ID  Full Name                           **
**    _______________  ________  _________                           **
**    ASCB       R     IHAASCB   Address Space Control Block.        **
**    CVT        R     CVT       Communication Vector Table.         **
**    GDA        R     IHAGDA    VSM's Global Data Area.             **
**    JMR        R     IEFJMR    Job's Job Management Record.        **
**    OUCB       R     IRAOUCB   Resources manager User Control Block**
**    PSA        R     IHAPSA    Prefixed Save Area.                 **
**                                                                   **
**    (C=Create, R=Read, W=Write, D=Delete)                          **
**                                                                   **
** Macros-Executable:                                                **
**    SAVE                                                           **
**                                                                   **
***********************************************************************
***********************************************************************
         EJECT 1
IEFUSI   CSECT
         SAVE  (14,12),,IEFUSI.&SYSDATE..&SYSTIME
         SPACE 1
         LR    R12,R15             Load base register.
         USING IEFUSI,R12          Set program base addressability.
         LR    R8,R1               Load address of parameter list.
         USING EXTPARM,R8          Set parameter list addressability.
         L     R11,EXTAEPA         Load address of exit parameter area.
         USING JMR,R11             Map SMF exit parameter area to JMR.
         SPACE 1
*        WTO   'IEFUSI has become active'
***********************************************************************
* Handle region / GETMAIN limits.                                     *
```

```
************************************************************************
         ICM   R7,15,EXTAVSMP      Fetch pointer to VSM parameter list.
         USING VSMPARM,R7          VSM parameter list addressability.
         BZ    LIMRGN9             Bypass if not provided.
         ICM   R1,15,EXTASRMI      Fetch pointer to SRM info. word.
         BZ    LIMRGN9             Bypass if not provided.
         TM    Ø(R1),X'8Ø'         V=R job ?
         BO    LIMRGN9             Bypass if V=R job.
         SPACE 1
         L     R5,CVTPTR           Load address of CVT.
         USING CVTMAP,R5           Establish CVT addressability.
         L     R14,CVTGDA          Load GDA address.
         L     R14,GDAPVTSZ-GDA(,R14) Fetch private area size from GDA.
         DROP  R5                  Release CVT base.
         SPACE 1
         L     R5,PSAAOLD-PSA      Load address of our ASCB.
         USING ASCB,R5             Establish ASCB addressability.
         L     R5,ASCBOUCB         Load address of OUCB.
         DROP  R5                  Release ASCB base.
         USING OUCB,R5             Establish OUCB addressability.
         LA    R15,1               Preset STC switch (R15 > Ø).
         CLC   OUCBSUBN(3),=C'STC' Is it started task ?
         BE    LIMRGN              Yes, continue.
         LNR   R15,R15             Preset TSO switch (R15 < Ø).
         CLC   OUCBSUBN(3),=C'TSO' Is it TSO session ?
         BE    LIMRGN              Yes, continue.
         SR    R15,R15             Else, set batch switch (R15 = Ø).
         DROP  R5                  Release OUCB base.
         SPACE 1
************************************************************************
* Handle REGION = ØK.                                                  *
* ------------------                                                   *
* STC: Let system handle REGION=ØK, otherwise:                         *
*                                                                      *
* . Set region size equal to system's private area size - 64KB.        *
* . Set GETMAIN limit equal to system's private area size.             *
************************************************************************
LIMRGN   DS    ØH
         ICM   RØ,15,VSMREGRQ      Fetch requested region size.
         BNZ   LIMRGNØØ            Bypass if region > Ø.
         LTR   R15,R15             Is it started task (STC) ?
         BP    LIMRGN9             Let system handle STC REGION=ØK.
         B     LIMRGNØ2            Else, go limit to private area size.
         SPACE 1
************************************************************************
* Handle REGION > ØK < installation minimum.                           *
* -------------------------------------------                          *
* . Set minimum region below to installation-defined value.            *
* . Add 64KB for the GETMAIN limit.                                    *
************************************************************************
```

```
LIMRGNØØ DS    ØH
         C     RØ,MINBELOW         Region < installation minimum ?
         BNL   LIMRGNØ1            No, bypass.
         L     RØ,MINBELOW         Else, use installation minimum for
         LR    R1,RØ               region size below 16MB and
         LR    R2,RØ               GETMAIN limit below 16MB.
         A     R2,K64              Add 64KB to GETMAIN limit.
         B     LIMRGN1             Go save new values.
         SPACE 1
************************************************************************
* Handle REGION < system's private area size.                          *
* ------------------------------------------                           *
* . Set region size equal to requested region.                         *
* . Set GETMAIN limit equal to requested region + 64KB.                *
************************************************************************
LIMRGNØ1 DS    ØH
         LR    R1,RØ               Preset region size = req. region.
         LR    R2,RØ               Preset GETMAIN limit = req. region.
         A     R2,K64              Add 64KB to GETMAIN limit.
         CR    R2,R14              Is GETMAIN limit > private area ?
         BNH   LIMRGN1             No, go use these values.
         SPACE 1
************************************************************************
* Handle REGION > system's private area size.                          *
* ------------------------------------------                           *
* . Set region size equal to system's private area size - 64KB.        *
* . Set GETMAIN limit equal to system's private area size.             *
************************************************************************
LIMRGNØ2 DS    ØH
         LTR   R1,R14              Fetch system's private area size.
         BZ    LIMRGN1             Bypass if zero.
         S     R1,K64              Subtract 64KB for region size.
         LR    R2,R14              Set GETMAIN limit to priv. area size
         B     LIMRGN1             Go, save the new values.
         SPACE 1
************************************************************************
* Handle REGION > installation-defined  maximum.                       *
* --------------------------------------------                         *
* . STC:       Accept Region > installation-defined maximum,           *
*              with the exception of the CICS STC with the             *
*              stepname CICPRD.                                        *
* . Batch/TSO: Set region size to installation-defined maximum.        *
*              Set GETMAIN limit to inst. defined maximum + 64KB.      *
************************************************************************
LIMRGN1  DS    ØH
         C     R2,MAXBELOW         GETMAIN limit > inst. maximum ?
         BNH   LIMRGN3             No, OK
         L     R4,EXTASTPN         Load address of step name
         CLC   Ø(6,R4),STEPNAME    Compare stepname
         BE    LIMRGN2             Set 7.5 MB REGION Size for STC
         LTR   R15,R15             Started task ?
         BP    LIMRGN3             Yes, honour it.
LIMRGN2  DS    ØH
         L     R1,MAXBELOW         Set region limit to inst. maximum.
         LR    R2,R1               Set GETMAIN limit to inst. maximum.
         A     R2,K64              Add 64KB to GETMAIN limit.
LIMRGN3  DS    ØH
         ST    R1,VSMRLIMB         Store REGION limit below 16MB.
         ST    R2,VSMGLIMB         Store GETMAIN limit below 16MB.
         SPACE 1
************************************************************************
* Handle extended region default values.                               *
* --------------------------------------                               *
* Set default region size above 16MB to installation-defined default. *
* . TSO:       Use special (reduced) TSO extended region size default.*
* . Batch/STC: Set installation default extended region size value.   *
* Use default extended region size as minimum extended region size.   *
************************************************************************
         L     R3,TSODFABV         Preset TSO default extended region.
         LTR   R15,R15             Is it TSO ?
         BM    LIMRGN4             Yes, use TSO special default/min.
         L     R3,DEFABOVE         Else, use batch/STC default/minimum.
LIMRGN4  DS    ØH
         LR    R1,RØ               Preset max. region = req. region.
         CR    RØ,R3               Region request > inst. default ?
         BH    LIMRGN5             Yes, bypass.
         LR    R1,R3               Set installation default as minimum.
         SPACE 1
************************************************************************
* Handle extended region limits.                                       *
* ------------------------------                                       *
* Set maximum region size above 16MB to installation-defined maximum. *
* . STC:   Allow any region size for started tasks.                    *
* . Batch: Limit extended region to installation-defined batch max.   *
* . TSO:   Use batch maximum value.                                    *
*          No special TSO maximum values are required, as TSO user    *
*          cannot request more than specified in UADS or RACF.        *
************************************************************************
LIMRGN5  DS    ØH
         LTR   R15,R15             Is it started task ?
         BP    LIMRGN6             Yes, accept region > inst. maximum.
         C     RØ,MAXABOVE         Is region request < inst. maximum ?
         BL    LIMRGN6             Yes, OK
         L     R1,MAXABOVE         Else, set installation maximum.
LIMRGN6  DS    ØH
         ST    R1,VSMRLIMA         Store extended region limit.
         ST    R1,VSMGLIMA         Use same value for ext. GETMAIN lim.
         OI    VSMFLAGW,X'8Ø'      Set flags "use IEFUSI limits",
         NI    VSMFLAGW,255-X'4Ø'  "check if region < 16MB fits",
         OI    VSMFLAGW,X'2Ø'      "check if region > 16MB fits".
```

```
LIMRGN9  DS    ØH
         SPACE 1
***********************************************************************
* Handle data space limits.                                           *
***********************************************************************
         ICM   R6,15,EXTARSMP      Fetch pointer to RSM parameter list.
         BZ    LIMDS9              Bypass, if not provided.
         USING RSMPARM,R6          Set RSM parm list addressability.
         SPACE 1
***********************************************************************
* Set installation-defined  default data space size.                   *
***********************************************************************
         L     R1,DEFDSSIZ         Fetch installation-defined default.
         ST    R1,RSMDFSIZ         Pass value to RSM.
         SPACE 1
***********************************************************************
* Set installation-defined maximum combined size for all user key     *
* data spaces owned within an Address Space.                          *
***********************************************************************
         L     R1,MAXDSSIZ         Fetch installation-defined maximum.
         ST    R1,RSMMXSIZ         Pass value to RSM.
         SPACE 1
***********************************************************************
* Set installation-defined maximum number of user key data spaces     *
* that may exist at any given time for an Address Space.              *
***********************************************************************
         L     R1,MAXDSNUM         Fetch installation-defined maximum.
         ST    R1,RSMMXNUM         Pass value to RSM.
LIMDS9   DS    ØH
         SPACE 1
***********************************************************************
*                                                                     *
* Return to calling program.                                          *
*                                                                     *
***********************************************************************
         SPACE 1
RETURN   DS    ØH
         LM    R14,R12,12(R13)     Restore registers at entry.
         SR    R15,R15             Set zero return code.
         BSM   Ø,14                Return to caller in its mode.
         SPACE  1
***********************************************************************
*                                                                     *
* Constants.                                                          *
*                                                                     *
***********************************************************************
K64      DC    A(64*1Ø24)          64KB constant.
         SPACE 1
***********************************************************************
* Define stepname to limit region > maxbelow                          *
```

```
************************************************************************
STEPNAME DC    CL6'CICPRD'
         DC    C'  '
         SPACE 1
************************************************************************
* Define installation minimum region size below the 16MB line.         *
************************************************************************
MINBELOW DC    A(1024*1024-64*1024) Set MINBELOW = 1MB - 64KB.
         SPACE 1
************************************************************************
* Define installation maximum region size below the 16MB line.         *
* Default: MAXBELOW DC    A(8*1024*1024-64*1024)                       *
* MAXBELOW DC    A(8*1024*1024-64*1024) Set MAXBELOW = 8MB - 64KB.     *
************************************************************************
MAXBELOW DC    A(7864320-64*1024) Set MAXBELOW = 7.5MB - 64KB.
         SPACE 1
************************************************************************
* Define installation default/minimum region size above the 16MB line.*
************************************************************************
DEFABOVE DC    A(32*1024*1024)     Set DEFABOVE = 32MB.
         SPACE 1
************************************************************************
* Define TSO installation default/minimum region above the 16MB line. *
************************************************************************
TSODFABV DC    A(16*1024*1024)     Set TSODFABV = 16MB.
         SPACE 1
************************************************************************
* Define installation maximum region size above the 16MB line.
************************************************************************
MAXABOVE DC    A(64*1024*1024)     Set MAXABOVE = 64MB.
         SPACE 1
************************************************************************
* Define installation default data space size in 4KB blocks.           *
*        System supplied default:  239 blocks (956KB).                 *
************************************************************************
DEFDSSIZ DC    A(956*1024/4096)    Set Inst. default = system default.
         SPACE 1
************************************************************************
* Define installation maximum combined size for all user key data      *
* spaces owned within an Address Space in megabytes.                   *
*        System supplied default:  256MB.                              *
************************************************************************
MAXDSSIZ DC    A(4096)             Set inst. maximum to 4096MB = 4GB.
         SPACE 1
************************************************************************
* Define installation maximum number of user key data spaces that may *
* exist at any given time for an Address Space.                        *
*        System supplied default:  256.                                *
************************************************************************
MAXDSNUM DC    A(64)               Set inst. maximum to 64.
```

```
         SPACE 1
*
         LTORG
         EJECT 1
         SPACE 1
***********************************************************************
* Register assigments.                                                *
***********************************************************************
RØ       EQU   Ø
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         SPACE 1
***********************************************************************
* Definition of IEFUSI installation exit parameter list.              *
***********************************************************************
EXTPARM  DSECT                     Pointed to by register one.
EXTAEPA  DS    A                   Address of SMF exit parameter area.
EXTASTPN DS    A                   Pointer to step name.
EXTAPGM  DS    A                   Pointer to program name.
EXTASACI DS    A                   Pointer to step accounting info.
EXTAVSMP DS    A                   Pointer to VSM parm list (VSMPARM).
EXTASRMI DS    A                   Pointer to SRM info. fullword.
EXTARSMP DS    A                   Pointer to RSM parm list (RSMPARM).
         SPACE 1
***********************************************************************
* Definition of Virtual Storage Management Parameter List (VSMPARM).  *
***********************************************************************
VSMPARM  DSECT                     Pointed to by EXTAVSMP.
VSMFLAGW DS    F                   VSM flag word
VSMREGRQ DS    F                   Region size requested.
VSMGLIMB DS    F                   GETMAIN limit below 16MB.
VSMRLIMB DS    F                   Region  limit below 16MB.
VSMGLIMA DS    F                   GETMAIN limit above 16MB.
VSMRLIMA DS    F                   Region  limit above 16MB.
         SPACE 1
***********************************************************************
* Definition of Real Storage Management Parameter List (RSMPARM).     *
***********************************************************************
```

```
RSMPARM  DSECT                         Pointed to by EXTARSMP.
RSMDFSIZ DS    F                       Default data space size in 4KB.
*                                      System default: 239 blocks (956KB).
RSMMXSIZ DS    F                       Maximum combined size for all
*                                      user key data spaces owned
*                                      within an Address Space.
*                                      System default: 256MB.
RSMMXNUM DS    F                       Maximum number of user key
*                                      data spaces that may exist at any
*                                      given time for an Address Space.
*                                      System default: 256
         SPACE 1
         PRINT OFF
***********************************************************************
* DSECT for Communication Vector Table (CVT).                         *
***********************************************************************
         CVT   DSECT=YES,PREFIX=YES,LIST=YES
         SPACE 1
***********************************************************************
* DSECT for Address Space Control Block (ASCB).                       *
***********************************************************************
         IHAASCB
         SPACE 1
***********************************************************************
* DSECT for Resources manager User Control Block (OUCB).              *
***********************************************************************
         IRAOUCB
         SPACE 1
***********************************************************************
* DSECT for Prefixed Save Area (PSA).                                 *
***********************************************************************
         IHAPSA
         SPACE 1
***********************************************************************
* DSECT for VSM's Global Data Area (GDA).                             *
***********************************************************************
         IHAGDA
         SPACE 1
***********************************************************************
* DSECT for JOB's SMF Job Management Record (JMR).                    *
***********************************************************************
         IEFJMR
         PRINT ON
         SPACE 1
         END
```

Giselher Wieland (Germany) © Xephon 1999

Accessing CICS control blocks in COBOL – part 2

This month we conclude the program that examines internal CICS structures in COBOL .

```
      *
      * Indicate the domain data being shown.
      *
           MOVE DOMAIN TO DDOMIDO
      *
      * Place the Job ID, Job Name and CICS release in the output map.
      *
           PERFORM ADDRESS-JSAB
           MOVE JSAB-JOBID   TO DJOBIDO
           MOVE JSAB-JOBNAME TO DJOBNMO
           MOVE WHAT-VERSION TO DRELO
      *
      * Determine how much data to display (maximum of 256 bytes per
      * page) and allow for short lines.
      *
           COMPUTE WORK-LTH = (CURRENT-PAGE - 1) * 256
           COMPUTE WORK-PTR = START-WORK + WORK-LTH
           COMPUTE WORK-LTH = TOTAL-LTH  - WORK-LTH
           SET ADDRESS OF ANCHOR TO ADDR-PTR
           IF  WORK-LTH > 256
               MOVE 256 TO WORK-LTH
               MOVE 15 TO LINE-LIMIT
               MOVE  4 TO SEG-LIMIT,
                          BYTE-LIMIT
           ELSE
               COMPUTE LINE-LIMIT = WORK-LTH
                                  / LENGTH OF A-LINE
               END-COMPUTE
               COMPUTE WORK-LIMIT = WORK-LTH
                                  - (LINE-LIMIT
                                  * LENGTH OF A-LINE)
               END-COMPUTE
               COMPUTE SEG-LIMIT  = WORK-LIMIT
                                  / LENGTH OF A-WORD
               END-COMPUTE
               COMPUTE BYTE-LIMIT = WORK-LIMIT
                                  - (SEG-LIMIT
                                  * LENGTH OF A-WORD)
               END-COMPUTE
           END-IF
      *
      * Do up to 16 lines of output.
```

```
*
     PERFORM VARYING VECTOR-IX FROM Ø BY 1
     UNTIL   VECTOR-IX > 15
         CALL 'HEXMANIP' USING ADDR-PTR,
                               LENGTH-4,
                               OA-ADDR
*
*        Keep the offset relative to the start of the block.
*
         ADD 16 TO WORK-PTR
         COMPUTE WORK-OFF = (VECTOR-IX * 16)
                          + (CURRENT-PAGE - 1) * 256
         CALL 'HEXMANIP' USING WO-LOW,
                               LENGTH-3,
                               OA-OFF
*
*        Display the data in both hexadecimal and 'printable'
*        character form.
*
         MOVE A-LINE(VECTOR-IX + 1) TO OA-EBCDIC
         PERFORM VARYING IX FROM 1 BY 1 UNTIL IX > 16
             IF  (OA-EBCDIC(IX:1) NOT ALPHABETIC )
             AND (OA-EBCDIC(IX:1) NOT NUMERIC    )
             AND (OA-EBCDIC(IX:1) NOT PUNCTUATION)
                 MOVE '.' TO OA-EBCDIC(IX:1)
             END-IF
         END-PERFORM
*
*        Allow for short pages.
*
         EVALUATE TRUE
             WHEN (VECTOR-IX < LINE-LIMIT)
                 MOVE 4 TO DATA-LIMIT,
                           WORK-LIMIT
             WHEN (VECTOR-IX = LINE-LIMIT)
                 IF  SEG-LIMIT  = Ø
                 AND BYTE-LIMIT = Ø
                     MOVE SPACES TO ADDRLNO(VECTOR-IX + 1)
                     SUBTRACT 1 FROM LINE-LIMIT
                 ELSE
                     MOVE SEG-LIMIT TO DATA-LIMIT
                     MOVE SPACES
                      TO  OA-EBCDIC((SEG-LIMIT * 4) +
                                     BYTE-LIMIT     +
                                     1:
                                    ((4 - SEG-LIMIT) * 4) +
                                     (4 - BYTE-LIMIT))
                 END-IF
             WHEN (VECTOR-IX > LINE-LIMIT)
```

```
                   MOVE SPACES TO ADDRLNO(VECTOR-IX + 1)
           END-EVALUATE
      *
      *    Allow for short lines.
      *
           IF  VECTOR-IX <= LINE-LIMIT
               PERFORM VARYING IW FROM 1 BY 1
               UNTIL   IW > DATA-LIMIT
                   CALL 'HEXMANIP' USING
                                   A-WORD(VECTOR-IX + 1, IW),
                                   LENGTH-4,
                                   OA-DATA(IW)
               END-PERFORM
      *
      *        Allow for areas which are not multiple words.
      *
               IF  BYTE-LIMIT < 4
               AND DATA-LIMIT < 4
               AND IW > DATA-LIMIT
               CALL 'HEXMANIP' USING
                               A-WORD(VECTOR-IX + 1, IW),
                               BYTE-LIMIT,
                               OA-DATA(IW)
               END-IF
               MOVE OUT-ADDR TO ADDRLNO(VECTOR-IX + 1)
               MOVE SPACES   TO OUT-ADDR
           END-IF
       END-PERFORM
       EXEC CICS SEND
                 MAP('DETLMAP')
                 MAPSET('ADDRMAP')
                 ERASE
                 CURSOR
                 NOHANDLE
       END-EXEC
       .
   ADDRESS-JSAB.
  *
  * Get access to the Job information. Note that Job Name can
  * be obtained from CICS via the INQUIRE SYSTEM command, but
  * the Job ID cannot be obtained from CICS.
  *
       SET ADDRESS OF ASCB TO PSA-ASCB-PTR
       SET ADDRESS OF ASSB TO ASCB-ASSB-PTR
       SET ADDRESS OF JSAB TO ASSB-JSAB-PTR
       .
   ADDRESS-KCB.
  *
  * The technique used here is based on the technique the CICS
```

```
* stub programs use to discover the location of DFHEIP. Thus
* it is unlikely that this code will need to change for a new
* release, although that cannot be guaranteed.
*
     SET ADDRESS OF TCB    TO PSA-TCB-PTR
     SET ADDRESS OF TCBEXT TO TCB-TCBEXT-PTR
     SET ADDRESS OF AFCB   TO TCBEXT-AFCB-PTR
     SET ADDR-PTR          TO ADDRESS OF AFCB
     ADD AFCB-VLIST-LTH    TO WORK-PTR
     ADD LENGTH  OF AFCB   TO WORK-PTR
     SET ADDRESS OF AFT    TO ADDR-PTR
     SET ADDRESS OF AFCS   TO AFT-AFCS-PTR
     SET ADDRESS OF KCB    TO AFCS-KCB-PTR
     .
 SEND-CONTROL.
*
* This is used to send error messages, mainly when
* the wrong key has been pressed by the user.
*
     IF  INPUT-MAIN
         MOVE LOW-VALUES TO MAINMAPO
         MOVE 'Place cursor on a valid domain and press <Enter>'
          TO  MMSGO
         MOVE DFHBMASB TO MMSGA
         MOVE -1       TO ANCHLNL(1)
         EXEC CICS SEND
                   MAP('MAINMAP')
                   MAPSET('ADDRMAP')
                   DATAONLY
                   ALARM
                   CURSOR
                   FREEKB
         END-EXEC
     ELSE
         MOVE LOW-VALUES TO DETLMAPO
         MOVE MSG(MSGNO) TO DMSGO
         MOVE DFHBMASB TO DMSGA
         EXEC CICS SEND
                   MAP('DETLMAP')
                   MAPSET('ADDRMAP')
                   DATAONLY
                   ALARM
                   FREEKB
         END-EXEC
     END-IF
     EXEC CICS RETURN
               TRANSID(EIBTRNID)
               COMMAREA(DFHCOMMAREA)
     END-EXEC
     .
 IDENTIFICATION DIVISION.
*
* This source code takes advantage of COBOL's ability to embed
* one program within another. In this way we obtain similar
* functionality exhibited by other languages such as PL/I and
* C — the ability to call a function with parameters.
*
* This embedded program (HEXMANIP) is used to convert data
* into hexadecimal display format which is needed in several
* places in the main program for addresses as well as data.
*
 PROGRAM-ID. HEXMANIP.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
*
* These are the various work areas required by the logic.
*
 Ø1  FILLER.
     Ø3  IX                    PIC 9(Ø4) COMP.
     Ø3  IH                    PIC 9(Ø4) COMP.
     Ø3  IL                    PIC 9(Ø4) COMP.
     Ø3  WS-HALF               PIC 9(Ø4) COMP.
     Ø3  FILLER REDEFINES WS-HALF.
         Ø5  WS-HIGH           PIC X(Ø1).
         Ø5  WS-LOW            PIC X(Ø1).
     Ø3  HEX-TABLE             PIC X(16) VALUE
         'Ø123456789ABCDEF'.
     Ø3  FILLER REDEFINES HEX-TABLE.
         Ø5  HEX-CHAR          PIC X(Ø1) OCCURS 16.
 LINKAGE SECTION.
*
* This is the data to be converted to hexadecimal display format.
*
 Ø1  DATA-VALUE.
     Ø3  WP-BYTE               PIC X(Ø1) OCCURS 4.
*
* This is the size of the data to be converted.
*
 Ø1  VALUE-LENGTH              PIC 9(Ø4) BINARY.
*
* This is the destination area where the
* converted data is to be placed.
*
 Ø1  VALUE-DEST                PIC X(Ø8).
 PROCEDURE DIVISION USING DATA-VALUE, VALUE-LENGTH, VALUE-DEST.
*
* Deal with one byte at a time.
*
```

```
     PERFORM VARYING IX FROM 1 BY 1 UNTIL IX > VALUE-LENGTH
*
*        Because COBOL does not have bit manipulation
*        capabilities, we must play about to get the values
*        we need to do the conversion. For each byte we need
*        to generate two hexadecimal characters. The part of
*        the value above 16 needs to be reflected in the first
*        hexadecimal digit and the part of the value below 16
*        needs to be reflected in the second hexadecimal digit.
*
         MOVE WP-BYTE(IX) TO WS-LOW
         MOVE LOW-VALUE   TO WS-HIGH
         MOVE ZERO        TO IH
         MOVE WS-HALF     TO IL
         PERFORM UNTIL IL < 16
             SUBTRACT 16 FROM IL
             ADD       1  TO  IH
         END-PERFORM
         ADD 1 TO IH
         ADD 1 TO IL
         MOVE HEX-CHAR(IH)
          TO  VALUE-DEST(((IX - 1) * 2) + 1:1)
         MOVE HEX-CHAR(IL)
          TO  VALUE-DEST(((IX - 1) * 2) + 2:1)
     END-PERFORM
     .
 END PROGRAM HEXMANIP.
 END PROGRAM ADDRDISP.
```

ADDRMAP

```
ADDRMAP  DFHMSD TYPE=MAP,                                              *
               MODE=INOUT,EXTATT=YES,                                  *
               CTRL=(FREEKB,FRSET),                                    *
               LANG=COBOL,                                             *
               TIOAPFX=YES
MAINMAP  DFHMDI SIZE=(24,8Ø),CURSLOC=YES
         DFHMDF POS=(1,23),                                            *
               LENGTH=35,ATTRB=(BRT,ASKIP),HILIGHT=REVERSE,            *
               INITIAL='                                   '
         DFHMDF POS=(1,59),                                            *
               LENGTH=1,                                               *
               ATTRB=(ASKIP)
         DFHMDF POS=(2,23),                                            *
               LENGTH=35,ATTRB=(BRT,ASKIP),HILIGHT=REVERSE,            *
               INITIAL='     Jerry''s Information Display     '
         DFHMDF POS=(2,59),                                            *
               LENGTH=1,                                               *
               ATTRB=(ASKIP)
         DFHMDF POS=(3,23),                                            *
               LENGTH=35,ATTRB=(BRT,ASKIP),HILIGHT=REVERSE,            *
               INITIAL='                                   '
         DFHMDF POS=(3,59),                                            *
               LENGTH=1,                                               *
               ATTRB=(ASKIP)
         DFHMDF POS=(5,15),                                            *
               LENGTH=7,                                               *
               INITIAL='Job ID:'
MJOBID   DFHMDF POS=(5,23),ATTRB=(ASKIP,BRT),                          *
               LENGTH=8
         DFHMDF POS=(5,32),                                            *
               LENGTH=9,                                               *
               INITIAL='Job Name:'
MJOBNM   DFHMDF POS=(5,42),ATTRB=(ASKIP,BRT),                          *
               LENGTH=8
         DFHMDF POS=(5,51),                                            *
               LENGTH=8,                                               *
               INITIAL='Release:'
MREL     DFHMDF POS=(5,6Ø),ATTRB=(ASKIP,BRT),                          *
               LENGTH=4
         DFHMDF POS=(7,1),OCCURS=5,                                    *
               LENGTH=15,                                              *
               INITIAL='ID IX A(ANCH)'
         DFHMDF POS=(8,1),OCCURS=5,                                    *
               LENGTH=15,                                              *
               INITIAL='- - ----'
*   NB  The UNPROT attribute is needed due to
*       incorrect 327Ø emulation on PCs.
ANCHLN   DFHMDF POS=(9,1),OCCURS=4Ø,                                   *
               ATTRB=UNPROT,                                           *
               LENGTH=15
MMSG     DFHMDF POS=(23,1Ø),                                           *
               LENGTH=6Ø,ATTRB=(ASKIP,DRK)
DETLMAP  DFHMDI SIZE=(24,8Ø)
         DFHMDF POS=(1,23),                                            *
               LENGTH=35,ATTRB=(BRT,ASKIP),HILIGHT=REVERSE,            *
               INITIAL='                                   '
         DFHMDF POS=(1,59),                                            *
               LENGTH=1,                                               *
               ATTRB=(ASKIP)
         DFHMDF POS=(2,23),                                            *
               LENGTH=35,ATTRB=(BRT,ASKIP),HILIGHT=REVERSE,            *
               INITIAL='       Jerry''s Anchor Display      '
         DFHMDF POS=(2,59),                                            *
               LENGTH=1,                                               *
               ATTRB=(ASKIP)
         DFHMDF POS=(3,23),                                            *
```

```
               LENGTH=35,ATTRB=(BRT,ASKIP),HILIGHT=REVERSE,           *
               INITIAL='                                    '
         DFHMDF POS=(3,59),                                           *
               LENGTH=1,                                              *
               ATTRB=(ASKIP)
         DFHMDF POS=(5,Ø9),                                           *
               LENGTH=7,                                              *
               INITIAL='Domain:'
DDOMID   DFHMDF POS=(5,17),ATTRB=(ASKIP,BRT),                         *
               LENGTH=2
         DFHMDF POS=(5,2Ø),                                           *
               LENGTH=7,                                              *
               INITIAL='Job ID:'
DJOBID   DFHMDF POS=(5,28),ATTRB=(ASKIP,BRT),                         *
               LENGTH=8
         DFHMDF POS=(5,37),                                           *
               LENGTH=9,                                              *
               INITIAL='Job Name:'
DJOBNM   DFHMDF POS=(5,47),ATTRB=(ASKIP,BRT),                         *
               LENGTH=8
         DFHMDF POS=(5,56),                                           *
               LENGTH=8,                                              *
               INITIAL='Release:'
DREL     DFHMDF POS=(5,65),ATTRB=(ASKIP,BRT),                         *
               LENGTH=4
ADDRLN   DFHMDF POS=(7,1),OCCURS=16,LENGTH=79
DMSG     DFHMDF POS=(23,1Ø),                                          *
               LENGTH=6Ø,ATTRB=(ASKIP,DRK)
         DFHMSD TYPE=FINAL
         END
```

CONCLUSION

The program illustrates two essential facts:

- COBOL can use pointers in the same way as other languages such as C.
- Addressing CICS control blocks is simple if you know the structures of some basic MVS and CICS areas.

However, you should always be aware that all control structures are release-dependent and may change in future versions.

Jerry Ozaniec
Circle Computer Group (UK)

Translating the timestamp to real time

I sometimes need to look in the SMF Records (Type 110). I’ve written two small programs for translating the timestamp to real time.

The input is the timestamp and the output is in the form of date/time.

You should note that, although the program works correctly, error messages will not be cleared.

SP80

```
***********************************************************
*                                                         *
* MAP FOR PROGRAM SYSTIME                                 *
*                                                         *
***********************************************************
*
         PRINT NOGEN
*                                       MAPSET
MPSETXX  DFHMSD TYPE=MAP,                                              -
               MODE=INOUT,                                             -
               LANG=ASM,                                               -
               STORAGE=AUTO,                                           -
               TIOAPFX=YES,                                            -
               CTRL=FREEKB,                                            -
               MAPATTS=(COLOR,HILIGHT),                                -
               DSATTS=(COLOR,HILIGHT)
*                                       MAP1
MAP1     DFHMDI SIZE=(24,8Ø),                                          -
               LINE=1,                                                 -
               COLUMN=1
*                                       MAP1 - HEADER
*                                       MAP1 - HEADER LINE 1
         DFHMDF POS=(1,1),                                             -
               LENGTH=12,                                              -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM),                                      -
               INITIAL='MAP1/MPSETXX'
         DFHMDF POS=(1,32),                                            -
               LENGTH=12,                                              -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM),                                      -
               INITIAL='            '
         DFHMDF POS=(1,65),                                            -
               LENGTH=15,                                              -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM),                                      -
               INITIAL='CHANGE:Ø2.Ø6.99'
*                                       MAP1 - HEADER LINE 3
         DFHMDF POS=(3,2Ø),                                            -
               LENGTH=36,                                              -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM),                                      -
               INITIAL='TRANSLATE  TIMESTAMP TO REAL TIME   '
*                                       MAP1 - INPUT  LINE 7
         DFHMDF POS=(7,1Ø),                                            -
               LENGTH=17,                                              -
               COLOR=GREEN,                                            -
               ATTRB=(PROT,NORM),                                      -
               INITIAL='TOD CLOCK FORMAT='
ETOD     DFHMDF POS=(7,28),                                            -
               LENGTH=16,                                              -
               JUSTIFY=(RIGHT,ZERO),                                   -
               COLOR=RED,                                              -
               ATTRB=(UNPROT,IC),                                      -
               HILIGHT=UNDERLINE
         DFHMDF POS=(7,45),LENGTH=1,ATTRB=(ASKIP,NORM)
*                                       MAP1 - OUTPUT LINE 11
         DFHMDF POS=(11,1Ø),                                           -
               LENGTH=17,                                              -
               COLOR=GREEN,                                            -
               ATTRB=(PROT,NORM,ASKIP),                                -
               INITIAL='TIME            ='
ESTD     DFHMDF POS=(11,28),                                           -
               LENGTH=2,                                               -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM,ASKIP)
         DFHMDF POS=(11,31),                                           -
               LENGTH=1,                                               -
               COLOR=GREEN,                                            -
               ATTRB=(PROT,NORM,ASKIP),                                -
               INITIAL=':'
EMIN     DFHMDF POS=(11,33),                                           -
               LENGTH=2,                                               -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM,ASKIP)
         DFHMDF POS=(11,36),                                           -
               LENGTH=1,                                               -
               COLOR=GREEN,                                            -
               ATTRB=(PROT,NORM,ASKIP),                                -
               INITIAL=':'
ESEC     DFHMDF POS=(11,38),                                           -
               LENGTH=2,                                               -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM,ASKIP)
         DFHMDF POS=(11,41),                                           -
               LENGTH=1,                                               -
               COLOR=GREEN,                                            -
               ATTRB=(PROT,NORM,ASKIP),                                -
               INITIAL='.'
EMIL     DFHMDF POS=(11,43),                                           -
               LENGTH=6,                                               -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM,ASKIP)
         DFHMDF POS=(11,5Ø),LENGTH=1,ATTRB=(ASKIP,NORM)
*                                       MAP1 - OUTPUT LINE 23
         DFHMDF POS=(13,1Ø),                                           -
               LENGTH=17,                                              -
               COLOR=GREEN,                                            -
               ATTRB=(PROT,NORM,ASKIP),                                -
               INITIAL='DATE            ='
ETT      DFHMDF POS=(13,28),                                           -
               LENGTH=2,                                               -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM,ASKIP)
         DFHMDF POS=(13,31),                                           -
               LENGTH=1,                                               -
               COLOR=GREEN,                                            -
               ATTRB=(PROT,NORM,ASKIP),                                -
               INITIAL='.'
EMM      DFHMDF POS=(13,33),                                           -
               LENGTH=2,                                               -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM,ASKIP)
         DFHMDF POS=(13,36),                                           -
               LENGTH=1,                                               -
               COLOR=GREEN,                                            -
               ATTRB=(PROT,NORM,ASKIP),                                -
               INITIAL='.'
EYY      DFHMDF POS=(13,38),                                           -
               LENGTH=4,                                               -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM,ASKIP)
         DFHMDF POS=(13,43),                                           -
               LENGTH=1,                                               -
               ATTRB=(ASKIP,NORM)
*                                       MAP1 - OUTPUT LINE 16
         DFHMDF POS=(16,5),                                            -
               LENGTH=35,                                              -
               COLOR=YELLOW,                                           -
               ATTRB=(PROT,NORM),                                      -
               INITIAL='PLEASE GIVE ME THE  TIMESTAMP   '
*                                       MAP1 - OUTPUT LINE 17
         DFHMDF POS=(17,5),                                            -
               LENGTH=35,                                              -
               COLOR=YELLOW,                                           -
               ATTRB=(PROT,NORM),                                      -
               INITIAL='DATE WILL BE  MM.DD.YYYY        '
*                                       MAP1 - ERROR  LINE 23
ERROR    DFHMDF POS=(23,1),                                            -
               LENGTH=79,                                              -
               COLOR=NEUTRAL,                                          -
               ATTRB=(PROT,NORM),                                      -
               JUSTIFY=(LEFT,BLANK)
*                                       MAP1 - PF-KEY  KEY 24
         DFHMDF POS=(24,1),                                            -
               LENGTH=6Ø,                                              -
               COLOR=BLUE,                                             -
               ATTRB=(PROT,NORM),                                      -
               INITIAL='DFUE=INPUT    F3=END '
         DFHMDF POS=(24,62),LENGTH=1,ATTRB=(ASKIP,NORM)
*                                       END  MAPSET
         DFHMSD TYPE=FINAL
         END
```

SP801

```
TITLE 'SYSTIME)'
*
*
         PRINT NOGEN
DFHEISTG DSECT
         COPY  MPSETXX
         COPY  DFHAID
         COPY  DFHBMSCA
NEWS     DS    CL79
ATTR     DS    CL2
         DS    ØF
WORK     DS    CL255
MIST     DS    CL1
ENDINF   DS    CL29
CURSOR   DS    H
R8S      DS    F
DWORT    DS    D
COMREG   DSECT
         DS    CL255
         EJECT
SYTIME   DFHEIENT CODEREG=3,DATAREG=13,EIBREG=6
SYTIME   AMODE 31
SYTIME   RMODE ANY
         USING COMREG,5
         L     5,DFHEICAP
         BAL   1Ø,CLEAR
         B     START
         DC    C'*SYTIME*V1.Ø*'
         DC    C'&SYSDATE*&SYSTIME*'
         DS    ØH
START    EQU   *
         CLI   COMREG,C'4'
         BNE   START1
         CLI   EIBAID,DFHPF3
         BE    *+14
         MVC   CURSOR,=H'5Ø8'
         B     SEND1
         MVC   ENDINF,=C'APPLICATION TIME WAS FINISHED'
         B     ENDE
START1   MVC   CURSOR,=H'5Ø8'
         CLC   EIBCALEN,=H'Ø'
         BH    KCHECK
         B     SEND1
SEND1    EQU   *                        1. PANEL OUT
         EXEC  CICS SEND MAP('MAP1')                                   -
               MAPSET('MPSETXX')                                       -
               ERASE                                                   -
               MAPONLY                                                 -
               FREEKB
         MVI   WORK,C'1'
         B     CHECK
SEND2    EQU   *                        1. PANEL WAS IN ERROR
         MVI   WORK,C'2'
         EXEC  CICS SEND MAP('MAP1')                                   -
               MAPSET('MPSETXX')                                       -
               FREEKB
         B     CHECK
SEND3    EQU   *                        1. PANEL WAS IN ERROR
         MVC   MAP1S(L'MAP1S),COMREG+2
         MVI   WORK,C'3'
         EXEC  CICS SEND MAP('MAP1')                                   -
               MAPSET('MPSETXX')                                       -
               FREEKB
         B     CHECK
SEND4    EQU   *                        RESULT OUT
         CLI   COMREG,C'4'
         EXEC  CICS SEND MAP('MAP1')                                   -
               MAPSET('MPSETXX')                                       -
               FREEKB
         B     CHECK
         EJECT
CHECK    EQU   *                        CALL TIME AGAIN
         EXEC  CICS SEND CONTROL CURSOR(CURSOR)
         MVC   WORK+1(1),MIST
         MVC   WORK+2(L'MAP1S),MAP1S
         EXEC  CICS RETURN                                             -
               TRANSID('TIME')                                         -
               COMMAREA(WORK)                                          -
               LENGTH(255)
```

```
KCHECK   EQU   *                        CHECK KEY
PCLEAR   CLI   EIBAID,DFHCLEAR          CLEAR-KEY ?
         BNE   PENTER                   NO
         B     SEND3
PENTER   CLI   EIBAID,DFHENTER          ENTER-KEY ?
         BNE   PKEY3                    NO
         B     DCHECK
PKEY3    CLI   EIBAID,DFHPF3            PF3-KEY ?
         BNE   PKEYXX                   NO
         MVC   ENDINF,=C'APPLICATION TIME WAS FINISHED'
         B     ENDE
PKEYXX   EQU   *
         MVI   ATTR,DFHRED
         MVC   NEWS(L'NEWS1),NEWS1
         B     FEHL
         EJECT
DCHECK   EQU   *                        CHECK PANEL
         EXEC  CICS HANDLE CONDITION MAPFAIL(NOMAP)
         EXEC  CICS RECEIVE MAP('MAP1')                                -
               MAPSET('MPSETXX')
         EXEC  CICS ASKTIME
         B     PRTOD
         EJECT
PRTOD    EQU   *                        CHECK TIME OF DAY
         TRT   ETODI,TABØ1
         BNZ   PRTODF
         B     TOD
         SPACE
PRTODF   EQU   *                        ERROR
         MVC   NEWS(L'NEWS3),NEWS3
         MVC   CURSOR,=H'5Ø8'
         MVI   ATTR,DFHRED
         B     FEHL
         SPACE
TOD      EQU   *                        WORK WITH TIME OF DAY
         XR    7,7
         XR    11,11
         LA    7,8
         LA    8,DWORT
         LA    9,ETODI
         LA    12,TABØ2
         SPACE
TODL     EQU   *                        WORK OF SLIP
         BAL   1Ø,TODU
         LA    8,1(Ø,8)
         LA    9,2(Ø,9)
         BCT   7,TODL
         B     TODM
         SPACE
TODU     EQU   *                        16 BYTE ===> 8 BYTE
```

```
         IC    11,Ø(Ø,9)
         IC    11,Ø(11,12)
         SLL   11,4
         STC   11,WORK
         IC    11,1(Ø,9)
         IC    11,Ø(11,12)
         STC   11,WORK+1
         MVN   WORK(1),WORK+1
         MVC   Ø(1,8),WORK
         BR    1Ø
         SPACE
TODM     EQU   *                        TOD MACRO
         STCKCONV STCKVAL=DWORT,                                       -
               CONVVAL=WORK,                                           -
               TIMETYPE=DEC,                                           -
               DATETYPE=DDMMYYYY
         LTR   15,Ø
         BZ    TODA
         MVC   NEWS(L'NEWS4),NEWS4
         MVC   CURSOR,=H'5Ø8'
         MVI   ATTR,DFHRED
         B     FEHL
         SPACE
TODA     EQU   *                        PROCESS RESULT
         LA    7,12                     LOAD SLIP
         LA    9,WORK
         LA    8,WORK
         LA    8,2Ø(Ø,8)
         ST    8,R8S
         XR    11,11
         SPACE
TODA1    EQU   *
         BAL   1Ø,TODA2
         LA    9,1(Ø,9)
         LA    8,2(Ø,8)
         BCT   7,TODA1
         B     TODA3
         SPACE
TODA2    EQU   *                        12 BYTE ===> 24 BYTE
         IC    11,Ø(Ø,9)
         SRL   11,4
         IC    11,Ø(11,12)
         STC   11,Ø(Ø,8)
         IC    11,Ø(Ø,9)
         SLL   11,28
         SRL   11,28
         IC    11,Ø(11,12)
         STC   11,1(Ø,8)
         BR    1Ø
         SPACE
```

```
TODA3    EQU   *                         PROCESS OUTPUT
         XC    MAP1O(MAP1E-MAP1O),MAP1O
         L     8,R8S
         MVC   ESTDO,Ø(8)                HOUR
         MVC   EMINO,2(8)                MINUTE
         MVC   ESECO,4(8)                SECOND
         MVC   EMILO,6(8)                MILLISECOND
         LA    8,16(Ø,8)
         MVC   ETTO,Ø(8)                 DAY
         MVC   EMMO,2(8)                 MONTH
         MVC   EYYO,4(8)                 YEAR
         B     SEND4
         EJECT
CLEAR    EQU   *                         CLEAR NEWS
         MVI   NEWS,X'4Ø'
         MVC   NEWS+1(L'NEWS-1),NEWS
         BR    1Ø
FEHL     EQU   *
         XC    MAP1O(MAP1E-MAP1O),MAP1O
         MVC   ERRORC(1),ATTR
         MVC   ERRORO(L'ERRORO),NEWS
         EXEC  CICS SEND CONTROL CURSOR(CURSOR)
         B     SEND2
         SPACE
NOMAP    EQU   *
         MVI   ATTR,DFHRED
         MVC   NEWS(L'NEWS2),NEWS2
         B     FEHL
         EJECT
ENDE     EQU   *
         EXEC  CICS SEND                                              -
               FROM(ENDINF)                                           -
               LENGTH(29)                                             -
               ERASE
         EXEC  CICS RETURN
         EJECT
         DS    ØF
NEWSØ    DC    C'IT WAS MISCELLANEOUS ERRORS BY INPUT !'
NEWS1    DC    C'MISTAKE PF-KEY'
NEWS2    DC    C'WHERE IS THE INPUT ?'
NEWS3    DC    C'WRONG CHARACTER IN TOD (Ø-F ONLY)'
NEWS4    DC    C'MACRO-ERROR STCKCONV'
         DS    ØF
TABØ1    EQU   *                         TABLE TO CHECK TOD-FIELD
         DC    129AL1(*+1-TABØ1)
         DC    X'ØØØØØØØØØØØØ'
         DC    58AL1(*-TABØ1)
         DC    X'ØØØØØØØØØØØØ'
         DC    41AL1(*-TABØ1)
         DC    X'ØØØØØØØØØØØØØØØØØØØØ'
         DC    6AL1(*-TABØ1)
         DS    ØF
TABØ2    EQU   *                        TABLE TO TRANSLATE TOD-FIELD
         DC    X'FØF1F2F3F4F5F6F7F8F9C1C2C3C4C5C6'
         DC    113AL1(*-*)
         DC    X'ØAØBØCØDØEØF'
         DC    58AL1(*-*)
         DC    X'ØAØBØCØDØEØF'
         DC    41AL1(*-*)
         DC    X'ØØØ1Ø2Ø3Ø4Ø5Ø6Ø7Ø8Ø9'
         DC    6AL1(*-*)
         LTORG
         END   SYTIME
```

Joachim Hoffmann
Systems Programmer (Germany)

ESDS DTB and emergency restart made simple

Dynamic Transaction Backout (DTB) backs out the modifications made to recoverable files when a transaction abends or following a User Syncpoint Rollback. Emergency restart backs out the modifications made to recoverable files following a CICS crash. Files are recovered to the state they were in at the beginning of a Logical Unit of Work (LUW).

THE PROBLEM

The VSAM ESDS files (and the BDAM one) do not support ‘delete’ and so you need to use the CICS exits XDBFERR and XRCFCER to ‘logically’ delete the record to be rolled back.

THE SOLUTION

The two exits presented here (XDBFERR and XRCFCER) are similar. They have been written to facilitate the logical delete and, using a simple table, allow you to determine:

- The VSAM ESDS files on which to perform this logical delete function.

- The hexadecimal configuration of the flag to be use to mark the record as logically deleted.
- The flag displacement in the record, or alternatively, if you wish, to extend the flag for the entire length of the record.

For the selected files, the exit programs signal the operation performed on the master terminal queue.

EXIT XDBFERR

```
*ASM XOPTS(NOEDF)
*ASM XOPTS(NOEPILOG)
         TITLE  'MACRO DEFINITIONS'
         MACRO                     MACRO HEADER
         CXID  &MEMBER,&R=         PROTYPE STATEMENT
         AGO   .PGNAME
.PGNAME  ANOP
.*
.*       THIS VARIABLE FOR TIME AND DATE STAMPING
         LCLC  &VMTMDT             TIME/DATE STAMP
         LCLC  &RELEASE            VERSION
.*
.*
        AIF  (T'&R NE 'O').SETR
&RELEASE     SETC 'Ø1Ø1'
        AGO  .DROP
.SETR   ANOP
&RELEASE     SETC '&R'
         SPACE 1
.DROP   ANOP
         PUSH PRINT
         PRINT GEN
***********************************************************************
         DC    C'*',C' '
         DC    C'PROGRAM NAME:'
         DC    CL8'&MEMBER'  NAME
         DC    C' ',C'*',C' '
         DC    C'PROGRAM VERSION:'
         DC    CL4'&RELEASE'
         DC    C' '
         DC    C'*',C' '
         SPACE
         DC    C'ASSEMBLY TIME(HH.MM):'
&VMTMDT  SETC  '&SYSTIME'
         DC    C'&VMTMDT'          ASSEMBLY TIME (HH.MM) AND
         DC    C' '
         DC    C'ASSEMBLY DATE(MM/DD/YY):'
```

```
&VMTMDT  SETC  '&SYSDATE'
         DC    C'&VMTMDT'          DATE (MM/DD/YY) SAME AS LISTING
************************************************************************
         POP   PRINT
         MEXIT
         MEND
*=================================================================
         MACRO
*
*
*
         CSNAME &NAME
         GBLC  &CSECT
         AIF ('&NAME' EQ '').NONAME
&CSECT   SETC  '&NAME'
         AGO   .SC
.NONAME  ANOP
&CSECT   SETC  '&SYSECT'
.SC      ANOP
         PUSH  PRINT
         PRINT GEN
*=========================================*
*                                         *
*                                         *
CSNAME   DC    CL8'&CSECT'
*                                         *
*                                         *
*=========================================*
         POP   PRINT
         MEND
         PRINT GEN
***************************************************
*                                                 *
* DYNAMIC TRANSACTION BACKOUT FILE ERROR EXIT     *
*                                                 *
***************************************************
*
* READ RECOVERY AND RESTART GUIDE
*
*
* REGISTER DETAIL:
* R1  ADDRESS OF EXIT PARAMETER-LIST DFHUEPAR
* R13 ADDRESS OF STD REGISTER SAVE AREA (ALSO IN FIELD UEPEPSA)
* R14 CONTAIN THE ADDRESS OF RETURN POINT
* R15 ENTRY ADDRESS OF EXIT PROGRAM JUST ENTERED
*
         DFHUEXIT TYPE=EP,ID=XDBFERR
*
         COPY  DFHDBRDS
*
```

```
         COPY  DFHFCTDS
*
DFHEISTG DSECT
SAVEREGS DS    5A
RC       DS    F
MSG      DS    ØCL64
CSECT    DS    CL8,CL1
FLNAME   DS    CL8,CL1
ERCODE   DS    CL2,CL1
ERCODE2  DS    CL2,CL1
MSGF1    DS    CL12
FUNCTION DS    CL2,CL1
ACTION   DS    CL26
*
RØ       EQU   Ø
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
EIBREG   EQU   R8
CODEREG  EQU   R9
DATAREG  EQU   R12
RUEPAR   EQU   R7
RDLR     EQU   R6
FCTDSBAR EQU   R1
RWKR1    EQU   R1
RWKR2    EQU   R2
RWKR1Ø   EQU   R1Ø
RWKR11   EQU   R11
RWKR14   EQU   R14
RWKR15   EQU   R15
*
         USING DFHUEPAR,RUEPAR
         USING DFHDBRDS,RDLR
*
XDBFERRP DFHEIENT CODEREG=(CODEREG),DATAREG=(DATAREG),EIBREG=(EIBREG)
XDBFERRP AMODE 31
XDBFERRP RMODE ANY
         B     ACXID
         CXID  XDBFERR,R=ØØØ1
ACXID    DS    ØH
         LR    RUEPAR,R1 SAVE PARAMETER LIST ADDRESS
*
         EXEC  CICS ADDRESS EIB(EIBREG)
*
         MVI   MSG,C' '
         MVC   MSG+1(L'MSG-1),MSG
         ICM   RWKR1,B'1111',UEPEXN ADDRESS OF EXIT NUMBER
         BZ    CONTINUE
         CLI   Ø(RWKR1),XDBFERR WRONG EXIT NUMBER
         BNE   CONTINUE
         ICM   RDLR,B'1111',UEPDLOGR LOAD ADDRESS OF DYNAMIC LOG RECORD
         BZ    CONTINUE
         MVC   CSECT,CSNAME
         MVC   FLNAME,DBRFID
         MVI   FLNAME+L'FLNAME,C'-'
         UNPK  ERCODE2(L'ERCODE2+1),DBRERRCD(2)
         TR    ERCODE2,TABEX-24Ø
         MVI   ERCODE2+L'ERCODE2,C'-'
         ICM   RWKR1,B'1111',UEPFCRSP
         BZ    NOERCODE
         UNPK  ERCODE(L'ERCODE+1),Ø(2,RWKR1)
         TR    ERCODE,TABEX-24Ø
NOERCODE DS    ØH
         MVI   ERCODE+L'ERCODE,C'-'
*
         ICM   FCTDSBAR,B'1111',UEPFCTE FCT ADDRESS
         BZ    CONTINUE
         TM    DBRERRCD,DBFEWA VSAM DELETE ERROR ?
         BZ    CONTINUE
* The record read from the restart dataset is a WRITE ADD, the record
* is read using GET UPDATE. For VSAM ESDS datasets, no delete function
* exists. Because DFHDBP does not attempt to delete the record that has
* been added, it is possible to MARK the existing record on file
* according to application-dependent logic (see table TABFILE1). The
* record version is addressed by UEPFDATA field. To reapply the record,
* specify return code UERCRTRY. UEPFLEN points to a field containing the
* length of the record.
         L     RWKR2,=A(TABFILE1)  USER DATA/SET TABLE
FILE1    DS    ØH
         CLC   Ø(8,RWKR2),FCTDSID DATA SET NAME
         BE    FLAGDS
         LA    RWKR2,L'TABFILE1(RWKR2)
         CLI   Ø(RWKR2),X'FF'
         BE    CONTINUE
         B     FILE1
FLAGDS   DS    ØH
         TM    11(RWKR2),X'8Ø' OC OR RESET RECORD FOR TOTAL LENGTH?
         BZ    OCR
         ICM   RWKR11,B'1111',UEPFLEN GET ADDRES OF RECORD LENGTH
         BZ    CONTINUE
         L     RWKR11,Ø(RWKR11) GET RECORD LENGTH
         ICM   RWKR1Ø,B'1111',UEPFDATA BEGIN OF DATA ADDRESS
         BZ    CONTINUE
         LR    RWKR14,RWKR1Ø   FROM ADDRESS IS ZERO
         SR    RWKR15,RWKR15   ZERO LENGTH.THE TO ADDRESS IS FILLED
*                              WITH PADDING BYTE
         ICM   RWKR15,B'1ØØØ',1Ø(RWKR2) MOVE MARK FIELD IN REGISTER AS
*                              PADDING BYTE
         MVCL  RWKR1Ø,RWKR14   PUT MARK FIELD INTO RECORD
         B     APPLY           RETURN WITH RETRY OPTION
OCR      DS    ØH
         LH    RWKR15,8(RWKR2) FLAG DISPLACEMENT IN THE RECORD
         ICM   RWKR11,B'1111',UEPFLEN GET LENGTH OF RECORD
         BZ    CONTINUE
         CR    RWKR15,RWKR11
         BH    CONTINUE
         L     RWKR1Ø,UEPFDATA BEGIN OF DATA ADDRESS
         AR    RWKR15,RWKR1Ø
         MVC   Ø(1,RWKR15),1Ø(RWKR2) MARK RECORD AS DELETED
APPLY    DS    ØH
         LA    RWKR15,UERCRTRY
         MVC   ACTION,=CL26'Reapply the record'
         B     RETURN
IGNORE   DS    ØH
         LA    RWKR15,UERCBYP
         MVC   ACTION,=CL26'Ignore - backout failure'
         B     RETURN
CONTINUE DS    ØH
         LA    RWKR15,UERCNORM
         MVC   ACTION,=CL26'Continue - backout failure'
RETURN   DS    ØH
         CLC   CSECT,CSNAME
         BNE   RETURNF
         ST    RWKR15,RC
         UNPK  FUNCTION(L'FUNCTION+1),RC+L'RC-1(2)
         TR    FUNCTION,TABEX-24Ø
         MVC   MSGF1,=CL12'Return code:'
*
         EXEC  CICS WRITEQ TD QUEUE('CSMT') FROM(MSG)                  *
               LENGTH(=Y(L'MSG)) NOHANDLE
*
         L     RWKR15,RC
RETURNF  DS    ØH
         DFHEIRET RCREG=15
         LTORG
         SPACE
TABFILE1 DS    ØXL12
```

```
* DATA SET NAME
* DISPLACEMENT MARK FIELD
* MARK FIELD
* MARK BYTE: BYTE Ø=OFF OR MARK FIELD FOR LENGTH 1
*            BYTE Ø=ON  MOVE MARK FIELD FOR RECORD LENGTH
*
* THE FOLLOWING TABLE ELEMENT IS USED FOR A VSAM FILE ESDS
* WITH FCT NAME "filename", MARKED FOR LOGICAL DELETE AT
* DISPLACEMENT Ø WITH X'FF'
         DC    CL8'filename',H'Ø',X'FF',X'ØØ'
*
* THE FOLLOWING TABLE ELEMENT IS USED FOR A VSAM FILE ESDS
* WITH FCT NAME "filenam2", MARKED FOR LOGICAL DELETE AT
* DISPLACEMENT 1ØØ WITH X'ØØ'
         DC    CL8'filenam2',H'1ØØ',X'ØØ',X'ØØ'
*
* THE FOLLOWING TABLE ELEMENT IS USED FOR A VSAM FILE ESDS
* WITH FCT NAME "filenam3". THE RECORD IS FILLED WITH "*"
*
         DC    CL8'filenam3',H'Ø',C'*',X'8Ø'
*
* END OF TABLE
         DC    X'FF'
TABEX    DC    C'Ø123456789ABCDEF'
         CSNAME
         END   XDBFERRP
```

EXIT XRCFCER

```
         TITLE  'MACRO DEFINITIONS'
         MACRO                       MACRO HEADER
         CXID  &MEMBER,&R=           PROTYPE STATEMENT
         AGO   .PGNAME
.PGNAME  ANOP
.*
.*       THIS VARIABLE FOR TIME AND DATE STAMPING
         LCLC  &VMTMDT               TIME/DATE STAMP
         LCLC  &RELEASE              VERSION
.*
.*
        AIF  (T'&R NE 'O').SETR
&RELEASE     SETC 'Ø1Ø1'
        AGO  .DROP
.SETR   ANOP
&RELEASE     SETC '&R'
         SPACE 1
.DROP   ANOP
         PUSH PRINT
         PRINT GEN
```

```
************************************************************************
         DC    C'*',C' '
         DC    C'PROGRAM NAME:'
         DC    CL8'&MEMBER'  NAME
         DC    C' ',C'*',C' '
         DC    C'PROGRAM VERSION:'
         DC    CL4'&RELEASE'
         DC    C' '
         DC    C'*',C' '
         SPACE
         DC    C'ASSEMBLY TIME(HH.MM):'
&VMTMDT  SETC  '&SYSTIME'
         DC    C'&VMTMDT'          ASSEMBLY TIME (HH.MM) AND
         DC    C' '
         DC    C'ASSEMBLY DATE(MM/DD/YY):'
&VMTMDT  SETC  '&SYSDATE'
         DC    C'&VMTMDT'          DATE (MM/DD/YY) SAME AS LISTING
************************************************************************
         POP   PRINT
         MEXIT
         MEND
*=================================================================
         MACRO
*
*
         CSNAME &NAME
         GBLC  &CSECT
         AIF ('&NAME' EQ '').NONAME
&CSECT   SETC  '&NAME'
         AGO   .SC
.NONAME  ANOP
&CSECT   SETC  '&SYSECT'
.SC      ANOP
         PUSH  PRINT
         PRINT GEN
*=========================================*
*                                         *
*                                         *
CSNAME   DC    CL8'&CSECT'
*                                         *
*                                         *
*=========================================*
         POP   PRINT
         MEND
*ASM XOPTS(NOEDF)
*ASM XOPTS(NOEPILOG)
         PRINT GEN
* XRCFCER FILE ERROR EXIT FOR RESOURCE BACKOUT OR RECOVERY AT
* EMERGENCY RESTART (DFHFCBP - FILE CONTROL BACKOUT)
************************************************************************
```

```
         TITLE 'CUSTOMER INFORMATION CONTROL GLOBAL   U S E R  E X I T'
***********************************************************************
* * * * *              U S E R  E X I T                       * * * * *
* * * * *                FILE ERROR                           * * * * *
***********************************************************************
* SPECIFIC INFORMATION ON USER EXIT:
* IF THE USER WANTS FWA VERSION TO BE REAPPLIED, RETURN SHOULD BE MADE
* THROUGH A B Ø(,R14).
* IF THE USER DOES NOT WANT THIS BUT WOULD RATHER BYPASS THE
* OPERATION, RETURN SHOULD BE MADE THROUGH A RETURN CODE 4
*
*
* REGISTER DETAIL:
* R1  ADDRESS OF EXIT PARAMETER-LIST DFHUEPAR
* R13 ADDRESS OF STD REGISTER SAVE AREA (ALSO IN FIELD UEPEPSA)
* R14 CONTAIN THE ADDRESS OF RETURN POINT
* R15 ENTRY ADDRESS OF EXIT PROGRAM JUST ENTERED
*
         DFHUEXIT TYPE=EP,ID=XRCFCER
*
         COPY  DFHFBODS
*
         COPY  DFHFCTDS
*
DFHEISTG DSECT
RC       DS    F
MSG      DS    ØCL64
CSECT    DS    CL8,CL1
FLNAME   DS    CL8,CL1
ERCODE   DS    CL2,CL1
ERCODE2  DS    CL2,CL1
MSGF1    DS    CL12
FUNCTION DS    CL2,CL1
ACTION   DS    CL26
*
RØ       EQU   Ø
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
R11      EQU   11
R12      EQU   12
R13      EQU   13
```

```
R14      EQU   14
R15      EQU   15
EIBREG   EQU   R8
CODEREG  EQU   R9
DATAREG  EQU   R12
RUEPAR   EQU   R7
RDLR     EQU   R6
FCTDSBAR EQU   R1
RWKR1    EQU   R1
RWKR2    EQU   R2
RWKR1Ø   EQU   R1Ø
RWKR11   EQU   R11
RWKR14   EQU   R14
RWKR15   EQU   R15
*
         USING DFHUEPAR,RUEPAR
         USING DFHFBODS,RDLR
*
XRCFCERP DFHEIENT CODEREG=(CODEREG),DATAREG=(DATAREG),EIBREG=(EIBREG)
XRCFCERP AMODE ANY
XRCFCERP RMODE ANY
         B     ACXID
         CXID  XRCFCER,R=ØØØ1
ACXID    DS    ØH
         LR    RUEPAR,R1 SAVE PARAMETER LIST ADDRESS
*
         EXEC  CICS ADDRESS EIB(EIBREG)
*
         MVI   MSG,C' '
         MVC   MSG+1(L'MSG-1),MSG
         ICM   FCTDSBAR,B'1111',UEPFCTE
         BZ    CONTINUE
         MVC   CSECT,CSNAME
         MVC   FLNAME,FCTDSID FILE IDENT
         MVI   FLNAME+L'FLNAME,C'-'
         ICM   RWKR1,B'1111',UEPERR
         BZ    NOERCOD
         UNPK  ERCODE2(L'ERCODE2+1),Ø(2,RWKR1)
         TR    ERCODE2,TABEX-24Ø
         MVI   ERCODE2+L'ERCODE2,C'-'
NOERCOD  DS    ØH
         ICM   RWKR1,B'1111',UEPFCRSP
         BZ    NOERCODE
         UNPK  ERCODE(L'ERCODE+1),Ø(2,RWKR1)
         TR    ERCODE,TABEX-24Ø
         MVI   ERCODE+L'ERCODE,C'-'
NOERCODE DS    ØH
         MVI   ERCODE+L'ERCODE,C'-'
         L     RWKR1,UEPERR
         TM    Ø(RWKR1),TBFEWA VSAM DELETE ERROR ?
         BZ    CONTINUE
*
* The record read from the restart dataset is a WRITE ADD, the record
* is read using A GET UPDATE. For VSAM ESDS datasets, no delete
* function exists. Because DFHFCBP does not attempt to delete the
* record added, it is possible to MARK the existing record on
* the file according to application-dependent logic (see table
* iTABFILE1). The record version is addressed by UEPFDATA field. To
* reapply the record, specify return code UERCRTRY. UEPFLEN points to
* a field containing the length of the record.
*
         L     RWKR2,=A(TABFILE1)  USER DATA/SET TABLE
         L     FCTDSBAR,UEPFCTE
FILE1    DS    ØH
         CLC   Ø(8,RWKR2),FCTDSID DATA SET NAME
         BE    FLAGDS
         LA    RWKR2,L'TABFILE1(RWKR2)
         CLI   Ø(RWKR2),X'FF'
         BE    CONTINUE
         B     FILE1
FLAGDS   DS    ØH
         TM    11(RWKR2),X'8Ø' OC OR RESET RECORD FOR TOTAL LENGTH?
         BZ    OCR
         ICM   RWKR11,B'1111',UEPFLEN GET ADDRES OF RECORD LENGTH
         BZ    CONTINUE
         L     RWKR11,Ø(RWKR11) GET RECORD LENGTH
         ICM   RWKR1Ø,B'1111',UEPFDATA BEGIN OF DATA ADDRESS
         BZ    CONTINUE
         LR    RWKR14,RWKR1Ø   FROM ADDRESS IS ZERO
         SR    RWKR15,RWKR15   ZERO LENGTH.THE TO ADDRESS IS FILLED
         ICM   RWKR15,B'1ØØØ',1Ø(RWKR2) MOVE MARK FIELD IN REGISTER AS
         MVCL  RWKR1Ø,RWKR14   PUT MARK FIELD INTO RECORD
         B     APPLY           RETURN WITH RETRY OPTION
OCR      DS    ØH
         LH    RWKR15,8(RWKR2) FLAG DISPLACEMENT IN THE RECORD
         ICM   RWKR11,B'1111',UEPFLEN GET LENGTH OF RECORD
         BZ    CONTINUE
         CR    RWKR15,RWKR11
         BH    CONTINUE
         L     RWKR1Ø,UEPFDATA BEGIN OF DATA ADDRESS
         AR    RWKR15,RWKR1Ø
         MVC   Ø(1,RWKR15),1Ø(RWKR2) MARK RECORD AS DELETED
APPLY    DS    ØH
         LA    RWKR15,UERCRTRY
         MVC   ACTION,=CL26'Reapply the record'
         B     RETURN
IGNORE   DS    ØH
         LA    RWKR15,UERCBYP
         MVC   ACTION,=CL26'Ignore - backout failure'
         B     RETURN
```

```
CONTINUE DS    ØH
         LA    RWKR15,UERCNORM
         MVC   ACTION,=CL26'Continue - backout failure'
RETURN   DS    ØH
         CLC   CSECT,CSNAME
         BNE   RETURNF
         ST    RWKR15,RC
         UNPK  FUNCTION(L'FUNCTION+1),RC+L'RC-1(2)
         TR    FUNCTION,TABEX-24Ø
         MVC   MSGF1,=CL12'Return code:'
*
         EXEC  CICS WRITEQ TD QUEUE('CSMT') FROM(MSG)                  *
               LENGTH(=Y(L'MSG)) NOHANDLE
*
         L     RWKR15,RC
RETURNF  DS    ØH
         DFHEIRET RCREG=15
         LTORG
TABFILE1 DS    ØXL12
* DATA SET NAME
* DISPLACEMENT MARK FIELD
* MARK FIELD
* MARK BYTE: BYTE Ø=OFF OR MARK FIELD FOR LENGTH 1
*            BYTE Ø=ON  MOVE MARK FIELD FOR RECORD LENGTH
*
* THE FOLLOWING TABLE ELEMENT IS USED FOR A VSAM FILE ESDS
* WITH FCT NAME "filename", MARKED FOR LOGICAL DELETE AT
* DISPLACEMENT Ø WITH X'FF'
         DC    CL8'filename',H'Ø',X'FF',X'ØØ'
*
* THE FOLLOWING TABLE ELEMENT IS USED FOR A VSAM FILE ESDS
* WITH FCT NAME "filenam2", MARKED FOR LOGICAL DELETE AT
* DISPLACEMENT 1ØØ WITH X'ØØ'
         DC    CL8'filenam2',H'1ØØ',X'ØØ',X'ØØ'
*
* THE FOLLOWING TABLE ELEMENT IS USED FOR A VSAM FILE ESDS
* WITH FCT NAME "filenam3". THE RECORD IS FILLED WITH "*"
*
         DC    CL8'filenam3',H'Ø',C'*',X'8Ø'
*
* END OF TABLE
         DC    X'FF'
TABEX    DC    C'Ø123456789ABCDEF'
         CSNAME
         END   XRCFCERP
```

Giuseppe Rallo
Senior Technical Analyst
Consulent (Italy)

CICS news

Forté has announced the availability of its Transaction Adapter for OS/390, part of the Fusion EAI suite. This is the first EAI offering that supports access to CICS/ESA and IMS subsystems directly from components running on System/390, eliminating the need for intermediate middleware and gateways and allowing users to take advantage of the mainframe's superior performance for EAI applications.

For further information contact:
Forté Software, 1800 Harrison Street, 24th Floor, Oakland, CA 94612, USA.
Tel: (510) 869 3400.
Forté Software, St James' House, Oldbury, Bracknell, Berkshire, RG12 8SA, UK.
Tel: (01344) 482100.
URL: http://www.forte.com.

* * *

Landmark Systems has announced Version 1.1 of The Monitor (TMON) for MQSeries, including better integration with TMON for CICS/ESA. This allows information and views from both mainframe and distributed applications to be presented together.

Installation and configuration are now easier, while the MVS Queue Manager Security Management feature displays MQSeries security switches and provides the ability to alter user-ids, time-out intervals, and refresh security.

Among the improvements is an Automatic Dead Letter Queue Processor with a rules-driven design. This reduces manual intervention and enables critical messages to be handled differently from non-critical messages.

There is also improved integration with MQControl and Message Management Facility Agents for management of MQSeries on distributed platforms while adding message management for all MQControl-supported platforms.

The software integrates with all other Landmark products, and provides the same look and feel across the rest of the range.

For further information contact:
Landmark Systems, 12700 Sunrise Valley Drive, Reston, Virginia 20191-5804, USA.
Tel: (703) 464 1300.
URL: http://www.landmark.com.

* * *

CICS '99, Xephon's leading independent forum for CICS performance and management specialists, takes place at the Millennium Chelsea Hotel, London on 8-9 December. This is the sixth annual conference that Xephon has run for CICS users; this year's multi-stream event focuses on e-commerce and Web enablement, as well as core mainframe performance issues.

CICS Update subscribers can attend *CICS '99* at the special rate of £555 plus £64.75 VAT. For further details, phone 01635 33823 or visit Xephon's Events page at www.xephon.com/zevent.html.

* * *