146 CICS update

January 1998

In this issue

Published by

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

North American office

Xephon
1301 West Highway 407, Suite 201-450
Lewisville, TX 75067, USA
Telephone: 940 455 7050

Australian office

Xephon/RSM
PO Box 6258, Halifax Street
Adelaide, SA 5000
Australia
Telephone: 08 223 1391

Contributions

If you have anything original to say about CICS, or any interesting experience to recount, why not spend an hour or two putting it on paper? The article need not be very long – two or three paragraphs could be sufficient. Not only will you be actively helping the free exchange of information, which benefits all CICS users, but you will also gain professional recognition for your expertise, and the expertise of your colleagues, as well as some material reward in the form of a publication fee – we pay at the rate of £170 ($250) per 1000 words for all original material published in *CICS Update*. If you would like to know a bit more before starting on an article, write to us at one of the above addresses, and we'll send you full details, without any obligation on your part.

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 £165.00 in the UK; $250.00 in the USA and Canada; £171.00 in Europe; £177.00 in Australasia and Japan; and £175.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1994 issue, are available separately to subscribers for £14.50 ($21.50) each including postage.

CICS Update on-line

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

Printed in England.

Warm keypoint indicator in CICS Version 5

CICS Versions 2, 3, and 4 provided a warm keypoint indicator (WMKP) as part of normal shutdown processing. This indicator is written by the warm keypoint program (DFHWKP) when all system activity has been successfully quiesed. In CICS Version 2 the WMKP is stored in the restart dataset. In CICS Version 3 and 4 this has been moved to the global catalog. In CICS Version 5 (CICS Transaction Server Version 1) the management of the type of CICS start has changed radically. The emergency/warm/cold/initial start indicator is now handled by the Recovery Manager Domain and the method of accessing this information is not provided by IBM. The record with the original WMKP has been discarded from the catalog.

Starting with CICS Version 3, we used this WMKP to automate our way to start CICS cold. A user program, the first step in our CICS start-up procedures (see Figure 1), always checks the WMKP. This idea originated from Marc Cop (Belgium) who described the problem in *Automating CICS cold start, CICS Update* Issue 98, January 1994, on pages 3-7, and he provided a sample program. Francois Le Maner (France) added some notes to this theme in *Automating CICS cold start revisited, CICS Update* Issue 108, November 1994, page 36.

I would like to continue this automation in CICS TS and therefore decided to write the 'old' WMKP back to the catalog. I enable the following XMEOUT exit during shutdown phase with a PLTSD program. The exit intercepts the new shutdown message DFHRM0204 which indicates that there are no in-doubt, commit-failed, or backout-failed UOWs. In this case, the exit sets the WMKP to true and rewrites the record to the catalog. Note that the exit expects there to be a record with such a key in the catalog (look in the program for the constant GCDKEY). You must ensure with your pre-DFHSIP program that this record is present and the WMKP will be reset to false at every CICS start.

It should be noted that IBM has warned against performing CICS dataset updates, especially during shutdown. Nevertheless, this is done by the exit. There have been no problems in our test and

production CICS environment to date, with the advantage that our pre-DFHSIP program is compatible across all CICS versions. If you want, you can write the WMKP to a separate dataset.

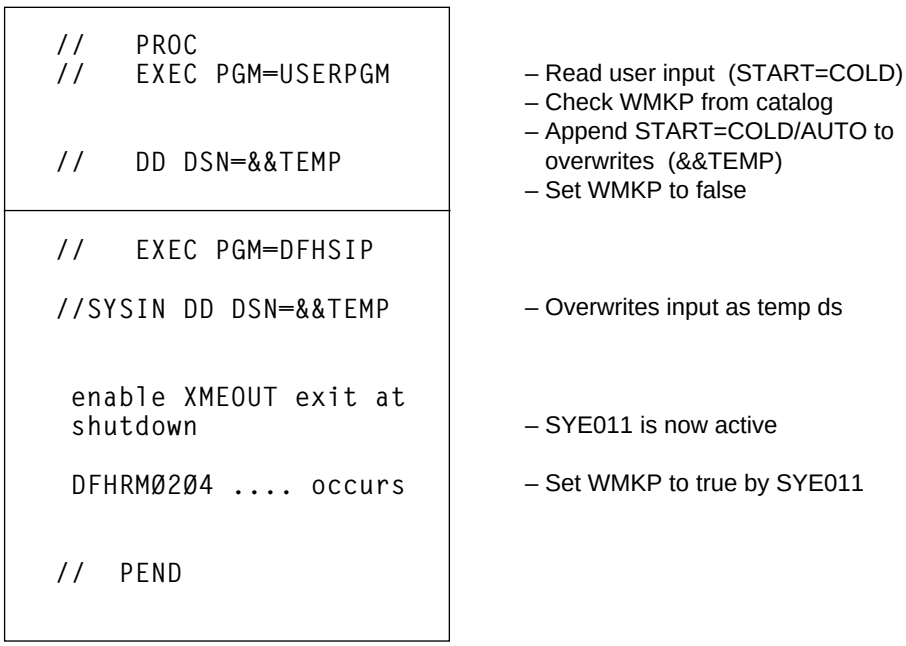

Figure 1: CICS TS Start procedure

EXIT PROGRAM

```
         TITLE 'SYE011 - XMEOUT GLOBAL USER EXIT WARM KEYPOINT'
 PRINT ON,NOGEN,NODATA
*********************************************************************
* PROGRAM-NAME: SYE011     - GLOBAL USER EXIT -                     *
* PURPOSE: USER EXIT TO BE PLACED AT THE XMEOUT EXIT ON CICS        *
*          TRANSACTION SERVER 1.1                                   *
*          IF MESSAGE DFHRM0204 OCCURS IN THE SHUTDOWN PHASE        *
*          THE OLD RESTART RECORD WILL BE WRITTEN ON GCD .          *
*                                                                   *
* INPUT:         GLOBAL CATALOG DATASET (GCD)                       *
* OUTPUT:        GLOBAL CATALOG DATASET (GCD)                       *
*********************************************************************
*
         DFHUEXIT TYPE=EP,ID=XMEOUT
```

```
TIOT     DSECT
         IEFTIOT1
*
*-------------------------------------------------------------------*
* REGISTER EQUATES                                                  *
*-------------------------------------------------------------------*
*
RØ       EQU   Ø        SYSTEM
R1       EQU   1        SYSTEM
R2       EQU   2        TIOT
R3       EQU   3        EXIT PARAMETER LIST
R4       EQU   4        DOMAIN IF CICS MESSAGE
R5       EQU   5        ADDRESS OF PREVIOUS RETURN CODE
R6       EQU   6        UNUSED
R7       EQU   7        UNUSED
R8       EQU   8        UNUSED
R9       EQU   9        UNUSED
R1Ø      EQU   1Ø       UNUSED
R11      EQU   11       BASE REGISTER FOR CSECT SYEØ11
R12      EQU   12       UNUSED
R13      EQU   13       SYSTEM
R14      EQU   14       SYSTEM
R15      EQU   15       SYSTEM
*-------------------------------------------------------------------*
* START OF MAINLINE                                                 *
*-------------------------------------------------------------------*
SYEØ11   CSECT                    /*
SYEØ11   AMODE 31                 /*
SYEØ11   RMODE ANY                /*
         SAVE  (14,12)            /* SAVE REGISTERS
         LR    R11,R15
         USING SYEØ11,R11
         ST    R13,SAVEAREA+4     /* ADDRESS OF HSA IN SAVE AREA
         LR    R1Ø,R13            /* NOTE OF ADDRESS OF HSA
         LA    R13,SAVEAREA       /* LOAD ADDRESS OF CURRENT SA
         ST    R13,8(R1Ø)         /* ADDRESS OF LSA IN HSA
         LR    R3,R1              /* PARM ADDRESS
         USING DFHUEPAR,R3        /* ADDRESS USER EXIT PARAMETER LIST
         B     LØØØ               /*  BRANCH AROUND  EYECATCH
         DC    CL2Ø'SYEØ11 VERSION Ø1   '
         DC    CL2Ø'&SYSDATE &SYSTIME   '
         DC    CL2Ø'(C) XEPHON   1997   '
         DC    CL2Ø'XMEOUT FOR RESTART  '
         DC    CL2Ø'SET WARM KEYPOINT   '
         DC    CL2Ø'INDICATOR           '
SAVEAREA DC    18A(Ø)             /* CURRENT SAVE AREA
         CNOP  Ø,4
LØØØ     EQU   *                  /*
         L     R4,UEPMDOM         DOMAIN IF CICS MESSAGE
         L     R5,UEPCRCA         GET RC FIELD ADDRESS
```

```
*--------------------------------------------------------------------*
* IGNORE ALL BUT RMØ2Ø4                                              *
*--------------------------------------------------------------------*
         CLC   Ø(2,4),=C'RM'      IF NOT RM
         BNE   RCNORMAL             /* DO NOTHING
         L     4,UEPMNUM          MESSAGE NUMBER
         CLC   Ø(4,4),=F'Ø2Ø4'    DFHRMØ2Ø4
         BNE   RCNORMAL           ... NO - EXIT
PROCESS  DS    ØH
*--------------------------------------------------------------------*
*        OPEN DFHGCD                                                 *
*--------------------------------------------------------------------*
         OPEN  (DFHGCD,OUTPUT),MODE=31
         LTR   R15,R15
         BZ    LØ1Ø
         WTO   'SYEØ11Ø1E OPEN ERROR WARM KEYPOINT INDICATOR'
         B     LØ8Ø
LØ1Ø     DS    ØH
*--------------------------------------------------------------------*
*        READ RECORD                                                 *
*--------------------------------------------------------------------*
LØ2Ø     DS    ØH
         GET   RPL=GCDRPL
         LTR   R15,R15
         BZ    LØ3Ø
         WTO   'SYEØ11Ø2E READ ERROR WARM KEYPOINT INDICATOR'
         B     LØ8Ø
LØ3Ø     DS    ØH
*--------------------------------------------------------------------*
* >>>>>> SET WARM KEYPOINT INDICATOR TO TRUE <<<<<<<<<<<<<<<<<<<<<< *
*--------------------------------------------------------------------*
         OI    KPLRSDIN,KPLWMKP
*--------------------------------------------------------------------*
*        WRITE RECORD                                                *
*--------------------------------------------------------------------*
LØ4Ø     DS    ØH
         PUT   RPL=GCDRPL
         LTR   R15,R15
         BZ    LØ5Ø
         WTO   'SYEØ11Ø3E WRITE ERROR WARM KEYPOINT INDICATOR'
         B     LØ8Ø
LØ5Ø     DS    ØH
*--------------------------------------------------------------------*
*        CLOSE DFHGCD                                                *
*--------------------------------------------------------------------*
LØ6Ø     DS    ØH
         CLOSE (DFHGCD),MODE=31
         LTR   R15,R15
         BZ    LØ7Ø
         WTO   'SYEØ11Ø4E CLOSE ERROR WARM KEYPOINT INDICATOR'
```

```
         B     LØ8Ø
LØ7Ø     DS    ØH
         EXTRACT TIOTADDR,FIELDS=(TIOT),MF=(E,DO_EXTRACT)
         L     R2,TIOTADDR        LOAD ADDRESS OF TIOT
         USING TIOT,R2            MAKE CVT ADRESSABLE
         MVC   JOBNAME,TIOCNJOB   MOVE JOBNAME TO MESSAGE AREA
         WTO   TEXT=GOODMSG
         B     RCNORMAL
*-------------------------------------------------------------------*
*   RCNORMAL WILL SET THE RETURN CODE TO UERCNORM                   *
*-------------------------------------------------------------------*
LØ8Ø     DS    ØH
RCNORMAL DS    ØH
         XR    R15,R15              /* Ø R15
         ICM   R15,B'ØØ11',Ø(R5)    /* SET R15 TO PREVIOUS CURRENT RC
         B     L9ØØ                 /* EXIT
*-------------------------------------------------------------------*
*   RCBYPASS WILL SET THE RETURN CODE TO UERCBYP                    *
*   RCBYPASS NOT EXECUTED - DFHRMØ2Ø4 NOT SUPPRESSED                *
*-------------------------------------------------------------------*
RCBYPASS DS    ØH
         MVI   1(R5),UERCBYP        /* INIT RC FIELD
         LA    R15,UERCBYP          /* INIT R15 RC
         B     L9ØØ                 /* EXIT
*-------------------------------------------------------------------*
* RESTORE REGISTERS, SET RETURN CODE, AND RETURN TO USER            *
* EXIT HANDLER.                                                     *
*-------------------------------------------------------------------*
L9ØØ     DS    ØH
         L     R13,UEPEPSA
         RETURN (14,12),RC=(15)
*-------------------------------------------------------------------*
* DEFINES FOR THE EXIT                                              *
*-------------------------------------------------------------------*
TIOTADDR DS    A              ADDRESS OF TIOT AFTER EXTRACT MACRO
DO_EXTRACT EXTRACT ,'S',FIELDS=(TIOT),MF=L
DFHGCD   ACB   DDNAME=DFHGCD,MACRF=(KEY,DIR,OUT),AM=VSAM,RMODE31=ALL
GCDRPL   RPL   ACB=DFHGCD,RECLEN=4Ø89,AREA=KSREC,ARG=GCDKEY,           X
               AREALEN=4Ø89,OPTCD=(KEY,DIR,UPD,MVE),AM=VSAM
GCDKEY   DC X'ØØØØØØ1ØC3E3D34ØØØØ4Ø4ØC3E3D34Ø4Ø4Ø4Ø4Ø4Ø4Ø4Ø4Ø4Ø4Ø4Ø4Ø4Ø'
GOODMSG  DC    H'44'
         DC    CL1Ø'SYEØ111Ø5 '
JOBNAME  DC    CL8'        '
         DC    CL26' Warm keypoint successful.'
         LTORG
KSREC      DS ØCL4Ø89
SCHLUESSEL DS  CL28
KPCTLDS    DS ØCL2Ø             RSD CONTROL RECORD DSECT
KPCTL      DC  CL4'CTL '        IDENTIFIER
KPLVOL     DC  CL6' '           JOURNAL TAPE VOL.SER
```

```
KPLRSDIN   DC  B'0'             RSD INDICATOR BYTE
KPLTAPE    EQU B'10000000'      SYSTEM LOG IS ON TAPE
KPLDISK    EQU B'01000000'      SYSTEM LOG IS ON DISK
KPLWMKP    EQU B'00100000'      WARM KEYPOINT WAS TAKEN
           DC  B'0'             RESERVED
KPLTIME    DC  CL8' '           TIME DATE STAMP
KPLLN      EQU *-KPCTLDS        LENGTH OF CTL RECORD
           DS  CL4009
         END SYE011
```

Erhard Woerner
Systems Programmer
Deutsche Bank AG (Germany)

Setting the VSE return code

DPPLTI

The DPPLTI program sets the VSE return code during CICS start-up and normal shutdown, so that conditional JCL can be used to restart it automatically if the CICS system terminated abnormally. It also determines whether DTSANALS needs to be run and, if it does, submits a job to perform a RECOVER function. This program performs two distinct functions:

1 Sets the VSE return code (ie $RC/$MRC) to a 1 when CICS is first started, then, at normal shutdown (ie CEMT PER SHUT and CEMT PER SHUT IMM), sets the VSE return code to a 2. Then, using conditional JCL, the $MRC return code is checked. If it's not a 2, assuming that CICS has terminated abnormally, it will restart the affected CICS automatically, using conditional JCL. The capability to automatically restart a terminated CICS is critical to those installations that run unattended for part of the time but need their CICS systems up and running all the time.

2 Determines whether the DTSFILE needs to be recovered because of an improper shutdown the last time CICS/ICCF was activated. If it does, a 'DTSANALS RECOVER' job is submitted to VSE/

POWER, along with other JCL, so that DTSANALS can determine whether a recovery is needed against the DTSFILE, and if necessary, perform the function. The capability to automatically submit a 'DTSANALS RECOVER' job is critical to those installations that run unattended part of the time but require their CICS/ICCF up and running all of the time. This facility also eliminates the need for an operator to monitor the console for message K008I, thus removing the need for the operator to release a 'DTSANALS RECOVER' job and a '/CONN DTSFILE' command on their own.

INSTALLATION

To carry out the installation:

1 Assemble the provided subroutines, DPCKJP and DPRTNC, as you would any normal Assembler subroutines. The subroutines must be catalogued as .OBJ member types and the library/ sublibrary they are catalogued into must be accessible (ie LIBDEFed via a 'search' statement) when assembling and link editing this program. Expect to receive a $RC of 0 when assembling these subroutines.

2 Assemble this program as you would any normal Assembler CICS command-level program, after first reading these notes and those contained within the code. Expect to receive a $RC of 4 when assembling the program, because of the use of 'EXEC CICS ADDRESS CSA(....)'.

Note: if you are not using the high-level Assembler you must remove or comment out the AMODE/RMODE statements, indicated by three hash signs (ie ###) in columns 69-71, otherwise an assembly error will occur.

3 Add a PPT/RDO entry for this program. Make sure you specify 'Assembler'. You may use any of the other PPT operands at your discretion.

4 If you require, a transaction-id can be attached to this program to be used for testing. There are three 'exec CICS enter traceid.....'

commands that have been commented out which must be uncommented if you wish to test this program first. The ‘enter traceid’ commands will allow the program to be run under EDF and alter the storage areas to make the program perform the various criteria, without having these criteria actually exist in a currently running CICS/ICCF.

5 Add this program to both your CICS start-up and shutdown PLTs. To do this, perform the following and then submit both for assembly. Note: before performing this task it is highly recommended that you copy both your current CICS PLT phases to another library/sublibrary. This will ensure that, should anything happen that prevents your CICS from starting up, you can simply copy them back to return to the position before any changes were made.

– PLTSI start-up – add the following statement ahead of the statement containing ‘dfhplt type=final’:

```
dpplti   dfhplt type=entry,program=dpplti
         dfhplt type=final
```

– PLTSI shutdown – add the following statement ahead of the statement containing ‘program=dfhdelim’:

```
dpplti   dfhplt type=entry,program=dpplti
         dfhplt type=entry,program=dfhdelim
```

The program can be added to any CICS PLT, even if ICCF is not used in a particular CICS. The code has been written such that, if ICCF is not in a particular CICS, the $RC/$MRC return codes will still be set, but the check for submitting the DTSANALS RECOVER function will be bypassed.

6 The use of this program, for the automatic submission of the DTSANALS RECOVER function, requires that you make one or both of the following changes:

– Remove or comment out the following two statements in your current CICS/ICCF start-up JCL:

```
// exec DTSANALS,size=DTSANALS
recover opt
```

The above statements should already be commented out. If they haven't been, you have been penalizing your non-ICCF users for the sake of your ICCF users, since leaving them in would delay the start-up of your CICS/ICCF during the DTSANALS recovery. It's likely that you have many more non-ICCF users than ICCF users. This scenario assumes that you are running normal transactions and ICCF in the same CICS.

– Check the JCL, at the back of this program, indicated by three percent signs (ie %%%) in columns 69-71, and change them as necessary to conform to your standards. You can add or remove statements without having to worry about changing any of the source code. It is believed that the code contained in this program is the same as that used by IBM to determine whether the DTSFILE needs DTSANALS recover function to be run. IBM issues message 'K008I ICCF library may have been destroyed – run DTSANALS recover'. This message is usually followed by 'K131I dtsfile is disconnected'. If these messages are issued and this program doesn't submit DTSANALS, then something is probably wrong. The most likely cause will be this program, possibly due to a change in ICCF control blocks.

7 The use of this program, for the automatic restarting of CICS, requires that you make the following changes:

– Your CICS start-up JCL:

1 Add the following statements immediately following the '// job' statement:

```
// on $abend or $cancel goto abend
// on $rc > 2 goto abend
```

2 Add the following statement immediately following the '// exec dfhsip' statement:

```
/. abend
```

3 Add the following statements immediately following the /* of the execute for DTRSETP. If you have removed

DTRSETP then add the following after the '/. abend' statement added in the above step:

```
// if $mrc ne 2 then
// pwr prelease rdr,*CICSjobn
```

CICSjobn is the POWER job name of your CICS.

4 Create an additional CICS start-up job in your on-line editor (ie ICCF, BIM-EDIT, etc). The member name will be different but not the POWER job name. When changes are made in the future, ensure that the changes are made to both copies, or delete one of the copies, make the changes to the remaining one, and then copy the changed one, using the name just deleted as the target name. The POWER disposition (ie disp=) of both jobs should be 'l'. After making these changes submit both jobs.

The basic point of this step is that there must be a way to release/submit another CICS start-up job from within the one that is still executing (active). VSE/POWER does not allow a queue entry to be 'altered' while the target job of the 'alter' is still executing. This is why you need to put another copy of your CICS start-up job in the reader queue. There are other ways to do this and, of course, you can use whichever way you wish, as long as it achieves the desired result. Be aware that, if a CICS is terminated with the POWER PFLUSH command, automatic restart will not occur because VSE/ESA does not trap this condition via the 'on abend' or 'on cancel' condition.

– Shutting down your CICSs:

With the addition of this program to your PLTs the shutting down of CICS may change from the current method and any new method must be adhered to. You must perform the following when shutting down your CICSs so that this program can be invoked:

```
cemt per shut
cemt per shut imm
```

Between the above two commands, normally entered from the system console, you may wait any number of seconds before issuing the second command. The above method is mandatory because, without it, CICS will not perform stage two PLTSD processing and the program will not be able to set the $RC/$MRC to 2, since it will never get invoked. If unable to set the $RC/$MRC value to 2, your CICS will automatically start up even though you may not have wanted it to do so. On the other hand, issuing the first command, without the second, assuming your CICS will come down without issuing the second command, would eliminate the need for the operator to issue a prelease RDR, or whatever, for a subsequent start-up of CICS.

– Abnormal termination of your CICSs:

1 You should be aware that a caveat exists, namely recursive recycling, which could occur regarding the method chosen to restart a CICS that has abnormally terminated. If your CICSs are unstable or encounter a transaction that continually trashes CICS, that CICS will continually be recycled, especially if the system is running unattended at the time this happens. Eventually, this recursive scenario will cease when the POWER LST queue gets filled up by all the dumps.

2 The setting of the $RC/$MRC return code to 1 upon start-up and to 2 upon shutdown is purely arbitrary and you may change it if desired. However, if either is changed you must make certain that the CICS execution JCL is also changed to check for the correct final return code.

3 This program might be able to execute above the 16MB line but this has not been put to any formal test.

4 The use of ‘DPRTNC’, from within an ICCF pseudo-partition, is not recommended because it may alter the $RC/$MRC that this program sets, there being only one set of these return codes per partition, which includes

ICCF pseudo-partitions.

5 This program was written and tested on VSE/SP and VSE/ESA, the most recent being VSE/ESA 1.3.6 with CICS 2.2. It is not known whether it will function on other versions/releases of either VSE or CICS without modification, although there is no reason to suspect that it wouldn't. This also applies to the two subroutines this program uses.

Subroutines DPCKJP and DPRTNC are called by this program.

```
DPPL     TITLE 'DPPLTI - 1.Ø - SET VSE RETURN CODE/SUBMIT DTSANALS PROGX
               RAM.'
* CICS CSA.
         DFHCSAD ,                 CSA DSECT.
*        COPY  DFHCSADS            COPY CSA DSECT.
* SYSTEM COMMUNICATIONS REGION. (SYSCOM).
SYSCOM   SYSCOM ,
* ICCF VECTOR TABLE. (DTSVECDS).
         DTSVECTB DSECT=YES ,
* ICCF MCS. (DTSMCSAD).
         DTSMCSAD ,
DFHEISTG DSECT                     EXEC INTERFACE STORAGE DSECT.
HERE     DS    CL12                STORAGE CONSTANT.
SCSASSI2 DS    C                   CICS CSA SYSTEM SIGNAL INDICATOR 2 S
SMCSLOPN DS    C                   ICCF DTSFILE OPEN SWITCH SAVE AREA.
SMCSFLG2 DS    C                   ICCF DTSFILE FLAG 2 SWITCH SAVE AREA
TOKEN    DS    XL8                 TOKEN FOR SPOOL COMMANDS.
RECORD   DS    CL8Ø                RECORD AREA USED TO SPOOL DTSANALS J
RESP     DS    F                   CICS RESPONSE AREA.
RESP2    DS    F                   CICS RESPONSE AREA TWO.
DPPLTI   DFHEIENT DATAREG=R2,CODEREG=R3,EIBREG=R4 , ESTABLISH BASE, COD
DPPLTI   AMODE 24                                                  ###
DPPLTI   RMODE 24                                                  ###
         MVC   HERE,=C'DPPLTI HERE.' INSERT EYE CATCHER.
         EXEC  CICS ADDRESS CSA(RD). GET CSA ADDRESS.
         MVC   SCSASSI2,CSASSI2    MVE SYSTEM SIGNAL INDICATOR  (X'49')
*        EXEC  CICS ENTER TRACEID(1) FROM(SCSASSI2). ???
* THE FOLLOWING CODE DETERMINES WHAT MODE CICS IS CURRENTLY IN. A
* CSASSI2 (SYSTEM SIGNAL INDICATOR 2) VALUE OF X'ØØ' INDICATES THAT
* CICS IS IN START-UP MODE (IE PLTPI MODE). A VALUE OF X'1Ø' INDICATES
* THAT CICS HAS BEEN STARTED BUT IS NOT IN EITHER PLT MODE (IE START-UP
* OR SHUTDOWN). IN THIS MODE THERE IS NO CHECKING DONE TO DETERMINE IF
* THE DTSFILE NEEDS TO BE RECOVERED OR THE SETTING OF THE $RC/$MRC VAL-
* UES. IF ANY OTHER VALUE IT IS ASSUMED THAT CICS IS IN SHUTDOWN MODE
* (IE PLTSD MODE). IN CICS 2.3, AND ABOVE, IBM HAS PROVIDED A SPI
* (SYSTEM PROGRAMMERS INTERFACE) TO DETERMINE WHAT MODE CICS IS CUR-
```

```
* RENTLY IN. THIS SPI AVOIDS THE USE OF THE CSA, WHICH WILL, IN THE
* FUTURE, BE UNSUPPORTED. IF RUNNING CICS 2.3 OR ABOVE IT'S SUGGESTED
* YOU REPLACE THE ABOVE THREE AND THE NEXT FOUR INSTRUCTIONS
* WITH 'EXEC CICS INQUIRE SYSTEM CICSSTATUS(CVDA)'. THE APPROPRIATE
* CVDA VALUES WILL BE 'START-UP' (TO REPLACE X'ØØ), 'ACTIVE' (TO RE-
* PLACE X'1Ø') AND 'FIRSTQUIESCE' (TO REPLACE NOT X'1Ø' AND X'ØØ').
         CLI   SCSASSI2,X'1Ø'      HAS CICS ALREADY BEEN STARTED. (NO A
         BE    RETURN3             YES-BRANCH TO RETURN3.
         CLI   SCSASSI2,X'ØØ'      IS CICS PLTPI COMPLETED.
         BNE   SHUT                YES-BRANCH TO SHUT.
STRT     EQU   *                   CICS STARTING UP ROUTINE.
* THE FOLLOWING CODE DETERMINES IF 'ICFSAV', 'ICFRES' OR 'DTSANALS'
* IS EXECUTING, IN ANY PARTITION, VIA DPCKJP. IF SO, THE SUBMITTING OF
* DTSANALS RECOVER FUNCTION IS NOT PERFORMED. THIS IS DONE TO PREVENT
* THE DTSANALS RECOVER JOB FROM BEING SUBMITTED DURING A DTSFILE BACK-
* UP/RESTORE AND TO PREVENT IT FROM BEING SUBMITTED MULTIPLE TIMES.
* NOTE: IF YOUR ICCF BACKUP OR ICCF RESTORE JOBS ARE NAMED DIFFERENTLY
* YOU MUST CHANGE THE STORAGE CONSTANTS, LOCATED AT THE BACK OF THIS
* PROGRAM INDICATED WITH THREE AT SIGNS (IE @@@) IN COLUMNS 69-71,
* TO MATCH THOSE OF YOUR ICCF BACK-UP AND RESTORE JOB NAMES.
         MVC   CKJPPM1,BLANKS      CLEAR CKJP PARAMETER.
         MVC   CKJPJOB,ICCFSAVE    SET JOB NAME.
         BAL   RA,CKJP             PERFORM CKJP ROUTINE.
         CLI   CKJPRCD,C'1'        IS 'ICFSAV' EXECUTING.
         BE    ALRDYR              YES-BRANCH TO ALRDYR.
         MVC   CKJPPM1,BLANKS      CLEAR CKJP PARAMETER.
         MVC   CKJPJOB,ICCFREST    SET JOB NAME.
         BAL   RA,CKJP             PERFORM CKJP ROUTINE.
         CLI   CKJPRCD,C'1'        IS 'ICFRES' EXECUTING.
         BE    ALRDYR              YES-BRANCH TO ALRDYR.
         MVC   CKJPPM1,BLANKS      CLEAR CKJP PARAMETER.
         MVC   CKJPPGM,DTSANALS    SET PROGRAM NAME.
         CALL  DPCKJP,(CKJPPM1)    GO CHECK IF 'DTSANALS' IS EXECUTING.
         CLI   CKJPRCD,C'1'        IS 'DTSANALS' EXECUTING.
         BE    ALRDYR              YES-BRANCH TO ALRDYR.
* THE FOLLOWING CODE DETERMINES IF ICCF IS ACTIVE IN THE SYSTEM, BY
* CHECKING IF THE ICCF VECTOR ADDRESS IS ZERO. IF ZERO (NO ICCF ACTIVE)
* WE SKIP ANY OTHER CHECKS. OTHERWISE WE USE THE 'DTSVECDS' DSECT TO
* ACQUIRE THE 'MCSA' ADDRESS IF IT TOO IS NOT ZERO. IF SO WE SKIP ANY
* OTHER CHECKS. OTHERWISE WE ACQUIRE THE 'DTSMCSAD' AND IF IT TOO IS
* ZERO WE SKIP ANY OTHER CHECKS. OTHERWISE WE PERFORM AN IDENTITY CHECK
* AND, IF THE LITERAL STRING EXISTS, WE SAVE THE 'MCSLOPEN' FLAG AND
* 'MCSFLAG2' BYTES. THE LATER IS SAVED BUT NOT USED. FINALLY, WE CHECK
* THE SAVED 'MCSLOPEN' BYTE TO SEE IF THE DTSFILE SHOULD BE RECOVERED.
* IF SO WE SUBMIT A DTSANALS JOB TO RECOVER THE DTSFILE. WE THEN BRANCH
* TO SET THE VSE/ESA $RC/$MRC VALUE TO 1 AND RETURN TO CICS.
         ASYSCOM (R1)              GET SYSTEM COMMUNICATIONS ADDRESS.
         USING SYSCOM,R1           INFORM ASSEMBLER.
         L     RE,IJBETSS          LOAD ADDRESS OF ICCF VECTOR (X'11C')
         LTR   RE,RE               IS IT ZERO.
```

```
         BZ    RETURN              YES-BRANCH TO RETURN.
         DROP  R1                  (SYSCOM).
         USING DTSVECDS,RE         INFORM ASSEMBLER.
         LTR   RE,RE               IS IT ZERO.
         BZ    RETURN              YES-BRANCH TO RETURN.
         L     R1,=X'FF000000'     SET ENABLE STORAGE PROT KEY.
         SVC   13                  GO DO IT.
         L     RF,DTSMCSA          LOAD ADDRESS OF MCSA TO REG (X'0C')
         DROP  RE                  (DTSVECDS).
         LTR   RF,RF               IS IT ZERO.
         BNZ   STRT1               NO-BRANCH TO STRT1.
         L     R1,=X'FF0000FF'     RESET ENABLE STORAGE PROT KEY.
         SVC   12                  GO DO IT.
         B     RETURN              BRANCH TO RETURN.
STRT1    EQU   *
         USING DTSMCSAD,RF         INFORM ASSEMBLER.
         CLC   =C'*$ICCF$*',MCSID  ARE WE IN ICCF'S PARTITION. (X'08')
         BE    STRT3               YES-BRANCH TO STRT3.
         L     R1,=X'FF0000FF'     RESET ENABLE STORAGE PROT KEY.
         SVC   12                  GO DO IT.
         B     RETURN              BRANCH TO RETURN.
STRT3    EQU   *
         MVC   SMCSLOPN,MCSLOPEN   SVE MCSLOPEN FLAG.          (X'F7')
         MVC   SMCSFLG2,MCSFLAG2   SVE MCSFLAG2 FLAG.          (X'138')
         DROP  RF                  (DTSMCSAD).
*        EXEC  CICS ENTER TRACEID(2) FROM(SMCSLOPN). ???
         TM    SMCSLOPN,X'C0'      DOES DTSFILE NEED RECOVER FUNCTION.
         BNO   RETURN              NO-BRANCH TO RETURN.
         EXEC  CICS SPOOLOPEN REPORT('DTSANALS') TOKEN(TOKEN) JCL      X
               LOGICAL RESP(RESP) RESP2(RESP2).
         CLC   RESP,DFHRESP(NORMAL) WAS SPOOLOPEN SUCCESSFUL.
         BNE   SPOLOE              NO-BRANCH TO SPOLOE.
         LA    RB,RECDS            LOAD ADDRESS OF RECDS TO REG 11.
STRT7    EQU   *
         MVC   RECORD,0(RB)
         EXEC  CICS SPOOLWRITE REPORT('DTSANALS') TOKEN(TOKEN)         X
               FROM(RECORD) FLENGTH(L'RECORD) RESP(RESP) RESP2(RESP2).
         CLC   RESP,DFHRESP(NORMAL) WAS SPOOLWRITE SUCCESSFUL.
         BNE   SPOLER              NO-BRANCH TO SPOLER.
         LA    RB,L'RECORD(RB)     INCREMENT TO NEXT RECORD.
         CLI   0(RB),X'FF'         ARE WE DONE.
         BNE   STRT7               NO-BRANCH TO STRT7.
         EXEC  CICS SPOOLCLOSE TOKEN(TOKEN) RESP(RESP) RESP2(RESP2).
         EXEC  CICS WRITE OPERATOR TEXT(SPOOLGD)                       X
               TEXTLENGTH(L'SPOOLGD).
         B     RETURN              BRANCH TO RETURN.
* THE FOLLOWING CODE SETS THE VSE $RC/$MRC TO 2 DURING THE SHUT DOWN
* PROCESS. WE THEN BRANCH TO RETURN TO CICS.
SHUT     EQU   *                   CICS SHUTTING DOWN ROUTINE.
*        EXEC  CICS ENTER TRACEID(3) FROM(SCSASSI2). ???
```

```
         MVC   RTNCVLU,=C'0002'    SET $RC/$MRC TO 2.
         BAL   RA,RTNC             PERFORM RTNC ROUTINE.
         B     RETURN3             BRANCH TO RETURN3.
RETURN   EQU   *                   RETURN TO CICS ROUTINE.
         MVC   RTNCVLU,=C'0001'    SET $RC/$MRC TO 1.
         BAL   RA,RTNC             PERFORM RTNC ROUTINE.
RETURN3  EQU   *
         EXEC  CICS RETURN.        RETURN TO CICS.
ALRDYR   EQU   *              ICFRES/ICFSAV/DTSANALS RUNNING ROUTINE.
         EXEC  CICS WRITE OPERATOR TEXT(ALRDYRU)                       X
               TEXTLENGTH(L'ALRDYRU). SEND MESSAGE.
         B     RETURN              BRANCH TO RETURN.
SPOLOE   EQU   *                   SPOOLOPEN ERROR ROUTINE.
         EXEC  CICS WRITE OPERATOR TEXT(SPOOLOE)                       X
               TEXTLENGTH(L'SPOOLOE). WRITE SPOOLOPEN ERROR MESSAGE.
         B     RETURN              BRANCH TO RETURN.
SPOLER   EQU   *                   SPOOLWRITE ERROR ROUTINE.
         EXEC  CICS SPOOLCLOSE REPORT('DTSANALS') TOKEN(TOKEN)         X
               DELETE NOHANDLE.    DELETE SPOOL.
         EXEC  CICS WRITE OPERATOR TEXT(SPOOLER)                       X
               TEXTLENGTH(L'SPOOLER). WRITE SPOOLWRITE ERROR MESSAGE.
         B     RETURN              BRANCH TO RETURN.
CKJP     EQU   *                   CALL DPCKJP ROUTINE.
         MVI   CKJPFUN,C'C'        SET FUNCTION CODE.
         MVI   CKJPRCD,C'Y'        INDICATE EXCLUDE EXECUTING PARTITION
         LA    RD,SAVEAREA         LOAD ADDRESS OF SAVEAREA TO REG 13.
         CALL  DPCKJP,(CKJPPM1)    GO CHECK IF JOB/PROGRAM IS EXECUTING
         BR    RA                  RETURN TO CALLER.
RTNC     EQU   *                   CALL DPRTNC ROUTINE.
         MVI   RTNCFUN,C'3'        INDICATE SET $RC/$MRC.
         LA    RD,SAVEAREA         LOAD ADDRESS OF SAVEAREA TO REG 13.
         CALL  DPRTNC,(RTNCPM1)    GO SET $RC/$MRC VALUE.
         CLI   RTNCFUN,C'6'        WAS THERE AN ERROR.
         BNH   RTNC3               NO-BRANCH TO RTNC3.
         MVC   RTNCERR+L'RTNCERR-1(1),RTNCFUN MVE RETURN CODE TO RTNCER
         EXEC  CICS WRITE OPERATOR TEXT(RTNCERR)                       X
               TEXTLENGTH(L'RTNCERR). SEND MESSAGE.
         BR    RA                  RETURN TO CALLER.
RTNC3    EQU   *
         MVC   RTNCOK+L'RTNCOK-1(1),RTNCVLU+3 MVE RETURN CODE VALUE TO
         EXEC  CICS WRITE OPERATOR TEXT(RTNCOK)                        X
               TEXTLENGTH(L'RTNCOK). SEND MESSAGE.
         BR    RA                  RETURN TO CALLER.
SAVEAREA DS    18F                 SVE AREA FOR CALLS.
BLANKS   DC    CL50' '             BLANKS USED FOR CLEARING/COMPARING.
ICCFSAVE DC    CL8'ICFSAV'         JOB NAME FOR ICCF BACKUP.        @@@
ICCFREST DC    CL8'ICFRES'         JOB NAME FOR ICCF RESTORE.       @@@
DTSANALS DC    CL8'DTSANALS'       PROGRAM NAME FOR DTSANALS.
ALRDYRU  DC    CL50'DPPLTI - ICFSAV/ICFRES/DTSANALS RUNNING.'
SPOOLGD  DC    CL50'DPPLTI - DTSANALS SUBMITTED.'
```

```
SPOOLOE  DC    CL5Ø'DPPLTI - SPOOL OPEN ERROR.'
SPOOLER  DC    CL5Ø'DPPLTI - DTSANALS SPOOL ERROR.'
RTNCOK   DC    CL21'DPPLTI - $RC SET TO X'
RTNCERR  DC    CL33'DPPLTI - DPRTNC CALL ERROR, R/C= '
CKJPPM1  DS    ØCL45          DPCKJP PARAMETER.
CKJPFUN  DS    C'C'                FUNCTION CODE. (IE. CHECK).
CKJPJOB  DS    CL8                 JOB NAME.
CKJPPGM  DS    CL8                 PROGRAM NAME.
CKJPRCD  DS    C                   RETURN CODE.
         DS    C                   OPTION BYTE. (NOT USED).
         DS    CL24                PARTITION-IDS. (NOT USED).
         DS    XL2                 NUMBER OF PARTITION-IDS. (NOT USED)
RTNCPM1  DS    ØCL5           DPRTNC PARAMETER.
RTNCFUN  DC    C'3'                FUNCTION CODE. (IE. SET).
RTNCVLU  DC    C'ØØØ1'             $RC/$MRC VALUE.
RECDS    DS    ØC             RECORDS SUBMITTED FOR DTSANALS RECOVER JO
         DC    CL8Ø'* $$ JOB JNM=DTSANALS,CLASS=R,PRI=9,USER=*OPS*' %%%
         DC    CL8Ø'* $$ LST DISP=H,CLASS=O,PRI=8,PURGE=1'          %%%
         DC    CL8Ø'// JOB DTSANALS DPØØ ICCF RECOVER DTSFILE.'     %%%
         DC    CL8Ø'// EXEC DTRIATTN,PARM=''L LST,*DTSANALS'''      %%%
         DC    CL8Ø'*  JOB DTSANALS - DTSANALS - RECOVER DTSFILE.'  %%%
         DC    CL8Ø'// EXEC DTRIATTN,PARM=''/DISC DTSFILE'''        %%%
         DC    CL8Ø'// EXEC IESWAIT,PARM=''5'''                     %%%
         DC    CL8Ø'// EXEC PROC=DTRICCF'                           %%%
         DC    CL8Ø'// EXEC DTSANALS,SIZE=DTSANALS'                 %%%
         DC    CL8Ø'RECOVER OPT'                                    %%%
         DC    CL8Ø'/*'                                             %%%
         DC    CL8Ø'*  JOB DTSANALS - DTRIATTN - CONNECT DTSFILE.'  %%%
         DC    CL8Ø'// EXEC DTRIATTN,PARM=''/CONN DTSFILE'''        %%%
         DC    CL8Ø'/&&'                                            %%%
         DC    CL8Ø'* $$ EOJ'                                       %%%
         DC    X'FF'               END OF TABLE-DO NOT MOVE/REMOVE.
         LTORG ,                   DISPLAY LITERALS.
         END   DPPLTI
```

DPCKJP

The DPCKJP subroutine will check each partition and inform the calling program whether a passed job and/or program is executing, and in what partition it's executing.

One parameter must be passed, consisting of seven fields, while a second parameter consisting of two fields is optional. The first parameter must consist of the following.

The first field is a one-byte code indicating the function to be performed. Specify 'C' to indicate that a check is to be made for the

specified job/program names entered in the next two fields, or specify 'W' to indicate that an automatic wait is to be performed if the specified job/program names entered in the next two fields are executing. If 'W' isn't specified, 'C' is defaulted to. See notes for more information.

The second field consists of eight bytes containing a job name that is to be checked or waited upon. If it is less than eight bytes, left justify and pad it on the right with blanks. If you're not interested in a specific job name set this field to blanks. See notes for more information.

The third field consists of eight bytes, containing a program name that is to be checked or waited upon. If it is less than eight bytes, left justify and pad it on the right with blanks. If you're not interested in a specific program name set this field to blanks. See notes for more information.

The fourth field is one byte and will contain one of the following values upon return to the calling program:

0 Either the passed job and/or program name were not found in any partition. Function 'W' always returns this value, assuming return codes 4 through 9 weren't returned.

1 A request to check the passed job name only was satisfied. The partition(s) it was located in are indicated in the sixth field.

2 A request to check the passed program name only was satisfied. The partition(s) it was located in are indicated in the sixth field.

3 A request to check the passed job and program names was satisfied. The partition(s) they were located in are indicated in the sixth field.

4 Mixed generic and skip characters were specified. The first position of either the job or program name contained an asterisk and some other position contained a plus sign, or some position contained a plus sign and another an asterisk.

5 A request to exclude/include specific partition-ids was made but the partition-ids entered in the fifth field were not BG, FA, FB, or F1 through F9.

6 Either the passed job name or the passed program name were

blank in the first position but not both.

7 Both the passed job and program names were blank.

8 There are too many active dynamic partitions, which are held in an internal table capable of holding 64 entries.

9 An internal error occurred. This error should never occur and, if it does, indicates that your system isn't capable of supporting this subroutine.

If this field contains the character 'Y', prior to each call, then the partition the calling program is executing in will be bypassed. The default is to check all partitions that were generated via the 'NPARTS=' parameter in the supervisor, active or inactive. It's not necessary to clear this field prior to each call unless you wish to bypass the partition of the calling program.

The fifth field is one byte and will contain an option byte indicating one of the following values:

E indicates that the sixth field contains a list of partition-ids that are to be excluded from checking.

I indicates that the sixth field contains a list of partition-ids that are to be included for checking. Note: a request to include a partition-id, for example F4, together with bypassing the partition-id the calling program is executing in, for example F4, is ignored.

X where 'X' indicates any other value. If 'E' or 'I' are not specified this is the default (ie all partitions, except that indicated by a 'Y' in the fourth field, are to be included).

If you wish to exclude/include partition-ids you must set this field to 'E' or 'I' prior to each call.

The sixth field is twenty-four bytes and will contain the partition-ids the specified job/program was executing in, if found, upon return to the calling program. For example, if a check was requested to locate a job name of 'CICS' and two CICS systems are running, one in F2 and the other in F9, this field will contain:

```
'f9f2'
```

If no match occurred this field will be blank. The partition-ids are returned with BG first, followed by FB, FA, F9 through F1, assuming all partitions were generated. If the first position contains the character 'Y' then the partition-ids and both the fields of the second parameter, if passed, will be returned in priority order as specified via the 'PRTY' command. If you are excluding/including partition-ids, as indicated by the previous field, you must move the desired partition-ids that are to be excluded/included prior to each call, otherwise set this field to blanks prior to each call. The partition-ids moved must be BG, FA, FB, or F1 through F9 left justified.

The seventh field is two bytes and will contain a binary value containing the number of partitions the specified job/program was executing in, if found. It's not necessary to clear this field prior to each call.

The optional second parameter must consist of the following.

The first field is 192 bytes and will contain the full names of the job/programs found executing in the partition(s) that were returned in the fifth field of the first parameter. The entries consist of twelve 8-byte pairs containing the job and program name contained in the partition-ids. Using the above example this field would contain:

```
'CICSf9  dfhsip  CICSf2  dtsinit ........'
```

If no match occurred this field will be blank. The pairs are returned in the same manner as described for the partition-ids. Both fields are returned, regardless of the specification of the second two fields of the first parameter. It's not necessary to clear this field prior to each call.

The second field is 48 bytes and will contain the full word partition communications address of the partition(s) that were returned in the fifth field of the first parameter. If no match occurred this field will contain low-values. The addresses are returned as described above. It's not neccessary to clear this field prior to each call.

Notes:

1 When the 'W' (wait) function is used, the partition in which the calling program is executing is bypassed to avoid the calling program waiting on itself (ie the fourth field of the first parameter

is set to 'Y'). A check is then made to determine whether more than one partition is waiting for the same job/program. This avoids the problem of multiple partitions waiting on each other. If so, the partition with the highest dispatching priority, as set via the 'PRTY' command, will be allowed to run while the other partition(s) will wait for the completion of the higher priority partition. If not, the partition will simply wait until completion of the specified job/program. Prior to waiting, one of the following messages will be issued to the system console:

```
waiting on job jjjjjjjj in xx,xx,...
waiting on program pppppppp in xx,xx,...
waiting on job/program jjjjjjjj/pppppppp in xx,xx,...
```

This message is issued once for each partition waiting. If other partitions are running the same job/program and they are waiting, the message will be reissued. In this case the message is reissued with the additional partition-id(s) contained within the message. A 30-second wait occurs for each partition that is waiting, at which time another check is made to determine whether the specified job/program has completed. If so, return is made to the calling program, otherwise the process is repeated.

2 If the return code is greater than '3' any information returned won't be valid. Therefore, this field should be checked after each call.

3 Either the job name field or the program name field may be blank, but not both. Specifying non-blanks in the job name field, and setting the program name field to blanks, causes a check to be made for the job name only. Specifying non-blanks in the program name field, and setting the job name field to blanks, causes a check to be made for the program name only. When both fields aren't blank, a positive condition is returned only if both the job and program names were found to be executing.

4 If you don't wish to specify 'DTSINIT' or 'DFHSIP,'a special program name of '--CICS--' may be specified. This special name is useful if ICCF/CICS or CICS has been started by a program that attached DTSINIT or DFHSIP. Thus the communication region phase name field would contain the executed program name (ie

the attaching program name) rather than DTSINIT or DFHSIP.

5 Generic specification is supported for both the job and program name fields. For example, it's possible to check for a job name containing specific characters anywhere within the job name field. The only restriction is that you can't begin each field with a blank. Use of this method, together with the skip method discussed below, will cause an error. Some examples of job/program fields are:

a '*paychks'

b '*init '

c '*init* '

d 'dfh* '

e '*prog+ '

f '* '

Where:

a causes a positive condition to be returned if any job/program contains the characters 'PAYCHKS'. Specifying an asterisk in the first position causes a match if the characters to the right of the asterisk are located anywhere within the job/program name.

b causes a positive condition to be returned if any job/program contains the characters 'INIT'.

c same as 'b'.

d causes a positive condition to be returned if any job/program contains the characters 'DFH' in positions 1-3 of the job/program name. Specifying an asterisk preceeded by one or more characters causes a match if the characters to the left of the asterisk are located in the exact positions within the job/program name.

e causes an error return code of 4 to be returned, because mixing of generic and skip characters isn't allowed.

f causes a return code of 0 to be returned.

6 Skip characters are also supported for both the job and program name fields. For example, it's possible to check for a job name beginning with specific characters or containing specific characters in certain positions. The only restriction is that you can't begin each field with a blank. Note that the first occurrence of a blank terminates the search. Use of this method, together with the generic method discussed above, will cause an error. Some examples of job/program fields are:

a '*paychks '

b '+pw++++r'

c '++++CICS'

d '+ '

e 'dt+++++ '

f 'dt+ '

g 'no+name '

h 'no name '

i ' test '

j ' '

k '+prog* '

Where:

a causes a positive condition to be returned if any job/program contains the characters 'PAYCHKS' in positions 1-8.

b causes a positive condition to be returned if any job/program contains the characters 'P' and 'W' in positions 2-3 and 'R' in position 8.

c causes a positive condition to be returned if any job/program contains the characters 'CICS' in positions 5-8.

d causes a positive condition to be returned if any job/program

contains any characters in positions 1-8.

e causes a positive condition to be returned if any job/program contains the characters ‘DT’ in positions 1-2 and any characters in positions 3-8.

f same as ‘e’. A blank following a plus sign terminates the search, so it’s not necessary to fill the right positions with additional plus signs.

g causes a positive condition to be returned if any job/program contains the characters ‘no’ in positions 1-2, and ‘name ‘ in positions 4-8. The use of these characters can inform you which partitions aren’t currently in use.

h causes a positive condition to be returned if any job/program contains the characters ‘no’ in positions 1-2. Note that the remaining portion (ie ‘name’) is ignored because of the embedded blank.

i causes an error return code of 6 if entered for a job name or ‘7’ if entered for a program name.

j causes an error return code of 8 if entered for both the job name and program name.

k causes an error return code of 4 to be returned, as mixing of skip and generic characters isn’t allowed.

The calling sequence is:

COBOL:

```
call 'dpckjp' using parm1.
```

or:

```
call 'dpckjp' using parm1,  parm2.
```

ALC:

```
la    13,savearea           (13 can also be r13 or rd).
call  dpckjp,(parm1)
```

or:

```
call  dpckjp,(parm1,parm2)
```

```
          .
          .  (main part of program).
          .
        savearea dc    18f'Ø'
```

RPG2:

```
        call 'dpckjp'
        parm           parm1
```

or:

```
        parm           parm2
```

An 18-word save area must be passed through register 13 by the user (std COBOL linkage).

```
DPCK    TITLE 'DPCK JP - 1.Ø - CHECK PARTITIONS FOR PASSED JOB/PROGRAM X
              SUBROUTINE.'
DPCKJP  CSECT Ø
        SAVE  (14,12)
        BALR  R2,Ø
        USING *,R2,R3              INFORM ASSEMBLER.
        LA    R3,4Ø95(R2)          LOAD SECOND BASE REG WITH
        LA    R3,1(R3)                CONTENTS OF FIRST +4Ø96.
        ST    RD,SAVEAREA+4
        LA    RD,SAVEAREA
        B     CKBEG
        DC    C'DPCKJP STARTS HERE. ' INSERT EYE CATCHER.
CKBEG   EQU   *
        ST    R1,SVR1              SVE CONTENTS OF REG 1.
        LA    R6,TAB               LOAD ADDRESS OF TAB TO REG 6.
CKBEG2  EQU   *
        MVC   1(L'PIBLOGID,R6),BLANKS CLEAR PARTITION-ID.
        MVC   3(L'PIBLOGID,R6),=F'Ø' CLEAR PIK.
        MVC   5(L'INPJOBN+L'INPPGMN,R6),BLANKS CLEAR JOB/PROGRAM NAMES
        MVC   21(L'PIBLOGID+2,R6),=F'Ø' CLEAR COMMUNICATIONS ADD
        LA    R6,L'TAB(R6)         INCREMENT REG 6 TO NEXT POSITIONS.
        CLI   Ø(R6),X'FF'          ARE WE DONE.
        BNE   CKBEG2               NO-BRANCH TO CKBEG2.
        MVI   ESASW,C'Ø'           INDICATE NOT RUNNING VSE/ESA.
        ASYSCOM (R1)               GET SYSTEM COMMUNICATIONS ADDRESS.
        ST    R1,ASYSCOM           SVE IT.
        USING SYSCOM,R1            INFORM ASSEMBLER.
        L     R6,IJBRFTAB          LOAD RECORDER FILE TABLE TO REG 6.
        USING RFTABLE,R6           INFORM ASSEMBLER.
        CLI   RFREL,X'33'          ARE WE RUNNING ANY VSE/ESA RELEASE.
        BL    CKBEG23              NO-BRANCH TO CKBEG23.
        DROP  R1,R6                (SYSCOM) (RFTABLE).
        MVI   ESASW,C'1'           INDICATE RUNNING VSE/ESA.
```

```
CKBEG23  EQU   *
         COMRG                     GET COMMUNICATIONS REGION.
         ST    R1,ACOMRG           SVE IT.
         USING COMREG,R1           INFORM ASSEMBLER.
         MVC   JOBNAME,COMNAME     SVE JOB NAME. (X'18').
         MVC   PGMNAME,IJBPHNAM    SVE EXECUTING PROGRAM NAME. (X'D8').
         DROP  R1                  (COMREG).
         AMODESW QRY ,             GET AMODE.
         STCM  R1,B'1000',AMODE    SVE HIGH ORDER BYTE.
         L     R1,SVR1             RESTORE CONTENTS OF REG 1.
         L     R4,0(R1)            LOAD PASSED PARAMETER TO REG 4.
         MVC   INPFLD1,0(R4)       MVE IT TO INPFLD1.
         CLI   INPFUNC,C'P' @@@
         BE    CKBEG25      @@@
         CLI   INPFUNC,C'G' @@@
         BNE   CKBEG27      @@@
CKBEG25  EQU   *            @@@
         MVC   INPFUNCS,INPFUNC @@@
CKBEG27  EQU   *            @@@
         CLI   INPFUNC,C'1'        IS FUNCTION '1'. (SAME AS 'C').
         BE    CKBEG5              NO-BRANCH TO CKBEG5.
         CLI   INPFUNC,C'2'        IS FUNCTION '2'. (SAME AS 'W').
         BE    *+12                YES-SKIP NEXT TWO (2) INST.
         CLI   INPFUNC,C'W'        IS FUNCTION 'W'.
         BNE   CKBEG3              NO-BRANCH TO CKBEG3.
         MVI   INPRCDE,C'Y'        INDICATE BYPASS PARTITION WERE RUNNI
         B     CKBEG5              BRANCH TO CKBEG5.
CKBEG3   EQU   *
         MVI   INPFUNC,C'C'        FORCE FUNCTION TO 'C'.
CKBEG5   EQU   *
         MVC   INPFLD1S,BLANKS     CLEAR PARAMETER 1 SAVE AREA.
         MVC   OPTN1(L'OPTN1+L'OPTN2+L'OPTN3),BLANKS
         MVC   OPTN1,INPRCDE       SVE OPTION BYTE ONE (1).
         MVC   OPTN2,INPPIDS       SVE OPTION BYTE TWO (2).
         MVC   OPTN3,INPOPTN       SVE OPTION BYTE THREE (3) AND PARTIT
         MVC   INPPIDSS,INPPIDS @@@
         MVI   NUMPRM,X'00'        SET NUMBER OF PARMETERS TO ZERO.
         SR    R6,R6               SET ARGUMENT COUNT TO ZERO.
CKARG    EQU   *
         TM    0(R1),X'80'         ARE WE DONE.
         BO    CKLST               YES-BRANCH TO CKLST.
         LA    R6,4(R6)            INCREMENT REG 6 BY ONE (1).
         LA    R1,4(R1)            INCREMENT REG 1 TO NEXT PARAMETER.
         B     CKARG               BRANCH TO CKARG.
CKLST    EQU   *
         SR    R1,R6               RESTORE REG 1.
         SRL   R6,2                DIVIDE REG 6 BY 2.
         LA    R6,1(R6)            INCREMENT BY ONE (1) FOR FIRST TIME.
         STC   R6,NUMPRM           SVE NUMBER OF PARAMETERS PASSED.
         CLI   NUMPRM,X'02'        WERE TWO (2) PARAMETERS PASSED.
```

```
         BNE   CKSTR               NO-BRANCH TO CKSTR.
         L     R5,4(R1)            LOAD ADDRESS OF SECOND PARAMETER.
         ST    R5,SVR5             SVE IT.
CKLST3   EQU   *
         MVC   INPFLD2,Ø(R5)       MVE PARAMETER TO INPFLD2.
         MVC   INPJBPG,BLANKS      CLEAR JOB/PROGRAM FIELD.
         XC    INPCOMR,INPCOMR     CLEAR PARTITION COMMUNICATIONS ADDRE
         LA    R5,INPJBPG          LOAD ADDRESS OF JOB/PROGRAM FIELD TO
         ST    R5,SVR5A            SVE IT.
         LA    R5,INPCOMR          LOAD ADDRESS OF PARTITION COMMUNICAT
         ST    R5,SVR5B            SVE IT.
CKSTR    EQU   *
         SR    R5,R5               CLEAR REG 5.
         MVI   PCBSTAP,Ø           CLEAR PCBSTAP.
         MVC   PCBSTAP+1(47),PCBSTAP ...
         MVI   PCBDYNP,Ø           CLEAR PCBDYNP.
         MVC   PCBDYNP+1(255),PCBDYNP ...
         MVC   PIDSD,BLANKS        CLEAR PIDSD.
         MVI   DYNSW,C'Ø'          INDICATE NOT RUNNING IN DYNAMIC PART
         CLI   ESASW,C'Ø'          ARE WE RUNNING UNDER VSE/ESA.
         BE    CKSTR5              NO-BRANCH TO CKSTR5.
         USING COMREG,RB           MAP TO COMREG.
         USING PCBADR,R6           MAP TO PCB/PCE.
         L     RA,APCBATAB         LOAD ADDRESS OF PCBATAB TO  (X'2C4')
         L     RA,Ø(RA)            LOAD ADDRESS OF PCBTAB TO REG 1Ø.
         MVC   SVAPCB,Ø(RA)        SVE FIRST 184 BYTES.
         CLI   PART,C'B'           ARE WE RUNNING IN A DYNAMIC PARTITIO
         BE    CKSTR13             NO-BRANCH TO CKSTR13.
         CLI   PART,C'F'           ARE WE RUNNING IN A DYNAMIC PARTITIO
         BE    CKSTR13             NO-BRANCH TO CKSTR13.
         MVI   DYNSW,C'1'          INDICATE RUNNING IN DYNAMIC PARTITIO
CKSTR13  EQU   *
         LA    RC,PCBSTAP          LOAD ADDRESS OF STATIC PARTITION PCB
         LA    RD,PIDS             LOAD ADDRESS OF STATIC PARTITION-IDS
CKSTR3   EQU   *
         LA    RA,4(RA)            INCREMENT TO NEXT PCB ENTRY.
         CLC   EPCBATAB,Ø(RA)      ARE WE AT THE END OF THE TABLE.
         BE    CKSTR5              YES-BRANCH TO CKSTR5.
         CLC   =F'Ø',Ø(RA)         IS THIS AN UNUSED DYNAMIC PART ENTRY
         BE    CKSTR3              YES-BRANCH TO CKSTR3.
         L     R6,Ø(RA)            LOAD ADDRESS OF ACTIVE DYNAMIC PCB T
         L     RB,PCECOMRA         LOAD ADDRESS OF ACTIVE DYNAMI (X'19Ø
         MVC   Ø(L'PCBDYNP,RC),Ø(RA) MVE DYNAMIC PARTITION PCB ADDRESS
         LA    RC,L'PCBDYNP(RC)    INCREMENT TO NEXT POSITION.
         MVC   Ø(L'PIDS,RD),IJBSPNLI MVE PARTITION-ID TO PIDS. (X'D4')
         LA    RD,L'PIDS(RD)       INCREMENT TO NEXT POSITIONS.
         LA    R5,1(R5)            INCREMENT REG 5 BY ONE (1).
         ST    R5,SVR5D
         CLI   Ø(RD),X'FF'         ARE WE AT THE END OF THE TABLE.
         BNE   CKSTR3              NO-BRANCH TO CKSTR3.
```

```
         LA    R6,1                SET DUMP INDICATOR.
CKSTR33  EQU   *
         MVI   INPRCDE,C'8'        INDICATE TOO MANY DYNAMIC PARTITIONS
         PDUMP DPCKJPS,DPCKJPM
         B     CKRTN9              BRANCH TO CKRTN9.
         DROP  RB,R6               (COMREG) (PCBADR).
CKSTR5   EQU   *
         EXTRACT ID=CPUID,AREA=CPUID,LEN=L'CPUID GET CPU/PARTITION-ID.
         LTR   RF,RF               WAS EXTRACT SUCCESSFUL.
         BZ    CKSTR53             YES-BRANCH TO CKSTR53.
         MVI   INPRCDE,C'9'        INDICATE EXTRACT MACRO FAILURE.
         B     CKRTN9              BRANCH TO CKRTN9.
CKSTR53  EQU   *
         MVC   INPRFLD,BLANKS      CLEAR RETURN FIELD.
         XC    INPPCNT,INPPCNT     CLEAR PARTITION COUNTER.
         CLI   OPTN3,C'E'          ARE WE EXCLUDING PARTITION-IDS.
         BE    CKSTR7              YES-BRANCH TO CKSTR7.
         CLI   OPTN3,C'I'          ARE WE INCLUDING PARTITION-IDS.
         BNE   CKBY9               NO-BRANCH TO CKBY9.
CKSTR7   EQU   *
         CLI   OPTN3+1,C' '        ARE ANY PARTITION-IDS SPECIFIED.
         BE    CKBY5               NO-BRANCH TO CKBY5.
         LA    RD,OPTN3+1          LOAD ADDRESS OF PASSED PARTITION-ID
CKBY1    EQU   *
         LA    RE,PIDS             LOAD ADDRESS OF PARTITION-IDS TO RE
         LA    RF,76               LOAD BRANCH COUNTER TO REG 15.
CKBY3    EQU   *
         CLC   =X'5ØD7',Ø(RD)      IS PARTITION-ID &P.
         BE    CKBY6               YES-BRANCH TO CKBY6.
         CLC   Ø(L'PIDS,RD),Ø(RE)  IS SPECIFIED PARTITION-ID VALID.
         BE    CKBY7               YES-BRANCH TO CKBY7.
         LA    RE,L'PIDS(RE)       INCREMENT REG 14 TO NEXT POSITIONS.
         BCT   RF,CKBY3            BRANCH TO CKBY7 UNTIL REG 15 ZERO.
CKBY5    EQU   *
         MVI   INPRCDE,C'5'        INDICATE PARTITION-IDS ERROR.
         B     CKRTN9              BRANCH TO CKRTN9.
CKBY6    EQU   *
         MVC   Ø(L'PART,RD),PART   REPLACE &P WITH CURRENT PARTITION-ID
CKBY7    EQU   *
         LA    RD,L'PIDS(RD)       INCREMENT REG 13 TO NEXT POSITIONS.
         CLI   Ø(RD),C' '          ARE WE DONE.
         BNE   CKBY1               NO-BRANCH TO CKBY1.
CKBY9    EQU   *
         CLI   PTYSW,C'1'          DID WE DO PRTY.
         BE    CKSKP               YES-BRANCH TO CKSKP.
         MVI   PTYSW,C'1'          INDICATE WE DID PRTY.
         LA    RØ,L'PIK            LOAD LENGTH OF PIK TO REG Ø.
         SR    R6,R6               CLEAR REG 6.
         AMODESW SET,AMODE=24,SAVE=(R6) SWITCH TO AMODE=24.
         GETPRTY PIK,(Ø) TYPE=ALL  GET PARTITION PRIORITIES.
```

```
         AMODESW SET,AMODE=(R6)    SWITCH TO AMODE=31.
CKSKP    EQU   *
         LA    R1,PIK              LOAD ADDRESS OF PIK TO REG 1.
         LA    R6,TAB              LOAD ADDRESS OF TAB TO REG 6.
CKNXT    EQU   *
         MVC   1(L'PIBLOGID,R6),BLANKS CLEAR PARTITION-ID.
         MVC   3(L'PIBLOGID,R6),Ø(R1) MVE PIK TO TABLE.
         MVC   5(L'INPJOBN+L'INPPGMN,R6),BLANKS CLEAR JOB/PROGRAM NAMES
         MVC   21(L'PIBLOGID+2,R6),=F'Ø' CLEAR COMMUNICATIONS ADDRESS.
         LA    R6,L'TAB(R6)        INCREMENT REG 6 TO NEXT POSITIONS.
         LA    R1,3(R1)            INCREMENT REG 1 TO NEXT POSITIONS.
         CLI   Ø(R1),X'FF'         ARE WE DONE.
         BNE   CKNXT               NO-BRANCH TO CKNXT.
         CLC   INPJOBN(L'INPJOBN+L'INPPGMN),BLANKS ARE JOB/PROGRAM NAME
         BNE   CKNXT5              NO-BRANCH TO CKNXT5.
         MVI   INPRCDE,C'7'        INDICATE JOB/PROGRAM FIELDS BLANK.
         B     CKRTN9              BRANCH TO CKRTN9.
CKNXT5   EQU   *
         L     R1,ASYSCOM          LOAD ADDRESS OF SYSTEM COMMUNICATION
         USING SYSCOM,R1           INFORM ASSEMBLER.
         LH    R5,IJBNPART         LOAD NPARTS VALUE TO REG 5. (X'2C').
         DROP  R1                  (SYSCOM).
         SR    R6,R6               CLEAR REG 6.
         STH   R6,COUNT            SET COUNT TO ZERO.
         ST    R6,SVR6C
         L     R1,ACOMRG           LOAD ADDRESS OF COMMUNICATIONS REGIO
         USING COMREG,R1           INFORM ASSEMBLER.
         LH    R7,PIBPT            LOAD PIB ADDRESS TO REG 7. (X'5A').
         LH    R8,PIB2PTR          LOAD PIB2 ADDRESS TO REG 8. (X'7C').
         DROP  R1                  (COMREG).
         STM   R5,R8,SVREGS        SVE REGS 5 THRU 8.
         L     R5,SVR5D
         LA    R6,PCBSTAP-L'PCBSTAP LOAD ADDRESS OF PCBDYNP TO REG 6.
         ST    R6,SVR6X
CKNXT7   EQU   *
         MVI   INPRCDE,C'Ø'        INDICATE WE'VE FOUND NOTHING.
         LA    R9,INPPIDS          LOAD ADDRESS OF INPPIDS TO REG 9.
         CLC   INPJOBN,BLANKS      ARE WE CHECKING FOR PROGRAM NAME ONL
         BE    CKPRG               YES-BRANCH TO CKPRG.
         CLC   INPPGMN,BLANKS      ARE WE CHECKING FOR JOB NAME ONLY.
         BE    CKJOB               YES-BRANCH TO CKJOB.
```

Editor's note: this article will be continued next month.

Robert Botsis
Senior Systems Programmer (USA)

EXEC CICS LINK and the External CICS Interface

There is a manual dedicated to the External CICS Interface, but here is a simple outline example of what has been used in many environments. In this example, a batch job passes parameters on the JCL EXEC card to an MVS client batch program (SUNEXCI). The batch program issues an EXEC CICS LINK, with a COMMAREA, to a CICS/ESA 4.1 server program. One of the parameters on the JCL EXEC card is the CICS VTAM APPLID. Hence the batch program may issue the 'link' to any valid CICS region. The function of the CICS server program is not detailed here because it is the mechanism of the EXEC CICS LINK that is of interest.

The MVS client program, SUNEXCI, must be link-edited with the DFHXCSTB stub from the CICS410.SDHEXCI dataset. For example:

```
//*====================================================================
//*        LINKAGE EDITOR
//*====================================================================
//LKED      EXEC PGM=IEWL,REGION=1024K,COND=(4,LT,ASM),
//          PARM='XREF,AMODE=31,RMODE=ANY'
//SYSPRINT  DD   SYSOUT=*
//SYSLIB    DD   DSN=CICS410.LOADLIB,DISP=SHR
//          DD   DISP=SHR,DSN=CICS410.SDFHEXCI
//SYSUT1    DD   UNIT=SYSDA,DCB=BLKSIZE=1024,
//               SPACE=(1024,(200,20))
//SYSLIN    DD   DSN=&&LINKIN,DISP=(OLD,DELETE)
//          DD   DDNAME=SYSIN
//SYSLMOD   DD   DSN=CICS410.USERLOAD,DISP=SHR
//SYSIN     DD   *
 INCLUDE SYSLIB(DFHXCSTB)
 NAME SUNEXCI(R)
//
```

The MVS batch job JCL EXEC card for SUNEXCI is as follows:

```
//STEP1     EXEC PGM=SUNEXCI,PARM='ASUNCICS,READSUN1'
```

The External CICS Interface uses DFHIRP (the Interregion Communication Program) and therefore we require an MRO connection with session definitions in the CICS region to support the mechanism.

THE CONNECTION DEFINITION

On the connection definition, the EXCI option must be specified on the PROTOCOL attribute to indicate that the connection is for use by an MVS client program using the External CICS Interface. There is also the CONNTYPE attribute provided on the CONNECTION resource definition. For EXCI connections with programs using EXEC CICS LINK we should specify GENERIC. For example, our connection name is SUN1.

```
 OBJECT CHARACTERISTICS
   CEDA  View Connection( SUN1 )
 +  REMOTEName      :
    REMOTESYSNet    :
   CONNECTION PROPERTIES
    ACcessmethod        : IRc               Vtam | IRc | INdirect | Xm
    PRotocol            : Exci              Appc | Lu61 | Exci
    Conntype            : Generic           Generic | Specific
```

THE SESSION DEFINITION

For the session resource definition, we indicate on the PROTOCOL attribute that we are using the External CICS Interface. We must also leave the SENDcount and SENDPfx blank for EXCI sessions.

Our sessions and the significant attributes are as follows:

```
OBJECT CHARACTERISTICS
  CEDA  View Sessions(SUN1SESS )
   Sessions             : SUN1SESS
   Group                : SUN1GRP
   DEscription          :
  SESSION IDENTIFIERS
   Connection           : SUN1
   SESSName             :
   NETnameq             :
   MOdename             :
  SESSION PROPERTIES
   Protocol             : Exci              Appc | Lu61 | Exci
   MAximum              : 000 , 000         0-999
   RECEIVEPfx           : @
   RECEIVECount         : 005               1-999
   SENDPfx              :
   SENDCount            :                   1-999
   SENDSize             : 04096             1-30720
+  RECEIVESize          : 04096             1-30720
```

THE TRANSACTION DEFINITION

The EXEC CICS LINK call from the batch program must specify a TRANSID(name) parameter, where 'name' is the name of a transaction defined in the CICS/ESA 4.1 region. This transaction must specify a program name of DFHMIRS. For example, we use transaction SUNT:

```
OBJECT CHARACTERISTICS
  CEDA  View TRANSaction( SUNT )
   TRANSaction    : SUNT
   Group          : SUNTRAN
   DEscription    :
   PROGram        : DFHMIRS
```

THE PROGRAM

The section of the batch program (SUNEXCI) that issues the EXEC CICS LINK and the related data areas and constants are as follows:

```
                ..
                ..
*=====================================================================*
*   Retrieve the target CICS ApplId passed as a parameter on the JCL  *
*   EXEC card                                                         *
*=====================================================================*
*
START   DS    ØH
        L     R1,Ø(,R1)                       R1-->JCL Plist.
        XR    R7,R7                           Clear R7
        LH    R7,Ø(R1)                        PICK UP PARM LENGTH
        CH    R7,=H'17'                       check parameter length
        BL    EXIT                            EXIT if not long enough
        MVC   TARGET_SYSTEM(8),2(R1)          pick up CICS Applid
        MVC   COMM_CMD(8),11(R1)              pick up command
*
*=====================================================================*
*   Clear Working Storage;                                            *
*=====================================================================*
*
        LA    R9,EXEC_RETAREA                  R9--> EXEC return area
        USING EXCI_EXEC_RETURN_CODE,R9
        MVC   EXECEYE(8),=C'EXECRET:'          Eyecatcher
*
*=====================================================================*
*   Perform LINK request.                                             *
*=====================================================================*
*
        EXEC  CICS LINK PROGRAM(TARGET_PROGRAM)                        *
```

```
              TRANSID(TARGET_TRANSID)                                  *
              APPLID(TARGET_SYSTEM)                                    *
              COMMAREA(COMMAREA)                                       *
              LENGTH(EXEC_COM_LEN)                                     *
              DATALENGTH(EXEC_DAT_LEN)                                 *
              RETCODE(EXEC_RETAREA)                                    *
              SYNCONRETURN
*
*=======================================================================*
*   Did the call fail?  Check the Return Codes.                         *
*=======================================================================*
*
        CLC   EXEC_RESP,=F'Ø'
        BE    EXIT
                 ..
                 ..


...... data constants
                                          ..
                                          ..

*
TARGET_INFO     DS ØF                 Target Variable Constants.
TARGET_PROGRAM  DC CL8'SUNCOMM'       Server program name .
TARGET_TRANSID  DC CL4'SUNT'          Name of Target Transaction.
*
EXEC_INFO       DS ØF                 EXEC level specific information
EXEC_DAT_LEN    DC  AL2(12)           Outbound length(EXEC call)
EXEC_COM_LEN    DC  AL2(12)           Inbound length(EXEC call)

EXECEYE         DS 8C
EXEC_RETAREA    DS CL(EXCI_EXEC_RETURN_CODE__LEN)
                          ..
                          ..

......data areas
                                       ..
                                       ..
COMMAREA        DS ØF
COMM_CMD        DS CL8
COMM_RETCODE    DS F
COMMEND         EQU *
*
TARGET_SYSTEM   DS 8C                 Name of Target CICS system
```

Auto-install for programs – part 2

This month we continue the code for a program used to auto-install user-written programs and maps with the same simple attributes. This means the attributes of the CICS-supplied auto-install models.

```
*---     LOAD LANGUAGE + DISABLE  TABLE CICSAUPT   ONE TIME ONLY
C1000    DS    0H
         CLC   CWAAUPOT,=C'ERRO'       ERROR OCCURRED ALREADY
         BE    C1500                   YES, END OF LOAD ACTIONS
         CLC   CWAAUPOT,=C'LOAD'       IN LOAD STATUS
         BNE   C1050                   NO
         DS    0H                      YES, PREVENT FROM DOING
         DS    0H                      2 OR MORE LOADS
         DS    0H                      DUE TO REENTRANCY
         EXEC CICS DELAY INTERVAL (1)
         B     C1000                   NO
C1050    DS    0H
         L     R14,CWAAUPOT            ALREADY LOADED
         LTR   R14,R14                 TEST IT
         BZ    C1100                   NO
         B     C1500                   YES, END OF LOAD ACTIONS
C1100    DS    0H                      TRY TO LOAD
         MVC   CWAAUPOT,=C'LOAD'       SIGNAL LOAD STATUS
         EXEC CICS LOAD PROGRAM ('CICSAUPT') HOLD                      *
               SET (R15) ENTRY (R14) RESP (AUPTRESP)
         OC    AUPTRESP,AUPTRESP       ERROR
         BZ    C1200                   NO
         MVC   CWAAUPOT,=C'ERRO'       REMEMBER ERROR
         MVC   PROBBEG,=CL40'CICSAUPO SYSI PROBLEM CICSAUPT LOAD'
         BAS   R14,PROBWRIT            CONSOLE MESSAGE
         B     C1500                   GO TO END OF LOAD
C1200    DS    0H
         ST    R14,AUPTENTR            REMEMBER ENTRY POINT
         ST    R15,AUPTLOAD            REMEMBER LOAD POINT
         DS    0H                      CHECK HEADER.......
         CLC   0(8,R14),=CL8'CICSAUPT' EYECATCHER FITS...
         BE    C1300                   YES
         MVC   CWAAUPOT,=C'ERRO'       REMEMBER ERROR
         MVC   PROBBEG,=CL40'CICSAUPO SYSI PROBLEM CICSAUPT HEAD'
         BAS   R14,PROBWRIT            CONSOLE MESSAGE
         B     C1500                   GO TO END OF LOAD
C1300    DS    0H
         MVC   CWAAUPOT,AUPTENTR       REMEMBER ENTRY POINT
C1500    DS    0H                      END OF LOAD LANGUAGE TABLE
*---     ENTRY MESSAGE
C9000    DS    0H
```

```
         AP    RFENTRY,=P'1'           ADD  COUNTER
         BAS   R14,MESSFILL            DEBUG MESSAGE FILL
         MVC   MSGTYPE,=CL4'ENTR'      DEBUG MESSAGE FILL
         BAS   R14,MESSWRIT            DEBUG MESSAGE OUTPUT ENTRY
*---     SELECT WHICH RESOURCE IS TO BE INSTALLED                    *
E000     DS    0H
         MVC   LASTRESC,PGAC_PROGRAM   REMEMBER LAST RESOURCE COMING
         MVC   TASTRESC,EIBTIME        LAST RESOURCE INCOMING TIME
         CLI    PGAC_MODULE_TYPE,PGAC_TYPE_MAPSET
         BE     E010
         CLI    PGAC_MODULE_TYPE,PGAC_TYPE_PROGRAM
         BE     E020
         CLI    PGAC_MODULE_TYPE,PGAC_TYPE_PARTITIONSET
         BE     E030
         B      E090
*          M A P S
E010     DS    0H
         CLC   PGAC_PROGRAM(2),=C'MP'  APPLICATION MAP
         BNE   E012                    NO
         CLI   PGAC_PROGRAM+7,C' '     APPLICATION MAP
         BNE   E012                    NO
         AP    RFMAPS,=P'1'            COUNT FOR MAPS
         MVC   PGAC_MODEL_NAME,=CL8'DFHPGAMP'  MODEL FOR MAPS
         B     RETURNGO                RETURN GOOD
E012     DS    0H
         AP    RFBMAPS,=P'1'           COUNT FOR BAD MAP NAMES
         B     RETURNBA                RETURN BAD
*          P R O G R A M S
*           SUBFUNCTION:  FORCE ASRA
E020     DS    0H
         CLC   PGAC_PROGRAM,=C'HASSASRA' TO FORCE ASRA..
         DS    0H                      SAY CECI LOAD PROGRAM HASSASRA
         BE    E021                    ASRA WANTED, YES...
         CLC   PGAC_PROGRAM(5),=C'CPEXI' SYSTEM PROGRAM
         BE    E026                    YES, DO NOT INSTALL
         CLC   PGAC_PROGRAM(5),=C'CPTRU' SYSTEM PROGRAM
         BE    E026                    YES, DO NOT INSTALL
         DS    0H
         CLC   PGAC_PROGRAM(2),=C'CT'  APPLICATION PROGRAM
         BE    E022                    YES
         CLC   PGAC_PROGRAM(2),=C'CP'  APPLICATION PROGRAM
         BE    E022                    YES
         B     E026                    NO, DO NOT INSTALL
E021     DS    0H
         DC    H'0',C'FORCE ASRA'      FORCE ASRA
         DS    0H
E022     DS    0H
         AP    RFPROGS,=P'1'           COUNT FOR PROGRAMS
         DS    0H                      ADDRESS LANGUAGE TABLE
         DS    0H
```

```
         DS    ØH                       TABLE NOT FOUND: ASSEMBLER
         DS    ØH                       PROG  NOT FOUND: PL/1
         DS    ØH                       ALL LANGUAGES ARE CONTAINED
         DS    ØH                       IN CICSAUPT TABLE
         DS    ØH                       BUT NOT NORMALLY PL/1
         DS    ØH                       PROG FOUND:
         DS    ØH                       LANGUAGE ASSEMBLR: ASSEMBLER
         DS    ØH                       LANGUAGE PL/1    : PL/1
         DS    ØH                       LANGUAGE PL/I    : PL/I
         DS    ØH                       LANGUAGE OTHER   : BAD PROGRAM
         DS    ØH                       RESOURCE NOT INSTALLED
         DS    ØH                       CICS CORRECTS WRONG DECIDED
         DS    ØH                       LANGUAGE TO THE CORRECT ONE
         DS    ØH                       BEFORE PROGRAM EXECUTION
         DS    ØH
         CLC   CWAAUPOT,=C'ERRO'        ERROR DURING LOAD
         BE    EØ24                     YES, DEFAULT PL/1
         L     R14,CWAAUPOT             ADDRESS HEADER OF AUPT TABLE
         USING AUPTDSEC,R14
         L     R1,AUPTLENG              GET ELEMENT LENGTH
         AR    R14,R1                   ADDRESS ADDRESS OF PROGRAMS
         L     R15,AUPTPOIN             ADDRESS TO FIRST PROGRAM
         DROP  R14
         USING AUPTDSEC,R15
EØ22B    DS    ØH
         CLC   AUPTPROG,=XL8'FFFFFFFFFFFFFFFF'  END OF TABLE
         BE    EØ24                     YES, DEFAULT PL/1
         CLC   PGAC_PROGRAM,AUPTPROG    SEARCH PROGRAM IN TABLE
         BE    EØ22E                    FOUND
         AR    R15,R1                   ADDRESS NEXT ELEMENT
         B     EØ22B                    NO, SEARCH AGAIN
EØ22E    DS    ØH
         MVC   ASGDISAB,BLANK8Ø         SIGNAL NO DISABLE IN MESSAGE
         CLC   AUPTDISA,=C'DISA'        IS PROGRAM DISABLED
         BE    EØ22F                    YES, DO NOT INSTALL
         B     EØ22H                    NO
EØ22F    DS    ØH
         MVC   ASGDISAB,=C'DISAB'       SIGNAL DISABLE IN MESSAGE
         B     RETURNBA                 RETURN BAD DUE TO DISABLE
EØ22H    DS    ØH
         CLC   AUPTLANG,=CL4'ASSE'      ASK FOR ASSEMBLER
         BE    EØ23                     YES
         CLC   AUPTLANG,=CL4'PL/1'      ASK FOR PL/1
         BE    EØ24                     YES
         CLC   AUPTLANG,=CL4'PL/I'      ASK FOR PL/I
         BE    EØ24                     YES
         B     EØ26                     BAD LANGUAGE FOR AUTOINSTALL
         DROP  R15
EØ23     DS    ØH                       ASSEMBLER
         DS    ØH
```

```
         AP    RFAPROGS,=P'1'          COUNT FOR  ASSEMBLER
         MVC   PGAC_MODEL_NAME,=CL8'DFHPGAPG'  MODEL FOR ASSEMBLER
         B     RETURNGO                RETURN GOOD
E024     DS    ØH                      PL/1
         DS    ØH
         AP    RFPPROGS,=P'1'          COUNT FOR  PL/1
         MVC   PGAC_MODEL_NAME,=CL8'DFHPGALX'  MODEL FOR PL/1
         B     RETURNGO                RETURN GOOD
E026     DS    ØH
         AP    RFBPROGS,=P'1'          COUNT FOR BAD PROGRAM NAMES
         B     RETURNBA                RETURN BAD
*          P A R T I T I O N S E T S
E03Ø     DS    ØH
         AP    RFPARTS,=P'1'           COUNT FOR PARTITION SETS
         MVC   PGAC_MODEL_NAME,=CL8'DFHPGAPT'  MODEL FOR PART SETS
         DS    ØH                       FORCE BAD INSTALL...
         MVC   PGAC_MODEL_NAME,=CL8'DFHPG...'  MODEL FOR PART SETS
         B     RETURNBA                RETURN BAD
*          O T H E R S
E09Ø     DS    ØH
         AP    RFOTHERS,=P'1'          COUNT FOR OTHER RESOURCES
         B     RETURNBA                RETURN BAD
E99Ø     DS    ØH
*---     RETURN BAD:  RETURN CODE BAD                                  *
RETURNBA DS    ØH
         MVI   PGAC_RETURN_CODE,PGAC_RETURN_DONT_DEFINE_PROGRAM
         B     RETURN9Ø
*---     RETURN GOOD                                                ---*
*         COLLECT INSTALLED RESOURCE IN TABLE COLLTABL  WRAPAROUND    *
*          IF TABLE IS FULL: START TRAN AUPO WITH TABLE DATA          *
*           START COLLECT AT TABLE BEGIN AGAIN                        *
*          IF PROGRAM CTAUPOZZ ENTERS, FROM PLTSD PROGRAM INITIATED   *
*           THIS IS PROBABLY THE LAST PROGRAM WHICH ENTERS.........   *
*            START TRAN AUPO WITH TABLE DATA                          *
*---      RETURN GOOD:  RETURN CODE GOOD                            ---*
RETURNGO DS    ØH
RETURNØ5 DS    ØH
         CP    COLLCNTR,=P'Ø'          IS START RUNNING
         BE    RETURNØ8                NO
         DS    ØH                      FILL TABLE IN SINGLE TASKING
         DS    ØH                      NOT DURING START AUPO
         AP    COLLDELA,=P'1'          COUNT DELAYS
         EXEC CICS DELAY INTERVAL (1)
         B     RETURNØ5                NO
RETURNØ8 DS    ØH
         L     R12,COLLPOIN            NEXT EVTL EMPTY ELEMENT
         CLC   COLLRESO,COLLTABE       IS ELEMENT THE END OF TABLE
         BE    RETURN3Ø                TABLE IS FULL..
RETURN2Ø DS    ØH
         CLC   PGAC_PROGRAM,=CL8'CTAUPOZZ'  ENTRY FROM CICSFINAL
```

```
         BNE   RETURN22               NO
         OI    COLLCNTL,COLLB8Ø       YES
         OI    COLLCNTL,COLLBØ8       YES
RETURN22 DS    ØH
         MVC   COLLRESO,PGAC_PROGRAM  INSERT RESOURCE IN TABLE
         L     R14,COLLLENG
         AH    R14,=AL2(COLLDSEE-COLLDSEB) NUMBER OF BYTES FOR START
         ST    R14,COLLLENG
         AH    R12,=AL2(COLLDSEE-COLLDSEB) ADDRESS NEXT ELEMENT
         CLC   COLLRESO,COLLTABE      IS ELEMENT THE END OF TABLE
         BE    RETURN25               TABLE IS FULL..
         MVI   COLLRESO,X'ØØ'         FORCE NEXT ELEMENT TO EMPTY
RETURN25 DS    ØH
         ST    R12,COLLPOIN           ADDRESS NEXT ELEMENT
         B     RETURN85               ALL DONE FOR THIS ELEMENT
RETURN3Ø DS    ØH                     TABLE IS FULL
         MVC   PROBBEG,=CL4Ø'CICSAUPO SYSI INFO: COLLTABL VOLL'
         BAS   R14,PROBWRIT           CONSOLE MESSAGE
         OI    COLLCNTL,COLLB2Ø       START RUNNING REASON FULL
         AP    COLLNUSF,=P'1'         COUNT IT
         BAS   R14,STARTASK           START TASK AUPO WITH COLLECTED
         NI    COLLCNTL,255-COLLB2Ø   START RUNNING REASON FULL END
         DS    ØH                     DATA
         LA    R12,COLLTABL           ADDRESS BEGIN OF TABLE
         ST    R12,COLLPOIN           ADDRESS NEXT ELEMENT
         MVI   COLLRESO,X'ØØ'         FORCE FIRST ELEMENT TO EMPTY
         B     RETURNØ5               DO NOW FOR THIS ELEMENT
RETURN85 DS    ØH
         TM    COLLCNTL,COLLBØ8       ENTRY FROM CTAUPOZZ PLTSD
         BZ    RETURN86
         MVC   PROBBEG,=CL4Ø'CICSAUPO SYSI INFO: CTAUPOZZ ENTRY'
         BAS   R14,PROBWRIT           CONSOLE MESSAGE
         DS    ØH                     START ONLY OR LAST AUPO TASK
         OI    COLLCNTL,COLLB1Ø       START RUNNING REASON CTAUPOZZ
         AP    COLLNUSZ,=P'1'         COUNT IT
         BAS   R14,STARTASK           START TASK AUPO WITH COLLECTED
         NI    COLLCNTL,255-COLLB1Ø   START RUNNING REASON CTAUPOZZ
         NI    COLLCNTL,255-COLLBØ8   CTAUPOZZ END
RETURN86 DS    ØH
         MVC   LASTRESI,PGAC_PROGRAM  REMEMBER LAST RESOURCE INSTALLD
         MVC   TASTRESI,EIBTIME       REMEMBER LAST RESOURCE INST TIME
         MVC   LASTMODE,PGAC_MODEL_NAME  REMEMBER LAST MODEL USED
         MVI   PGAC_RETURN_CODE,PGAC_RETURN_OK
         AP    RFINSTAL,=P'1'         COUNT FOR INSTALLS
         B     RETURN9Ø
*---     PROGRAM'S END                                          ---*
RETURN9Ø DS    ØH
         BAS   R14,MESSFILL           DEBUG MESSAGE FILL
         MVC   MSGTYPE,=CL4'EXIT'     DEBUG MESSAGE FILL
         BAS   R14,MESSWRIT           DEBUG MESSAGE OUTPUT   EXIT
```

```
         DS    ØH                       TYPE PROGRAM
         EXEC CICS RETURN
*---     START TASK AUPO WITH COLLECTED AUTOINSTALL RESOURCES
STARTASK ST    R14,STARSAVE
         MVC   COLLTIME,EIBTIME         EIBTIME
         MVC   COLLDATE,EIBDATE         EIBDATE
         AP    COLLNUST,=P'1'           NUMBER OF STARTS
         OI    COLLCNTL,COLLB4Ø         START RUNNING
         AP    COLLCNTR,=P'1'           START RUNNING
         L     R14,COLLLENG             LENGTH PLUS ONE ENTRY
         AH    R14,=AL2(COLLDSEE-COLLDSEB) NUMBER OF BYTES FOR START
         ST    R14,COLLLENG
         EXEC CICS START TRANSID ('AUPO') FROM (COLLDATA)              X
              REQID ('CICSAUPO') INTERVAL (Ø)                          X
              LENGTH (COLLLEN2)  RESP (STARRESP)
         OC    STARRESP,STARRESP
         BZ    STARTAS1
         MVC   PROBBEG,=CL4Ø'CICSAUPO SYSI PROBLEM START AUPO '
         BAS   R14,PROBWRIT             CONSOLE MESSAGE
STARTAS1 DS    ØH
         MVC   COLLLENG,=AL4(COLLTABL-COLLDATA) LENGTH OF START DATA
         NI    COLLCNTL,255-COLLB4Ø     START RUNNING ENDED
         SP    COLLCNTR,=P'1'           START RUNNING ENDED
         L     R14,STARSAVE
         BR    R14
*---     FILL DEBUG MESSAGE
MESSFILL DS    ØH
         ST    R14,MESSSAVE
         MVI   MESSBEG,C' '             BLANK MESSAGE COMPLETELY
         MVC   MESSBEG+1(MESSEND-MESSBEG-1),MESSBEG
         UNPK  MSGTASKN,EIBTASKN        TASK NUMBER
         OI    MSGTASKN+L'MSGTASKN-1,X'FØ'
         MVC   MSGTRNID,EIBTRNID        TRANID
         MVC   MSGTRMID,EIBTRMID        TERMID
         OC    MSGTRMID,MSGTRMID        TERMID
         BNZ   *+1Ø
         MVC   MSGTRMID,=CL4'----'
         MVC   MSGPROG,ASGPROG          THIS PROGRAMS NAME
         MVC   MSGSYSI,ASGSYSI          THIS REGIONS SYSID
         MVC   MSGDISAB,ASGDISAB        PROGRAM IS DISABLED, NO INSTALL
         UNPK  MSGENTRY,RFENTRY         ENTRY NUMBER
         OI    MSGENTRY+L'MSGENTRY-1,X'FØ'
         MVC   MSGPROGR,PGAC_PROGRAM    MODUL NAME
         CLI   MSGPROGR,C' '
         BNE   *+8
         MVI   MSGPROGR,C'-'
         MVC   MSGMODUT,PGAC_MODULE_TYPE MODULE TYPE
         CLI   MSGMODUT,C' '
         BNE   *+8
         MVI   MSGMODUT,C'-'
```

```
MVC   MSGMODEN,PGAC_MODEL_NAME  MODEL NAME
CLI   MSGMODEN,C' '
BNE   *+10
MVC   MSGMODEN,=CL8'--------'
MVC   MSGLANGU,PGAC_LANGUAGE  LANGUAGE
CLI   MSGLANGU,C' '
BNE   *+8
MVI   MSGLANGU,C'-'
MVC   MSGCEDFS,PGAC_CEDF_STATUS CEDF STATUS
CLI   MSGCEDFS,C' '
BNE   *+8
MVI   MSGCEDFS,C'-'
MVC   MSGDATAL,PGAC_DATA_LOCATION DATA LOCATION
CLI   MSGDATAL,C' '
BNE   *+8
MVI   MSGDATAL,C'-'
MVC   MSGEXECK,PGAC_EXECUTION_KEY EXECUTION KEY
CLI   MSGEXECK,C' '
BNE   *+8
MVI   MSGEXECK,C'-'
MVC   MSGLOADA,PGAC_LOAD_ATTRIBUTE LOAD ATTRIBUTE
CLI   MSGLOADA,C' '
BNE   *+8
MVI   MSGLOADA,C'-'
MVC   MSGUSELP,PGAC_USE_LPA_COPY USE LPA COPY
CLI   MSGUSELP,C' '
BNE   *+8
MVI   MSGUSELP,C'-'
MVC   MSGEXECS,PGAC_EXECUTION_SET EXECUTION SET
CLI   MSGEXECS,C' '
BNE   *+8
MVI   MSGEXECS,C'-'
MVC   MSGREMOS,PGAC_REMOTE_SYSID REMOTE SYSID
CLI   MSGREMOS,C' '
BNE   *+10
MVC   MSGREMOS,=CL4'----'
MVC   MSGREMOP,PGAC_REMOTE_PROGID REMOTE PROGID
CLI   MSGREMOP,C' '
BNE   *+10
MVC   MSGREMOP,=CL8'--------'
MVC   MSGREMOT,PGAC_REMOTE_TRANSID REMOTE TRANSID
CLI   MSGREMOT,C' '
BNE   *+10
MVC   MSGREMOT,=CL4'----'
MVC   MSGRETUT,=C'RC='
MVC   MSGRETUR,PGAC_RETURN_CODE RETURN CODE
CLI   MSGRETUR,C' '
BNE   *+8
MVI   MSGRETUR,C'-'
L     R14,MESSSAVE
```

```
         BR    R14
*---     DEBUG MESSAGE OUTPUT
MESSWRIT DS    ØH
         ST    R14,MESSSAVE
         CLI    PGAC_MODULE_TYPE,PGAC_TYPE_PROGRAM
         BE     MESSWR1Ø
         CLI    PGAC_MODULE_TYPE,PGAC_TYPE_MAPSET
         BE     MESSWR2Ø
         B      MESSWRØ5
MESSWRØ5 DS    ØH                     TYPE PROGRAM
         CP    RFOTHERS,=P'26'        MESSAGE LIMIT REACHED
         BL    MESSWR8Ø               NO
         B     MESSWR3Ø               YES
MESSWR1Ø DS    ØH                     TYPE PROGRAM
         CP    RFPROGS,=P'26'         MESSAGE LIMIT REACHED
         BL    MESSWR8Ø               NO
         B     MESSWR3Ø               YES
MESSWR2Ø DS    ØH                     TYPE MAPS
         CP    RFMAPS,=P'26'          MESSAGE LIMIT REACHED
         BL    MESSWR8Ø               NO
         B     MESSWR3Ø               YES
MESSWR3Ø DS    ØH                     GUARANTEE FOR MESSAGE PAIRS
         CLI   SWMESSPA,1             1. MESSAGE DONE
         BE    MESSWR8Ø               YES, DO 2.
         B     MESSWR9Ø                NO
MESSWR8Ø DS    ØH                     TYPE PROGRAM
         MVI   SWMESSPA,1             GUARANTEE FOR MESSAGE PAIRS
         DS    ØH
         XC    MESSBYTO,MESSBYTO      MESSAGE LENGTH WTO NULL
         MVC   MESSBYTE,MESSLENG      MESSAGE LENGTH WTO AND TD
         EXEC CICS WRITEQ TD QUEUE ('CSPL') FROM (MESSBEG)            X
               LENGTH (MESSBYTE)   RESP (WRITRESP)
         OC    WRITRESP,WRITRESP
         BNZ   ERRORØ6
         EXEC CICS WRITE OPERATOR TEXT (MESSBEG) TEXTLENGTH (MESSBYTO) X
                                  RESP (OPERRESP)
         OC    OPERRESP,OPERRESP
         BNZ   ERRORØ4
MESSWR9Ø DS    ØH
MESSENDE L     R14,MESSSAVE
         BR    R14
MESSLENG DC    AL2(MESSEND-MESSBEG)    LENGTH WRITEQ TD
*---     WRITE PROBLEM MESSSAGE TO CONSOLE                            *
* FORMAT: VARIABLE PROBBEG,=CL4Ø'CICSAUPO SYSI PROBLEM COLLTABL FULL'
*                                +123456789A123456789B
*                                         .... FOR SYSID TO INSERT
PROBWRIT DS    ØH
         ST    R14,PROBSAVE
         L     R14,CWAAUPOS             COUNTER EXISTS..
         LTR   R14,R14
```

```
         BZ    *+10                      NO, DO NOT ADD
         AP    RFPROBLE,=P'1'            COUNT PROBLEMS
         XC    PROBBYTO,PROBBYTO         MESSAGE LENGTH
         MVC   PROBBYTE,=AL2(L'PROBBEG)  MESSAGE LENGTH
         MVC   PROBBEG+09(L'ASGSYSI),ASGSYSI  SYSID IN MESSAGE
         EXEC CICS WRITE OPERATOR TEXT (PROBBEG) TEXTLENGTH (PROBBYTO) X
                                RESP (PROBRESP)
         OC    PROBRESP,PROBRESP
         BNZ   ERROR05
         DS    0H                        TYPE PROGRAM
PROBENDE L     R14,PROBSAVE
         BR    R14
*---     DEFINE CONSTANTS                                          ---*
BLANK80  DC    CL80' '                   80 BLANKS
*        ERRORS WITH ABEND
*        THESE ERRORS WILL NEVER OCCUR..
ERROR01  DC    H'00',H'01',C'CICSAUPO COMMAREA LENGTH WRONG 1'
ERROR02  DC    H'00',H'02',C'CICSAUPO COMMAREA LENGTH WRONG 2'
ERROR03  DC    H'00',H'03',C'CICSAUPO GETMAIN USER PROBLEM'
ERROR04  DC    H'00',H'04',C'CICSAUPO WRITE OPERATOR ERROR'
ERROR05  DC    H'00',H'05',C'CICSAUPO WRITE OPERATOR PROBLEM ERROR'
ERROR06  DC    H'00',H'05',C'CICSAUPO WRITEQ TD ERROR'
*---     LTORG
         LTORG
*---     ABEND HANDLING
* CICS DOES: NO TRAN DUMP
*            ONLY SYSDUMP   (THIS MAY BE SUPPRESSED DUE TO OUR
*                            INSTALLATION)
*            NO ENTRY TO DFHPEP
*             THAT MEANS  NO MEMO INFO
*            DISACTIVATE THE AUTO INSTALL EXIT
*             THAT IS THE WORST....
* W WANT:    A TRAN DUMP (OR ONE PER AREA)
*            NO SYSDUMP
*            ENTRY TO DFHPEP.. INITIATES A MEMO INFO
*            THE ABEND ATTRIBUTES OF THE ORIGINAL ABEND (PSW  ETC.)
* 4 DUMPS ARE PRODUCED:
*            ABEND CODE AUP1    TRANSACTION
*             CONTAINES THE ABEND ATTRIBUTES...
*            ABEND CODE AUP2    GETMAINED USER STORAGE (COUNTER+TABLE)
*            ABEND CODE AUP3    CICSAUPT  LANGUAGE TABLE
*            ABEND CODE ASRA    FORCED ASRA TO END THE TASK
* REGISTERS   ON ENTRY...
*              R15   ABEND LABEL
*              R0-14 CONTENTS OF LAST CICS SERVICE REQUEST
ABEND    DS   0H
         BASR  R15,0
         USING *,R15
         L     R3,ACSECT
         L     R4,ACSECT
```

```
         LA    R4,2Ø48(R4)
         LA    R4,2Ø48(R4)
         B     ABEND1
ACSECT   DC    A(CICSAUPO)             CSECT ADDRESS
         DROP  R15
ABEND1   DS    ØH
         EXEC CICS HANDLE ABEND CANCEL RESP (ABENRESP)
*   INSERT ABEND ATTRIBUTES TEXTS
         MVC   ABENTOAB,=CL12'ORGABCODE...'
         MVC   ABENFOAB,=CL4' '
         MVC   ABENTABC,=CLØ8'ABCODE..'
         MVC   ABENFABC,=CL4' '
         MVC   ABENTPRO,=CLØ8'ABPROGRAM'
         MVC   ABENFPRO,=CL4' '
         MVC   ABENTKEY,=CLØ8'ASRAKEY.'
         MVC   ABENFKEY,=CL4' '
         MVC   ABENTINT,=CL12'ASRAINTRPT..'
         MVC   ABENFINT,=CL4' '
         MVC   ABENTPSW,=CLØ8'ASRAPSW.'
         MVC   ABENFPSW,=CL4' '
         MVC   ABENTREG,=CLØ8'ASRAREGS'
         MVC   ABENFREG,=CL4' '
*   RE-ADDRESS CWA
         EXEC CICS ADDRESS CWA (CWAREG)
*   INSERT ABEND ATTRIBUTES
         EXEC CICS ASSIGN ORGABCODE (ABENDOAB)                          *
                          ABCODE    (ABENDABC)                          *
                          ABPROGRAM (ABENDPRO)                          *
                          ASRAKEY   (ABENDKEY)                          *
                          ASRAINTRPT(ABENDINT)                          *
                          ASRAPSW   (ABENDPSW)                          *
                          ASRAREGS  (ABENDREG)                          *
                          RESP      (ABENRESP)
*   SEND PROBLEM MESSAGE
         MVC   PROBBEG,=CL4Ø'CICSAUPO SYSI PROBLEM * ABEND *'
         BAS   R14,PROBWRIT             CONSOLE MESSAGE
*   LINK TO DFHPEP       START MEMO TO SYSPROG
         EXEC CICS LINK PROGRAM ('DFHPEP#2') RESP (ABENRESP)
*   DUMP TRANSACTION
         EXEC CICS DUMP TRANSACTION DUMPCODE ('AUP1') RESP (ABENRESP)
*   DUMP USER GETMAIN
         L     R5,CWAAUPOS              STORAGE GETMAINED
         LTR   R5,R5
         BZ    ABEND2Ø                  NO
         LR    R6,R5                    2.REGISTER FOR GETMAINED STORAGE
         LA    R6,2Ø48(R6)              2.REGISTER FOR GETMAINED STORAGE
         LA    R6,2Ø48(R6)              2.REGISTER FOR GETMAINED STORAGE
         MVC   SEGNUMS1,=F'1'           NUMBER OF SEGMENTS
         MVC   SEGADDR1,CWAAUPOS        ADDRESS OF SEGMENT
```

```
         MVC   SEGLENG1,=A(USERSTOE-USERSTOB) LENGTH  OF SEGMENT
         EXEC CICS DUMP TRANSACTION DUMPCODE ('AUP2') RESP (ABENRESP)  *
               SEGMENTLIST (SEGADDR1)                                  *
               LENGTHLIST  (SEGLENG1)                                  *
               NUMSEGMENTS (SEGNUMS1)
ABEND2Ø  DS    ØH
*   DUMP PROGRAM CICSAUPT TABLE
         CLC   CWAAUPOT,=C'ERRO'            CICSAUPT LOADED
         BE    ABEND3Ø                      NO
         L     R14,CWAAUPOT                 STORAGE GETMAINED
         LTR   R14,R14
         BZ    ABEND3Ø                      NO
         EXEC CICS INQUIRE PROGRAM ('CICSAUPT') RESP (ABENRESP)        *
               LENGTH (INQULENG)
         OC    ABENRESP,ABENRESP            ERROR...
         BNZ   ABEND3Ø                      YES
         CLC   INQULENG,=F'-1'              REMOTE (A JOKE..)
         BE    ABEND3Ø                      YES
         MVC   SEGNUMS1,=F'1'               NUMBER OF SEGMENTS
         MVC   SEGADDR1,CWAAUPOT            ADDRESS OF SEGMENT
         MVC   SEGLENG1,INQULENG            LENGTH  OF SEGMENT
         EXEC CICS DUMP TRANSACTION DUMPCODE ('AUP3') RESP (ABENRESP)  *
               SEGMENTLIST (SEGADDR1)                                  *
               LENGTHLIST  (SEGLENG1)                                  *
               NUMSEGMENTS (SEGNUMS1)
ABEND3Ø  DS    ØH
*   ABEND..    CICS DEACTIVATES NOW THE PROGRAM
         DC    H'Ø',C'CICSAUPO ABEND AFTER ABEND'
*   CONSTANTS AND LTORG FOR ABEND HANDLING
         LTORG
         END   CICSAUPO
```

CICSAUPT

```
*ASM CICS(NOEPILOG NOPROLOG)
AUPT     TITLE 'AUTOINSTALL FOR PROGRAMS AND MAPS   LANGUAGE + DISABLE *
               TABLE'
CICSAUPT CSECT
*   HEADER    LOCATION DEPENDANT CODE
         DC    CL8'CICSAUPT',AL4(PROG2-PROG1),CL4'DUMY'
         DC    A(PROG3)      ADDRESS OF FIRST PROGRAM
         DC    A(Ø)
         DC    A(Ø)
         DC    A(Ø)
*   PROGRAM ATTRIBUTES
CICSAUPT AMODE 31
CICSAUPT RMODE ANY
         DC    CL2Ø'***I AM CICSAUPT****'    EYECATCHER
*   ALIGNEMENT FOR BETTER VIEWING
```

```
         DS    D
*   TABLE ELEMENT LENGTH DEFINITION
PROG1    DC    CL8'ABCDEFG1',CL4'DUMY',CL4'EMPT'
PROG2    DC    CL8'ABCDEFG2',CL4'DUMY',CL4'EMPT'
*   FIRST PROGRAM
PROG3    DC    ØCL8' ',ØCL4' ',ØCL4' '
*   SAMPLE FOR DISABLE
PROG4    DC    CL8'CPAUPO96',CL4'ASSE',CL4'DISA'
PROG5    DC    CL8'CPAUPO97',CL4'PL/I',CL4'DISA'
*   THE LANGUAGE TABLE     DAILY REFRESHED
*    ALL NOT MENTIONED PROGRAMS ARE TREATED AS PL/1 PROGRAMS
         DC    CL8'CPADEBLA',CL4'ASSE',CL4'EMPT'
         DC    CL8'CPADEDSP',CL4'ASSE',CL4'EMPT'
         DC    CL8'CTADEKØØ',CL4'ASSE',CL4'EMPT'
         DC    CL8'CTADEOØØ',CL4'ASSE',CL4'EMPT'
*        DC    CL8'CTMIBUØØ',CL4'PL/I',CL4'EMPT'
*        DC    CL8'CTMIPSØØ',CL4'PL/I',CL4'EMPT'
         DC    XL8'FFFFFFFFFFFFFFFF'   END MARKER 1
         DC    XL8'FFFFFFFFFFFFFFFF'   END MARKER 2
         END   CICSAUPT
```

Günther Hassold
INA Werk Schaeffler (Germany)

Call for papers

Why not share your expertise and earn money at the same time? *CICS Update* is looking for JCL, macros, program code, etc, that experienced CICS users have written to make their life, or the lives of their users, easier. We will publish it (after vetting by our expert panel) and send you a cheque when the article is published. Articles can be of any length and can be sent or e-mailed to Robert Burgess at any of the addresses shown on page 2. Why not call now for a free copy of our *Notes for contributors*?

Transient data output management – revisited

This article was inspired by an article in *CICS Update: Managing CICS transient data output* by Hein Vandenabeele (October 1997, page 36).

We also had problems with very large log files (especially MSGUSR). To address this problem we added 28 DD statements (because we stop and start our CICS regions every 4 weeks) for the MSGUSR output file.

```
//MSGUSR    DD SYSOUT=X,FREE=CLOSE,SPIN=UNALLOC
//MSGUSR    DD SYSOUT=X,FREE=CLOSE,SPIN=UNALLOC
//MSGUSR    DD SYSOUT=X,FREE=CLOSE,SPIN=UNALLOC
//MSGUSR    DD SYSOUT=X,FREE=CLOSE,SPIN=UNALLOC
//MSGUSR    DD SYSOUT=X,FREE=CLOSE,SPIN=UNALLOC
//MSGUSR    DD SYSOUT=X,FREE=CLOSE,SPIN=UNALLOC
//MSGUSR    DD SYSOUT=X,FREE=CLOSE,SPIN=UNALLOC
//MSGUSR    DD SYSOUT=X,FREE=CLOSE,SPIN=UNALLOC etc. etc.
```

Then we close and open the CSSL TD queue every day at 24.00 hours. This is carried out by an automatically scheduled job from OPC. This will switch the destination from the TD queue to the next MSGUSR DD statement.

In this way the MSGUSR output file for a specific day can be selected.

Paul Jansen
Systems Programmer
Interpay (The Netherlands)

IBM has released CICSPlex System Manager Version 1.3. This includes Business Application Services aimed at simplifying the definition, installation, movement, cloning, activation, and backout of CICS systems through development, testing, and operation.

IBM has also announced Version 2 of its eNetwork Host On-Demand. Its Java software gives consistent one-click access to host data from PCs, network computers, and advanced workstations.

The software emulates CICS procedures, 3270, 5250, VT52, VT100, and VT220. Version 2 has multiple concurrent sessions with one or more hosts; no end-user installation; concurrent Web surfing; and authentication and encryption.

For further information contact your local IBM representative.

* * *

Sterling Software has released Version 6.0 of its Windows-based Vision:Inspect analysis tool for COBOL applications, along with an add-on for PL/I support. New features include an impact analysis tool, an automatic component locator, split screen editor, and a better cacheing system.

The automatic component locator means that, during initial setup, the administrator can specify a subset of modules, such as CICS tables or JCL, and use the locator to drill down, find and load all required components automatically. It's designed to make the initial setup of a project easier and faster by eliminating the need to add all the components manually.

For further information contact:
Sterling Software, 1800 Alexander Bell Drive, Reston, VA 22091, USA.
Tel: (703) 264 8000.
Sterling Software, 1 Longwalk Road, Stockley Park, Uxbridge, Middlesex, UB11 1DB.
Tel: (0181) 867 8000.

* * *

IBM's Transarc has announced its TXSeries transactional middleware bundle, aimed at speeding the development of transaction-intensive electronic-business and enterprise applications. This includes Version 2.5 of Encina for NT (See *CICS Update* 145).

The Unix and Windows NT package consists of IBM's CICS 2.1.2 mainframe query system and Transarc's Encina 2.5 TP monitors; IBM's DSSeries Distributed Computing Environment-based security, single sign-on, and directory services; MQSeries asynchronous messaging middleware; and Lotus Domino Go Web server. DE-Light CIC Internet gateway and CICS gateway for Java will also be included. IBM currently provides support for the Orbix CORBA ORB from Iona Technologies, however, in a future release it will bundle its own Component Broker ORB and services with TXSeries, but will retain support for Orbix.

For further information contact:
Transarc, 707 Grant Street, Gulf Tower, 20th Floor, Pittsburgh, PA 15219, USA.