173 CICS update

April 2000

In this issue

CICS Update

Published by

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

North American office

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

Contributions

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

CICS Update on-line

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

Editor

Trevor Eddolls

Disclaimer

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

Subscriptions and back-issues

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs £175.00 in the UK; $270.00 in the USA and Canada; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1994 issue, are available separately to subscribers for £16.00 ($23.50) each including postage.

Printed in England.

Practical use for EXEC CICS CREATE services

INTRODUCTION

IBM added the EXEC CICS CREATE (and related commands) to the SPI for OS/390 Transaction Server, and has now retrofitted them to CICS 4.1. They enable systems programmers to do in programs what they used to have to do using CECI DEFINE and INSTALL. This means that if there is something you need to do with file definitions on a regular basis, you can now write a transaction to do it automatically.

For example, at our site, we have several CICS files that are accessed remotely, ie they are owned by a different region from the one in which the file read is issued. In the test environment this can sometimes be a problem because the file owning region is not always available – meaning that any attempt to read the file cannot succeed. At these times, it is useful to have the file defined as local to that region instead of remote. My example program (below) discards the existing file definition (WI10U in our case), then creates a new local definition using the EXEC CICS CREATE command. This means that the developers can change the file definition as and when they need to, and the systems programmer does not have to discard the old file definition and go in to CECI to install the new file definition every time it needs to be changed.

Once the basic principle has been understood, it is easy to give the user other options. In my example they can decide which file to allocate if the default is not wanted, and they can reinstall the remote definition if required. Other options could be added as required.

Note that the CICS translator option ‘SP’ is required.

AUTOTEXT

I read the article *Selecting the appropriate data location* in Issue 168 (November 1999) of *CICS Update* with interest. I have been using the

autoinstall process for some time, but the program in that article addressed some of the problems I was having.

Having used the program, I was having problems with the execution key of the program not matching that of the transaction, giving AEZ*x* abends.

Adding the following code after the label 'OMEG' seems to relieve this problem by checking the transaction definition and setting the program definition so that the two attributes match.

```
         EXEC CICS INQUIRE TRANSACTION(EIBTRNID)                    X
               TASKDATAKEY(TASKKEY)
         CLC   TASKKEY,DFHVALUE(CICSDATAKEY)
         BNE   USERKEY
CICSKEY  MVI   PGAC_EXECUTION_KEY,PGAC_CICS_KEY  SET EXECKEY CICS   */
         B     INIT01                  /* SKIP EXECKEY USER        */
USERKEY  MVI   PGAC_EXECUTION_KEY,PGAC_USER_KEY  SET EXECKEY USER   */
INIT01   MVI   EPGAI2API,X'00'         /*         INITIALIZE ...    */
```

BWWI1001

```
*ASM XOPTS(PROLOG,NOEPILOG,SP)
         TITLE 'BWWI1001 - TECHNICAL SERVICES MAKE WI10 LOCAL'
*********************************************************************
* BWWI1001:                                                         *
* --------                                                          *
*                                                                   *
* DESCRIPTION:                                                      *
*   - UTILITY TO CHANGE WI10 FROM REMOTE TO LOCAL OR VICE VERSA     *
*                                                                   *
* ATTRIBUTES:                                                       *
*   - HIGH-LEVEL ASSEMBLER                                          *
*   - COMMAND LEVEL CICS/ESA V4.1.0                                 *
*   - CICS TRANSLATOR OPTION "SP" IS REQUIRED                       *
*   - EXECKEY(CICS)                                                 *
*   - TASKDATAKEY(CICS)                                             *
*   - 31 BIT ADDRESSING                                             *
*                                                                   *
* PROCESSING:                                                       *
*   - DISCARD CURRENT WI10 DEFINITION                               *
*   - IF JUST WI10 INPUT DISPLAY HELP SCREEN                        *
*   - IF WI10 R  INPUT, CREATE NEW DEFINITION AS REMOTE             *
*   - IF WI10 L  INPUT CREATE NEW DEFINITION AS LOCAL WITH DSNAME   *
*     OF YINV1.WI10.£IAM                                            *
*   - IF WI10 L FILENAME.MSTR INPUT                                 *
```

```
*     CREATE NEW DEFINITION AS LOCAL WITH DSNAME OF FILENAME.MSTR    *
*   - RETURN TO CICS                                                 *
*                                                                    *
**********************************************************************
         TITLE 'BWWI1ØØ1 - DFHEISTG DSECT'
DFHEISTG DSECT
INPUT    DS    CL51
         ORG INPUT
TRANC    DS    CL4
         DS    CL1
OPTION   DS    CL1
         DS    CL1
CONNID   DS    CL4
         DS    CL4Ø
         ORG CONNID
NEWNAME  DS    CL44
RESP     DS    F
RESP2    DS    F
HW       DS    H
DW       DS    D
TEMP     DS    CL1
ATTR     DS    CL16Ø
OUTLINE  DS    CL79' '
ATTRCNT  DS    PL5
ZONED    DS    CL5
         TITLE 'BWWI1ØØ1 - REGISTER EQUATES'
RØ       EQU   Ø
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         TITLE 'BWWI1ØØ1 - CICS PROLOG CODE / ADDRESSING MODE'
BWWI1ØØ1 CSECT
BWWI1ØØ1 AMODE 31
BWWI1ØØ1 RMODE ANY
         MACRO
&ADDIT   ADDIT &STR
         LCLA  &ADD
```

```
&ADD     SETA  L'&STR
         MVC   Ø(L'&STR,R7),&STR          MOVE STRING TO ATTR
         SR    R6,R6                      CLEAR R6
         LA    R6,&ADD                    MOVE LENGTH OF STRING TO R6
         AR    R8,R6                      ADD STR LEN TO LENGTH OF ATTR
         AR    R7,R6                      BUMP POS IN ATTR BY STR LEN
         MEND
         TITLE 'BWWI1ØØ1 - MAINLINE CODE'
*********************************************************************
* BRANCH PAST PROGRAM NAME "EYECATCHER".                            *
*********************************************************************
MAIN     DS    ØH
         B     MAIN1ØØ
         DC    CL2'<<'
         DC    CL1' '
         DC    CL39'BRISTOL AND WEST PLC (TSG)'
         DC    CL1' '
         DC    CL8'&SYSDATE'
         DC    CL1' '
         DC    CL5'&SYSTIME'
         DC    CL1' '
         DC    CL4Ø'BWWI1ØØ1 - SET WI1ØU DEFINITION'
         DC    CL1' '
         DC    CL2'>>'
*********************************************************************
* MAINLINE CODE                                                     *
*********************************************************************
MAIN1ØØ  DS    ØH
*
         MVC   INPUT,=CL51' '
         MVC   ATTR,=CL16Ø' '
         MVC   RECLEN,=H'51'
         SR    R8,R8                      CLEAR R8 FOR USE AS ATTRLEN
         EXEC CICS RECEIVE INTO(INPUT)                                X
               LENGTH(RECLEN) RESP(RESP) NOHANDLE
         CLC   RESP,DFHRESP(EOC)
         BNE   ERR4
         CLC   TRANC,=C'WI1Ø'
         BNE   ERR1
*    CLOSE THE CURRENT FILE AND DISCARD IT
DISCARD  DS    ØH
         EXEC CICS SET FILE(FILENAME) CLOSED DISABLED                 X
               RESP(RESP) NOHANDLE
         CLC   RESP,DFHRESP(NORMAL)         IF NORMAL DISCARD THE
         BE    DISCARD1                     ENTRY
         CLC   RESP,DFHRESP(INVREQ)         IF INVREQ FILE IS PROBABLY
         BE    DISCARD1                     REMOTE SO GO DISCARD IT
         CLC   RESP,DFHRESP(FILENOTFOUND)   IF FILENOTFOUND GO DIRECTLY
```

```
         BE    DISCARDØ                      TO CREATE NEW FILE
         B     ERR7
DISCARD1 DS    ØH
         EXEC CICS DISCARD FILE(FILENAME) RESP(RESP) NOHANDLE
         CLC   RESP,DFHRESP(NORMAL)
         BNE   ERR5
DISCARDØ DS    ØH
         CLC   OPTION,=C'R'
         BE    REMOTE
         CLC   OPTION,=C'L'
         BE    LOCAL
         CLC   OPTION,=C' '
         BE    HELP
         B     ERR3
*   CREATE THE DEFINITION FOR THE REMOTE FILE
REMOTE   DS    ØH
         CLC   CONNID,=CL4'   '
         BNE   REMOTE3
         MVC   CONNID,=C'PATM'
REMOTE3  DS    ØH
         LA    R7,ATTR
         ADDIT REMOTEN
         ADDIT LSRPNONE
         ADDIT REMOTES
         ADDIT CONNID
         ADDIT CLOSBRAK
         STH   R8,HW
         EXEC CICS CREATE FILE(FILENAME) ATTRIBUTES(ATTR)              X
               ATTRLEN(HW) RESP(RESP) NOHANDLE
         CLC   RESP,DFHRESP(NORMAL)
         BNE   ERR6
         MVC   OUTLINE,=79C' '
         MVC   OUTLINE,=CL79'REMOTE WI1ØU DEFINED'
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         MVC   OUTLINE,=79C' '
         MVC   OUTLINE(11),=CL79'REMOTE ID: '
         MVC   OUTLINE+12(44),CONNID
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         EXEC CICS SEND PAGE
         B     RETURN
*  CREATE THE DEFINITION FOR THE LOCAL FILE
LOCAL    DS    ØH
         CLC   NEWNAME(1),=X'4Ø'
         BNE   LOCAL1
         MVC   NEWNAME,=44C' '
         MVC   NEWNAME(15),=C'YINV1.WI1Ø.£IAM'
```

```
LOCAL1   DS    ØH
         LA    R7,ATTR
         ADDIT ADDY
         ADDIT BROWSEY
         ADDIT DELETEY
         ADDIT UPDATEY
         ADDIT READY
         ADDIT LSRPNONE
         ADDIT DSNAME
         LA    R6,NEWNAME                FIND THE LENGTH OF THE DSNAME
         LA    R5,NEWNAME
         A     R6,DSLEN
LOCAL2   DS    ØH
         CLC   Ø(1,R6),=C' '
         BNE   LOCAL3
         BCT   R6,LOCAL2
LOCAL3   DS    ØH
         SR    R6,R5
         A     R6,ONE
         MVC   Ø(44,R7),NEWNAME          MOVE STRING TO ATTR
         AR    R8,R6                     ADD STR LEN TO LENGTH OF ATTR
         AR    R7,R6                     BUMP POS IN ATTR BY STR LEN
         ADDIT CLOSBRAK
         STH   R8,HW
         EXEC CICS CREATE FILE(FILENAME) ATTRIBUTES(ATTR)             X
               ATTRLEN(HW) RESP(RESP) NOHANDLE
         EXEC CICS SET FILE(FILENAME) CLOSED ENABLED                  X
               RESP(RESP) NOHANDLE
         MVC   OUTLINE,=79C' '
         MVC   OUTLINE,=CL79'LOCAL WI1ØU DEFINED'
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)          X
               ERASE ACCUM
         MVC   OUTLINE,=79C' '
         MVC   OUTLINE(1Ø),=CL79'FILENAME: '
         MVC   OUTLINE+11(44),NEWNAME
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)          X
               ERASE ACCUM
         EXEC CICS SEND PAGE
         B     RETURN
*
*  SEND HELP SCREEN IF NO PARAMETER ENTERED ON SCREEN
*
HELP     MVC   OUTLINE,=CL79'   TRANSACTION WI1Ø TOGGLES THE WI1Ø FILE X
               BETWEEN LOCAL AND REMOTE            '
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)          X
               ERASE ACCUM
         MVC   OUTLINE,=CL79'                                         X
                                                   '
```

```
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         MVC   OUTLINE,=CL79' FORMAT FOR SWITCHING TO LOCAL DEFINITION:X
                                                 '
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         MVC   OUTLINE,=CL79'                                          X
                                                 '
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         MVC   OUTLINE,=CL79'WI1Ø L FILENAME      NOTE: FILENAME DEFAULX
               TS TO YINV1.WI1Ø.£IAM IF LEFT BLANK  '
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         MVC   OUTLINE,=CL79'                                          X
                                                 '
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         MVC   OUTLINE,=CL79' FORMAT FOR SWITCHING TO REMOTE DEFINITIONX
               :                                 '
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         MVC   OUTLINE,=CL79'                                          X
                                                 '
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         MVC   OUTLINE,=CL79'WI1Ø R CONNID        NOTE: CONNID IS A 4 CX
               HARACTER ID WHICH REPRESENTS THE    '
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         MVC   OUTLINE,=CL79'                            NAME OF THE REMX
               OTE SYSTEM. DEFAULT IS PATM          '
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         MVC   OUTLINE,=CL79'                                          X
                                                 '
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         MVC   OUTLINE,=CL79' TYPING WI1Ø ONLY GETS THIS HELP SCREEN   X
                                                 '
         EXEC CICS SEND TEXT FROM(OUTLINE) LENGTH(L'OUTLINE)           X
               ERASE ACCUM
         EXEC CICS SEND PAGE
         B     RETURN
ERR1     MVC   OUTLINE(L'ERRM1),ERRM1
         B     SEND
ERR2     MVC   OUTLINE(L'ERRM2),ERRM2
         B     SEND
```

```
ERR3     MVC   OUTLINE(L'ERRM3),ERRM3
         B     SEND
ERR4     MVC   OUTLINE(L'ERRM4),ERRM4
         B     SEND
ERR5     MVC   OUTLINE(L'ERRM5),ERRM5
         B     SEND
ERR6     MVC   OUTLINE(L'ERRM6),ERRM6
         B     SEND
ERR7     MVC   OUTLINE(L'ERRM7),ERRM7
         B     SEND
ERR8     MVC   OUTLINE(L'ERRM8),ERRM8
         B     SEND
SEND     DS    ØH
         EXEC CICS SEND FROM(OUTLINE) LENGTH(L'OUTLINE) ERASE
RETURN   DS    ØH
         EXEC CICS RETURN
*
         TITLE 'BWWI1ØØ1 - LITERAL POOL'
*        LTORG
RECLEN   DC    H'51'
DSLEN    DC    F'43'
ONE      DC    F'1'
FILENAME DC    C'WI1ØU   '
ERRM1    DC    C'INVALID TRANSACTION CODE'
ERRM2    DC    C'CREATE REMOTE FILE FAILED'
ERRM3    DC    C'WI1Ø INVALID OPTION'
ERRM4    DC    C'RECIEVE FAILED'
ERRM5    DC    C'DISCARD OF WI1ØU FAILED'
ERRM6    DC    C'CREATE OF REMOTE WI1ØU FAILED'
ERRM7    DC    C'CLOSE OF WI1ØU FAILED'
ERRM8    DC    C'OPEN OF NEW FILE WI1ØU FAILED'
REMOTES  DC    C'REMOTES('
REMOTEN  DC    C'REMOTEN(WI1ØU) '
CLOSBRAK DC    C')'
UPDATEY  DC    C'UPDATE(YES) '
READY    DC    C'READ(YES) '
ADDY     DC    C'ADD(YES) '
BROWSEY  DC    C'BROWSE(YES) '
DELETEY  DC    C'DELETE(YES) '
DSNAME   DC    C'DSNAME('
LSRPNONE DC    C'LSRP(NONE) '
         END
```

Graham Clark
Senior Systems Programmer
Bristol & West (UK)

Listing the unreferenced groups in the CICS DFHCSD

Maintaining the contents and integrity of the DFHCSD datasets is one of the many duties of the CICS systems programmer. With the amount of day-to-day changes involved with the resource definitions in the CICS regions, there are times when group lists are deleted and some of the resource groups that are only referenced by that list are left behind. There are also resource groups that are defined and not put into any group lists, intentionally or unintentionally.

Unfortunately, CICS does not have a way to list resource groups in the CSD that are not referenced by any group list. Therefore this SAS program was created to help the systems programmer identify the CICS resource groups that may no longer be needed and could be removed from the DFHCSD.

One point the users of this program may notice is that the option RECFM=VB is specified in the first DATA procedure to generate the CICSDATA datasets. This option is specified so that SAS will treat the carriage control character of the DFHCSD output dataset consistently regardless of how the handling of VBA datasets is specified by the installation SAS defaults.

CSDGROUP

```
//&SYSUID$ JOB (1Ø2331Ø),'SHOW UNUSED GROUPS',CLASS=A,MSGCLASS=A,
//  MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=144Ø,REGION=8M
//*****************************************************************
//*
//* SAS PROGRAM TO LIST ALL CSD GROUPS NOT REFERENCED IN ANY
//*     GROUP LISTS
//*
//* PROGRAM RAN SUCCESSFULLY AGAINST DFHCSDUP OUTPUT FOR
//*   CICS V4.1 AND CICS/TS V1 (CICS V5.1)
//*
//*                                JACK HWANG
//*                                CSHWANG@HOTMAIL.COM
//*
```

```
//*******************************************************************
//*
//* FIRST ALL THE GROUP LISTS AND THE INDIVIDUAL GROUPS ARE
//*  PRODUCED TO SYSPRINT WHICH IS PASSED
//*
//UTIL   EXEC PGM=DFHCSDUP,PARM='CSD(READONLY)'
//STEPLIB  DD DISP=SHR,DSN=CICS.SDFHLOAD
//DFHCSD   DD DISP=SHR,DSN=CICS.DFHCSD
//SYSPRINT DD DISP=(MOD,PASS),DSN=&&SYSPRINT,UNIT=SYSALLDA,
//         SPACE=(CYL,(1,1))
//SYSUDUMP DD SYSOUT=*
//*
//SYSIN    DD *
 LIST LIST(*)
 LIST GROUP(*)
//*
//* NOW THE SAS PROGRAM TO PROCESS THE DFHCSDUP LIST OUTPUT
//*   AND PRODUCE LIST OF GROUPS NOT REFERENCED
//*
//SASCICSL EXEC SAS,COND=(4,LT)
//CSDOUT   DD   DISP=(OLD,DELETE),DSN=&&SYSPRINT
//SYSIN    DD   *

 /*                                                          */
 /* FIRST PRODUCE A SAS DATASET OF TWO VARIABLES: GROUP      */
 /* NAME AND LIST NAME, ONE OBSERVATION FOR EACH             */
 /* GROUP IN EACH LIST.  IN ADDITION, EVERY GROUP IN         */
 /* THE LIST WILL HAVE AN OBSERVATION WITH LIST NAME         */
 /* OF X'FFFFFFFF'                                           */
 /*                                                          */

 DATA CICSDATA (KEEP=GROUP LIST);
   INFILE CSDOUT RECFM=VB;
   INPUT @2  RES1 $10.
         @6  RES2 $5. @;
   FORMAT LIST  $8.;
   FORMAT GROUP $8.;
   RETAIN LIST;
   IF RES1 =: 'LIST NAME' THEN DO;
     INPUT @17 LIST $8.;
     RETURN;
   END;
   ELSE IF RES1 =: 'GROUP NAME' THEN DO;
     INPUT @14 GROUP $8.;
     LIST = 'FFFFFFFF'X;
     OUTPUT;
     RETURN;
   END;
```

```
  ELSE IF RES2 =: 'GROUP' THEN DO;
    INPUT @17 GROUP $8.;
    OUTPUT;
    RETURN;
  END;
  ELSE RETURN;

/*                                                    */
/* SORT THE PRODUCED DATASETS BY GROUP AND LIST SEQ.  */
/* THE GROUP ENTRY WITH X'FFFFFFFF' LIST NAME WILL BE */
/* THE LAST OBSERVATION FOR EACH GROUP                */
/*                                                    */

PROC SORT  DATA=CICSDATA;
  BY GROUP LIST ;

/*                                                    */
/* SORT THE SORTED DATASET AGAIN, THIS TIME ONLY WITH */
/* THE GROUP KEY.  SINCE THE DATASET IS ALREADY IN    */
/* SORTED ORDER, THE NODUPKEY OPRAND WILL SERVE TO    */
/* DROP ALL OBSERVATIONS FOR EACH GROUP EXCEPT THE    */
/* THE FIRST OBSERVATION, HAVING EITHER A VALID       */
/* LIST NAME OR LIST NAME OF X'FFFFFFFF'              */
/*                                                    */

PROC SORT  DATA=CICSDATA NODUPKEY;
  BY GROUP ;

/*                                                    */
/* READ IN THE SORTED DATASET, AND OUTPUT ONLY THE    */
/* GROUP NAMES WITH X'FFFFFFFF' FOR THE LIST NAME     */
/* TO A NEW SAS DATASET.                              */
/*                                                    */

DATA UNUSED (KEEP=GROUP) ;
  SET CICSDATA;
  IF LIST = 'FFFFFFFF'X THEN OUTPUT;
  ELSE RETURN;

/*                                                    */
/* PRINT THE LIST OF UNREFERENCED GROUPS IN THE CSD   */
/*                                                    */

PROC PRINT UNIFORM NOOBS;
  TITLE 'DEFINED BUT UNUSED GROUPS IN CSD' ;
//
```

Jack Hwang
Principal
HSA Systems (USA)

CICS/TS 1.3 newcopy facility for doctemplates

Anyone who has used the CICS Web Interface with any of the previous releases of CICS will by now have experienced some of the limitations of the DFHHTML template library, which are:

- Does not support concatenated libraries
- Does not support secondary extents
- Has no way of cacheing Web pages in memory.

The first two bullets simply make the management of this file awkward if it is to be shared between a large number of developers.

The last bullet has both a negative and a positive impact because:

1 CICS always re-reads the DFHHTML PDS when there is a call from the template manager. This incurs I/O for each template request.

2 CICS always re-reads the PDS; if the contents on a template have been changed, the changes will be visible as soon as the Web browser issues a refresh (IE) or reload (Netscape). If a refresh or a reload is not issued, you will still be looking at the original copy of your document in the browser's cache.

With CICS/TS 1.3 we are no longer limited to using just the DFHHTML PDS, which is still supported for compatibility.

Document templates can now be defined to CICS in a doctemplate resource definition type. This new RDO resource gives much greater flexibility for the deployment of templates.

CICS/TS 1.3 DOCTEMPLATES

Document templates can now be stored in the following places:

- A CICS file
- An extrapartition transient data (TD) queue
- A TS queue

- CICS program storage
- A URM-managed repository (for example DB2)
- An MVS partioned dataset (PDS).

For example:

```
CEDA  View DOctemplate( GLOGO    )
  DOctemplate    : GLOGO
  Group          : DAC13
  DEscription    : GIF EXAMPLE - NOTICE USE OF BINARY
 FULL TEMPLATE NAME
  TEmplatename   : GLOGO
 ASSOCIATED CICS RESOURCE
  File           :
  TSqueue        :
  TDqueue        :
  Program        :
  Exitpgm        :
 PARTITIONED DATASET
  DDname         : DFHHTML
  Membername     : GLOGO
 TEMPLATE PROPERTIES
  Appendcrlf     : Yes               Yes | No
  TYpe           : Binary            Binary | Ebcdic
```

Where:

- Doctemplate specifies the eight-character name of this DOCTEMPLATE definition, which is used by CEDA for management purposes.
- TEMPLATE(name) specifies the 48-character name of a template as used by the new CICS/TS 1.3 Web API.
- ASSOCIATED CICS RESOURCE specifies the appropriate resource name if using any of these media.
- PARTITIONED DATASET gives you the ability to specify the required DDname and member name.

Using doctemplates circumvents many of the management and performance problems described earlier. For example:

- It is now possible to create and maintain templates in memory, which is advantageous from a performance viewpoint.
- Note, if you are using a PDS to store your templates, concatenated libraries and secondary extents are still not supported.

- Now, however, each developer can effectively have their own library, and, in so doing, ease the management problem.

What's the problem with doctemplates then?

The problem with doctemplates is that in order to reduce the I/O associated with repetitively searching the PDS to retrieve a template, doctemplates are effectively cached in CICS memory.

Why is that a problem?

CICS/TS 1.3 does not support a newcopy function by either CEMT or the SPI. So changes you make to the HTML templates are not reflected back to the browser. In many ways doctemplates act like traditional CICS programs.

When a reference is made to a doctemplate using the new Web API and a doctemplate has been defined, a copy of the template will be brought into memory. (Note: doctemplates must be predefined if using the new CICS Web API. If using the old template manager call mechanism, doctemplates may be auto-installed. In either case, in order to pick up changes to your templates you need to discard the current version of the doctemplate using CEMT or the SPI , DISCARD DOCTEMPLATE function.) CICS then refers to this original version on all future CICS Web API requests.

What is the impact ?

Although newcopy of a doctemplate is not supported, you can discard and re-install the resource, which has the effect of doing a newcopy. This is cumbersome and there is also the question of security – do we want programmers to have CEDA INSTALL access?

IBM has accepted the need for a 'CEMT SET DOCTEMPLATE NEWCOPY' function as a future requirement.

The programs that follow provide a discard/install function and also demonstrate many of the new Web API commands. Please note that one of the problems inherent with this technique is that the entries which are dynamically created are not stored on the CSD but they are catalogued. Hence, if you do an initial start, your dynamically added entries are lost.

You could of course use one of the techniques described in *CICS Update*, issue 168, November 1999, to capture the define log, to process and install the resources in batch if necessary.

Software requirements

The software requirements are CICS/TS 1.3 and RDO definitions for the doctemplates XEPNTEM, XEPNTEM2, and XEPNTEM3 – see below:

```
DEFINE DOCTEMPLATE(XEPNTEM) GROUP(DAC13)
DESCRIPTION(HTML VERSION ON CREATE/DISCARD) TEMPLATENAME(XEPNTEM)
DDNAME(DFHHTML) MEMBERNAME(XEPNTEM) APPENDCRLF(YES) TYPE(EBCDIC)

DEFINE DOCTEMPLATE(XEPNTEM2) GROUP(DAC13)
DESCRIPTION(HTML VERSION ON CREATE/DISCARD) TEMPLATENAME(XEPNTEM2)
DDNAME(DFHHTML) MEMBERNAME(XEPNTEM2) APPENDCRLF(YES) TYPE(EBCDIC)

DEFINE DOCTEMPLATE(XEPNTEM3) GROUP(DAC13)
DESCRIPTION(HTML VERSION ON CREATE/DISCARD) TEMPLATENAME(XEPNTEM3)
DDNAME(DFHHTML) MEMBERNAME(XEPNTEM3) APPENDCRLF(YES) TYPE(EBCDIC)
```

Also needed are program definitions or autoinstall for the COBOL programs XEPNTEM, XEPNTEM2, XEPNTEM3, and XEPNTEM4.

CICS/TS 1.3 WEB API – LIGHTNING TOUR

The templates and the example programs that follow utilize the base address HTML tag, which makes life easy when trying to deploy on different IP addresses, whether invoked natively or through Domino Go Web server. This is a very useful option when migrating from development to production, or simply to a different MVS system.

In your HTML pages include the <BASE> tag, eg:

```
<BASE href="http://&hostv;">
```

In your code, ascertain what the host address is from the incoming headers:

```
TRANSACTION: CWBA PROGRAM: XEPNTEM3 TASK: 0000062 APPLID: CIRCTS02
DISPLAY:  00
  STATUS:  COMMAND EXECUTION COMPLETE
  EXEC CICS WEB READ
   HTTPHEADER ('Host')
   NAMELENGTH (4)
   VALUE ('r390.circle-group.com:6000')
```

```
   VALUELENGTH (24)
   NOHANDLE

  RESPONSE: NORMAL                          EIBRESP=Ø
```

Store the variable in your symbol table and refer to the symbol table when using the EXEC CICS DOCUMENT CREATE call:

```
Ø1 XEP-SYMBOLS.
          Ø3 filler pic x(8)  value 'docnm= '.
          Ø3 xep-state pic x(7) value spaces.
          Ø3 filler pic x(7) value '&hostv='.
          Ø3 xep-host pic x(24) value space.

     EXEC CICS
          DOCUMENT CREATE
          DOCTOKEN(XEP-TOKEN)
          DOCSIZE(XEP-RETRIEVE-LENGTH)
          TEMPLATE('XEPNTEM')
          SYMBOLLIST(XEP-SYMBOLS)
          LISTLENGTH(LENGTH OF XEP-SYMBOLS)
          RESP(XEP-RESP)
     END-EXEC
```

When you issue the DOCUMENT CREATE command CICS returns a 16-byte document token. If you want to add other templates to your document or perform any further symbol substitution, or simply want to send the current document to a browser, then you must include a reference to this document token:

```
     EXEC CICS
          WEB SEND
          DOCTOKEN(XEP-TOKEN)
          CLNTCODEPAGE('819')
          RESP(XEP-RESP)
     END-EXEC
```

Including a client code page on the send command instructs CICS to do EBCDIC/ASCII conversion.

The *CICS Internet Guide*, *Appendix J* now includes a list of the supported code pages.

On the CLNTCODEPAGE parameter of these commands, you can specify either the IANA value or the IBM CCSID value, eg IANA charset=iso-8859-1, IBM CCSID=819.

When processing form data, CICS/TS 1.3 now supports the EXEC

CICS WEB RECEIVE command:

```
EXEC CICS WEB
          RECEIVE
          SET(ADDRESS OF LS-BUFFER)
          LENGTH(XEP-BUFFER-LEN)
           CLNTCODEPAGE('819')
           HOSTCODEPAGE('Ø37')
END-EXEC
```

You need to specify the host code page for ASCII/EBCDIC translation.

If you install these programs on your system, after execution you might see the following in MSGUSR:

```
DFHDH0106 CIRCTS02 Document template definition SAMP04 has been deleted.
DFHDH0105 CIRCTS02 Document template definition SAMP04 has been added as
PDS-MEMBER(SAMP04) with template name SAMP04.

CREATE DOCTEMPLATE(SAMP04) DESCRIPTION(DYNAMIC) TEMPLATENAME(SAMP04)
DDNAME(DFHHTML) MEMBERNAME(SAMP04) APPENDCRLF(YES) TYPE(EBCDIC)
```

PROGRAM LOGIC

XEPNTEM

The primary purpose of this program is to send the initial template to the Web browser – see Figure 1.

XEPNTEM2

This program uses the Web API to receive the document name from the previous page – see Figure 2.

It then constructs some defaults for the various fields and sends out the second page.

This sample code deals only with templates stored in PDS files: with little effort it could support the other media (files, tsq etc) as well.

You could get really clever here and make the DDname the developer's user-id, or perhaps just supply a select box (see the *Append CRLF* field definition in XEPNTEM2 template).

XEPNTEM3

The purpose of this program is to receive the input from the previous

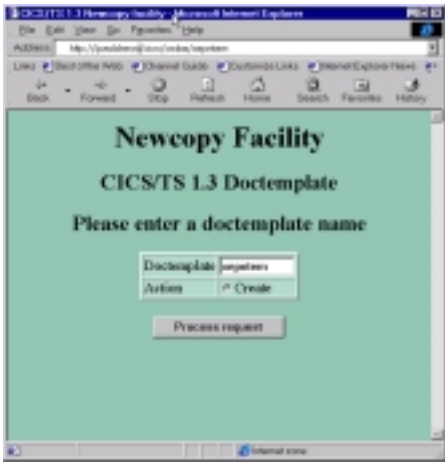

Figure 1: XEPNTEM screen

form, and reply with a success or failure message – see Figure 3. Program XEPNTEM4 actually does the discard/define processing.

If you do get a failure message, check the CICS log and/or MSGUSR. The most likely cause is message DFHAM4910 member name not found in the HTML library

From this point on, if you press 'Process request', the code just cycles through the second and last pages giving you the opportunity to newcopy some other pages.

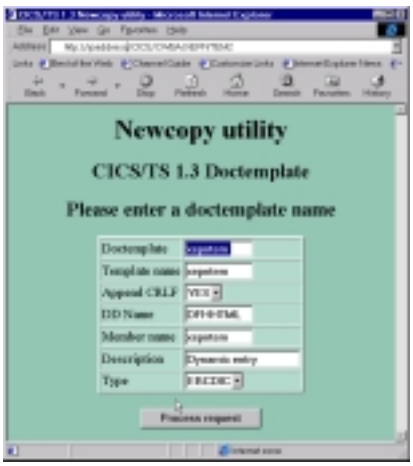

Figure 2: XEPNTEM2 screen

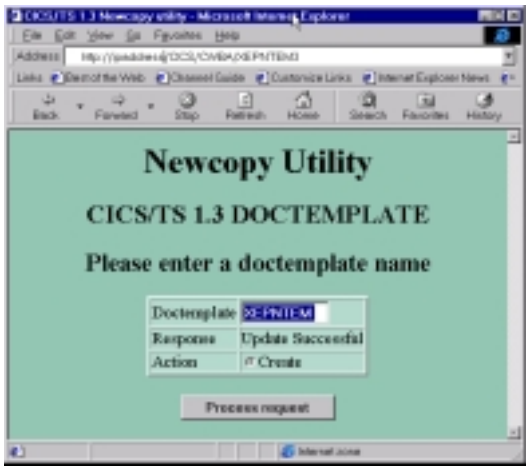

Figure 3: XEPNTEM3 screen

PROGRAM XEPNTEM

```
*****************************************
* MODULE NAME      XEPNTEM.COB
* HTML version of create/discard
* because CICS/TS 1.3 does not support
* new copy of DOCTEMPLATES
*****************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. XEPNTEM.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 01 XEP-TOKEN      PIC X(16).
 01 XEP-SYMBOLS.
    03 filler pic x(8)  value 'docnm= '.
    03 xep-state pic x(7) value spaces.
    03 filler pic x(7) value '&hostv='.
    03 xep-host pic x(24) value space.
 01 XEP-RETRIEVE-LENGTH PIC S9(8) BINARY.
 01 XEP-RESP            PIC S9(8) BINARY.
 01 XEP-HDR-NAME PIC X(100).
 01 XEP-HDR-VAL PIC X(200).
 LINKAGE SECTION.
 PROCEDURE DIVISION.
 AA-MAIN SECTION.
*    Find out what the host ip address is
     EXEC CICS WEB STARTBROWSE HTTPHEADER
     RESP(XEP-RESP)
     END-EXEC
     PERFORM UNTIL XEP-RESP NOT EQUAL DFHRESP(NORMAL)
     EXEC CICS WEB READNEXT
          HTTPHEADER (xep-hdr-NAME)
```

```
           NAMELENGTH(LENGTH OF xep-hdr-NAME)
           VALUE(xep-hdr-VAL)
           VALUELENGTH(LENGTH OF xep-hdr-VAL)
           RESP(XEP-RESP)
      END-EXEC
      if xep-hdr-name = "Host"
         move xep-hdr-val to XEP-host
         end-if
      end-perform
      EXEC CICS WEB ENDBROWSE HTTPHEADER
      RESP(XEP-RESP)
      END-EXEC
*     Use the document API to create a document
      EXEC CICS
           DOCUMENT CREATE
           DOCTOKEN(XEP-TOKEN)
           DOCSIZE(XEP-RETRIEVE-LENGTH)
           TEMPLATE('XEPNTEM')
           SYMBOLLIST(XEP-SYMBOLS)
           LISTLENGTH(LENGTH OF XEP-SYMBOLS)
           RESP(XEP-RESP)
      END-EXEC
*     Use the Web API to send it to the browser
      EXEC CICS
           WEB SEND
           DOCTOKEN(XEP-TOKEN)
           CLNTCODEPAGE('819')
           RESP(XEP-RESP)
      END-EXEC
      EXEC CICS RETURN END-EXEC.
 AA999-EXIT.
      EXIT.
                    STOP RUN.
```

PROGRAM XEPNTEM2

```
*****************************************
* MODULE NAME      XEPNTEM2.cob
*****************************************
 PROCESS XOPTS(NOLINKAGE)
 IDENTIFICATION DIVISION.
 PROGRAM-ID. XEPNTEM2.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
*        Symbols table
 Ø1 XEP-SYMBOLS.
    Ø3 filler             pic x(6)  value 'docnm='.
    Ø3 XEP-DOC-NM         PIC X(8) VALUE SPACE.
    Ø3 filler             pic x(7)  value '&doctn='.
    Ø3 XEP-DOC-TN         PIC X(48) VALUE SPACE.
```

```
    Ø3 filler             pic x(7)  value '&doccr='.
    Ø3 XEP-DOC-CR         PIC X(3) VALUE SPACE.
    Ø3 filler             pic x(7)  value '&docdd='.
    Ø3 XEP-DOC-DD         PIC X(8) VALUE SPACE.
    Ø3 filler             pic x(7)  value '&docmn='.
    Ø3 XEP-DOC-MN         PIC X(8) VALUE SPACE.
    Ø3 filler             pic x(7)  value '&docds='.
    Ø3 XEP-DOC-DS         PIC X(56) VALUE SPACE.
    Ø3 filler             pic x(7)  value '&docty='.
    Ø3 XEP-DOC-TY         PIC X(6) VALUE SPACE.
    Ø3 filler             pic x(5)  value '&act='.
    Ø3 XEP-DOC-ACT        PIC X(6) VALUE SPACES.
    Ø3 filler             PIC x(7) value '&hostv='.
    Ø3 XEP-host           pic x(24) value space.
*WORK AREA FOR UNSTRING
 Ø1  XEP-USER-VARS-PAIRS.
         15  XEP-STRING              PICTURE X(24Ø).
         15  XEP-STRING-LENGTH       PICTURE S9(Ø5) COMP-3.
         15  XEP-PARM-COUNT          PICTURE S9(Ø3) COMP-3.
         15  XEP-uvars-COUNT         PICTURE 9(Ø3) COMP-3.
         15  XEP-uvars-STRING OCCURS 1Ø TIMES
             INDEXED BY uvarsINDEX.
             2Ø  XEP-uvars-NAME      PICTURE X(1Ø).
             2Ø  XEP-uvars-VALUE     PICTURE X(1Ø).
 Ø1 XEP-SUB            PIC S9(8) COMP VALUE +Ø.
 Ø1 XEP-buffer-len     PIC S9(8) COMP VALUE +Ø.
 Ø1 XEP-RESP            PIC S9(8) BINARY.
 Ø1 XEP-hdr-name PIC x(1ØØ) value spaces.
 Ø1 XEP-hdr-val PIC x(2ØØ) value spaces.
 Ø1 XEP-TOKEN      PIC X(16).
 Ø1 XEP-RETRIEVE-LENGTH PIC S9(8) BINARY.
 Ø1 CRLF              PIC X(2)  VALUE IS X'ØD25'.
 LINKAGE SECTION.
 COPY DFHEIBLC.
  Ø1 DFHCOMMAREA  PIC X(1).
  Ø1 LS-BUFFER.
    Ø5 FILLER              PIC X(32764).
 PROCEDURE DIVISION USING DFHEIBLK DFHCOMMAREA.
*================================================================*
 AA-MAIN SECTION.
* MAIN CODE                                                      *
 CONTIBOARD.
     EXEC CICS WEB
          RECEIVE
          SET(ADDRESS OF LS-BUFFER)
          LENGTH(XEP-BUFFER-LEN)
          CLNTCODEPAGE('819')
          HOSTCODEPAGE('Ø37')
     END-EXEC
* Parse input data
     MOVE LOW-VALUES TO XEP-USER-VARS-PAIRS
```

```
     MOVE Ø TO XEP-STRING-LENGTH
              XEP-uvars-COUNT
     ADD XEP-BUFFER-LEN TO XEP-STRING-LENGTH
     UNSTRING ls-buffer
         DELIMITED BY '=' OR '&' or CRLF
         INTO XEP-uvars-NAME(Ø1) XEP-uvars-VALUE(Ø1)
              XEP-uvars-NAME(Ø2) XEP-uvars-VALUE(Ø2)
              XEP-uvars-NAME(Ø3) XEP-uvars-VALUE(Ø3)
              XEP-uvars-NAME(Ø4) XEP-uvars-VALUE(Ø4)
              XEP-uvars-NAME(Ø5) XEP-uvars-VALUE(Ø5)
              XEP-uvars-NAME(Ø6) XEP-uvars-VALUE(Ø6)
         TALLYING IN XEP-uvars-COUNT
     perform varying uvarsindex from 1 by 1 until
         uvarsindex > 6
         if xep-uvars-name(uvarsindex) = 'docnm'
            string xep-uvars-value(uvarsindex) delimited by '   '
            into xep-doc-nm
         end-if
         if xep-uvars-name(uvarsindex) = 'act'
            string xep-uvars-value(uvarsindex) delimited by '   '
            into xep-doc-act
         end-if
     END-PERFORM
* Set up some defaults
     MOVE XEP-DOC-NM TO XEP-DOC-TN XEP-DOC-MN
     move 'YES' to XEP-DOC-CR
     move 'DFHHTML' to XEP-DOC-DD
     move 'EBCDIC' to XEP-DOC-TY
     move 'Dynamic entry' to XEP-DOC-DS
* Find the Host name
* This example uses read rather than browse technique
* demonstrated in the first program
     MOVE 'Host' to XEP-hdr-NAME
     EXEC CICS WEB READ
          HTTPHEADER (XEP-hdr-NAME)
          NAMELENGTH(4)
          VALUE(XEP-hdr-VAL)
          VALUELENGTH(LENGTH OF XEP-hdr-VAL)
          RESP(XEP-RESP)
     END-EXEC
     MOVE XEP-hdr-VAL TO XEP-HOST
* add the header template to master document
     EXEC CICS
          DOCUMENT CREATE
          DOCTOKEN(XEP-TOKEN)
          DOCSIZE(XEP-RETRIEVE-LENGTH)
          TEMPLATE('XEPNTEM2')
          SYMBOLLIST(XEP-SYMBOLS)
          LISTLENGTH(LENGTH OF XEP-SYMBOLS)
     END-EXEC
* send the master document
```

```
     EXEC CICS
          WEB SEND
          DOCTOKEN(XEP-TOKEN)
          CLNTCODEPAGE('819')
     END-EXEC
     EXEC CICS RETURN END-EXEC.
 AA999-EXIT.
     EXIT.
                    STOP RUN.
```

PROGRAM XEPNTEM3

```
      ******************************************
      * MODULE NAME      XEPNTEM3.cob
      ******************************************
       PROCESS XOPTS(NOLINKAGE)
       IDENTIFICATION DIVISION.
       PROGRAM-ID. XEPNTEM3.
       ENVIRONMENT DIVISION.
       DATA DIVISION.
       WORKING-STORAGE SECTION.
      *        Symbols table
       Ø1 XEP-SYMBOLS.
          Ø3 filler             pic x(6)  value 'docnm='.
          Ø3 XEP-DOC-NM         PIC X(8) VALUE SPACE.
          Ø3 filler             pic x(7)  value '&doctn='.
          Ø3 XEP-DOC-TN         PIC X(48) VALUE SPACE.
          Ø3 filler             pic x(7)  value '&doccr='.
          Ø3 XEP-DOC-CR         PIC X(3) VALUE SPACE.
          Ø3 filler             pic x(7)  value '&docdd='.
          Ø3 XEP-DOC-DD         PIC X(8) VALUE SPACE.
          Ø3 filler             pic x(7)  value '&docmn='.
          Ø3 XEP-DOC-MN         PIC X(8) VALUE SPACE.
          Ø3 filler             pic x(7)  value '&docds='.
          Ø3 XEP-DOC-DS         PIC X(56) VALUE SPACE.
          Ø3 filler             pic x(7)  value '&docty='.
          Ø3 XEP-DOC-TY         PIC X(6) VALUE SPACE.
          Ø3 filler             pic x(5)  value '&act='.
          Ø3 XEP-DOC-ACT        PIC X(6) VALUE SPACES.
          Ø3 filler             PIC x(7) value '&hostv='.
          Ø3 XEP-host           pic x(24) value space.
          Ø3 filler             PIC x(7) value '&resp='.
          Ø3 XEP-resp-message   pic x(24) value space.
      *        commarea for call to XEPNTEM4
       Ø1 XEP-COMMAREA.
          Ø3  XEP-COM-NM PIC X(8).
          Ø3  XEP-COM-TN PIC X(48).
          Ø3  XEP-COM-CR PIC X(3).
          Ø3  XEP-COM-DD PIC X(8).
          Ø3  XEP-COM-MN PIC X(8).
```

```
    Ø3  XEP-COM-DS PIC X(58).
    Ø3  XEP-COM-TY PIC X(Ø6).
    Ø3  XEP-COM-RESP PIC S9(8) COMP.
*WORK AREA FOR UNSTRING
 Ø1  XEP-USER-VARS-PAIRS.
         15  XEP-STRING              PICTURE X(24Ø).
         15  XEP-STRING-LENGTH       PICTURE S9(Ø5) COMP-3.
         15  XEP-PARM-COUNT          PICTURE S9(Ø3) COMP-3.
         15  XEP-uvars-COUNT         PICTURE 9(Ø3) COMP-3.
         15  XEP-uvars-STRING OCCURS 1Ø TIMES
             INDEXED BY uvarsINDEX.
             2Ø  XEP-uvars-NAME      PICTURE X(1Ø).
             2Ø  XEP-uvars-VALUE     PICTURE X(1Ø).
 Ø1 XEP-SUB            PIC S9(8) COMP VALUE +Ø.
 Ø1 XEP-buffer-len     PIC S9(8) COMP VALUE +Ø.
 Ø1 XEP-RESP            PIC S9(8) BINARY.
 Ø1 xep-hdr-name PIC x(1ØØ) value spaces.
 Ø1 xep-hdr-val PIC x(2ØØ) value spaces.
 Ø1 XEP-TOKEN      PIC X(16).
 Ø1 XEP-RETRIEVE-LENGTH PIC S9(8) BINARY.
 Ø1 CRLF              PIC X(2)  VALUE IS X'ØD25'.
 77 counter            PIC S9(4) COMP.
 Ø1 lower  PIC X(27) VALUE 'abcdefghijklmnopqrstuvwxyz+'.
 Ø1 filler REDEFINES lower.
    Ø5  lowerc PIC X OCCURS 27 INDEXED BY lowerc-idx.
 Ø1 upper  PIC X(27) VALUE 'ABCDEFGHIJKLMNOPQRSTUVWXYZ '.
 Ø1 filler REDEFINES upper.
    Ø5  upperc PIC X OCCURS 27.
 LINKAGE SECTION.
 COPY DFHEIBLC.
  Ø1 DFHCOMMAREA  PIC X(1).
  Ø1 LS-BUFFER.
    Ø5 FILLER              PIC X(32764).
 PROCEDURE DIVISION USING DFHEIBLK DFHCOMMAREA.
*=================================================================*
 AA-MAIN SECTION.
* MAIN CODE                                                      *
 CONTIBOARD.
     EXEC CICS WEB
          RECEIVE
          SET(ADDRESS OF LS-BUFFER)
          LENGTH(XEP-BUFFER-LEN)
          CLNTCODEPAGE('819')
          HOSTCODEPAGE('Ø37')
     END-EXEC
* Parse input data
     MOVE LOW-VALUES TO XEP-USER-VARS-PAIRS
     MOVE Ø TO XEP-STRING-LENGTH
               XEP-uvars-COUNT
     ADD XEP-BUFFER-LEN TO XEP-STRING-LENGTH
     UNSTRING ls-buffer
```

```
           DELIMITED BY '=' OR '&' or CRLF
           INTO XEP-uvars-NAME(Ø1) XEP-uvars-VALUE(Ø1)
                XEP-uvars-NAME(Ø2) XEP-uvars-VALUE(Ø2)
                XEP-uvars-NAME(Ø3) XEP-uvars-VALUE(Ø3)
                XEP-uvars-NAME(Ø4) XEP-uvars-VALUE(Ø4)
                XEP-uvars-NAME(Ø5) XEP-uvars-VALUE(Ø5)
                XEP-uvars-NAME(Ø6) XEP-uvars-VALUE(Ø6)
                XEP-uvars-NAME(Ø7) XEP-uvars-VALUE(Ø7)
                XEP-uvars-NAME(Ø8) XEP-uvars-VALUE(Ø8)
                XEP-uvars-NAME(Ø9) XEP-uvars-VALUE(Ø9)
           TALLYING IN XEP-uvars-COUNT
     * put input data in appropriate fields
          PERFORM varying uvarsindex from 1 by 1 until
              uvarsindex > 9
              IF XEP-uvars-NAME(uvarsINDEX) = 'docnm'
                 STRING XEP-uvars-VALUE(uvarsINDEX) DELIMITED BY '   '
                 INTO XEP-DOC-NM
              END-IF
              IF XEP-uvars-NAME(uvarsINDEX) = 'doctn'
                 STRING XEP-uvars-VALUE(uvarsINDEX) DELIMITED BY '   '
                 INTO XEP-DOC-TN
              END-IF
              IF XEP-uvars-NAME(uvarsINDEX) = 'doccr'
                 STRING XEP-uvars-VALUE(uvarsINDEX) DELIMITED BY '   '
                 INTO XEP-DOC-CR
              END-IF
              IF XEP-uvars-NAME(uvarsINDEX) = 'docdd'
                 STRING XEP-uvars-VALUE(uvarsINDEX) DELIMITED BY '   '
                 INTO XEP-DOC-DD
              END-IF
              IF XEP-uvars-NAME(uvarsINDEX) = 'docmn'
                 STRING XEP-uvars-VALUE(uvarsINDEX) DELIMITED BY '   '
                 INTO XEP-DOC-MN
              END-IF
              IF XEP-uvars-NAME(uvarsINDEX) = 'docds'
                 STRING XEP-uvars-VALUE(uvarsINDEX) DELIMITED BY '   '
                 INTO XEP-DOC-DS
              END-IF
              IF XEP-uvars-NAME(uvarsINDEX) = 'docty'
                 STRING XEP-uvars-VALUE(uvarsINDEX) DELIMITED BY '   '
                 INTO XEP-DOC-TY
              END-IF
              IF XEP-uvars-NAME(uvarsINDEX) = 'act'
                 STRING XEP-uvars-VALUE(uvarsINDEX) DELIMITED BY '   '
                 INTO XEP-doc-ACT
              END-IF
          END-PERFORM
     * do uppercase translation
          PERFORM WITH TEST AFTER VARYING counter FROM 1 BY +1
              UNTIL counter = 27
              INSPECT XEP-doc-nm REPLACING ALL
```

```
            lowerc(counter) BY upperc(counter)
      END-PERFORM
      PERFORM WITH TEST AFTER VARYING counter FROM 1 BY +1
          UNTIL counter = 27
          INSPECT XEP-doc-tn REPLACING ALL
              lowerc(counter) BY upperc(counter)
      END-PERFORM
      PERFORM WITH TEST AFTER VARYING counter FROM 1 BY +1
          UNTIL counter = 27
          INSPECT XEP-doc-cr REPLACING ALL
              lowerc(counter) BY upperc(counter)
      END-PERFORM
      PERFORM WITH TEST AFTER VARYING counter FROM 1 BY +1
          UNTIL counter = 27
          INSPECT XEP-doc-dd REPLACING ALL
              lowerc(counter) BY upperc(counter)
      END-PERFORM
      PERFORM WITH TEST AFTER VARYING counter FROM 1 BY +1
          UNTIL counter = 27
          INSPECT XEP-doc-mn REPLACING ALL
              lowerc(counter) BY upperc(counter)
      END-PERFORM
      PERFORM WITH TEST AFTER VARYING counter FROM 1 BY +1
          UNTIL counter = 27
          INSPECT XEP-doc-ds REPLACING ALL
              lowerc(counter) BY upperc(counter)
      END-PERFORM
      PERFORM WITH TEST AFTER VARYING counter FROM 1 BY +1
          UNTIL counter = 27
          INSPECT XEP-doc-ty REPLACING ALL
              lowerc(counter) BY upperc(counter)
      END-PERFORM
  * build the commarea for call to XEPNTEM4
      move low-values to XEP-commarea
      MOVE XEP-DOC-NM TO XEP-COM-NM
      move XEP-DOC-TN TO XEP-COM-TN
      move XEP-DOC-CR TO XEP-COM-CR
      move XEP-DOC-DD TO XEP-COM-DD
      move XEP-DOC-MN TO XEP-COM-MN
      move XEP-DOC-DS TO XEP-COM-DS
      move XEP-DOC-TY TO XEP-COM-TY
      EXEC CICS LINK PROGRAM('XEPNTEM4')
                     COMMAREA(XEP-COMMAREA)
                     NOHANDLE
      END-EXEC
  * check the response from xepntem4
      evaluate XEP-com-resp
        when dfhresp(normal)
            move 'Update Successful' to XEP-resp-message
        when other
            move 'Failure          ' to XEP-resp-message
```

```
     end-evaluate
* Find the Host name
     MOVE 'Host' to xep-hdr-NAME
     EXEC CICS WEB READ
          HTTPHEADER (xep-hdr-NAME)
          NAMELENGTH(4)
          VALUE(xep-hdr-VAL)
          VALUELENGTH(LENGTH OF xep-hdr-VAL)
          RESP(XEP-RESP)
     END-EXEC
     MOVE xep-hdr-VAL TO XEP-HOST
* add the header template to master document
     EXEC CICS
          DOCUMENT CREATE
          DOCTOKEN(XEP-TOKEN)
          DOCSIZE(XEP-RETRIEVE-LENGTH)
          TEMPLATE('XEPNTEM3')
          SYMBOLLIST(XEP-SYMBOLS)
          LISTLENGTH(LENGTH OF XEP-SYMBOLS)
     END-EXEC
* SEND THE MASTER DOCUMENT
     EXEC CICS
          WEB SEND
          DOCTOKEN(XEP-TOKEN)
          CLNTCODEPAGE('819')
     END-EXEC
     EXEC CICS RETURN END-EXEC.
 AA999-EXIT.
     EXIT.
                    STOP RUN.
```

Editor's note: this article will be concluded in next month's issue.

David Clancy
Circle Computer Group (UK)

Finding DFHCSD duplicates and DFH$* groups – part 2

This month we conclude this article, which includes JCL to find duplicate entries in DFHCSD and also a procedure to remove DFH$ groups in your production DFHCSD. Included is a PL/I program to delete such keys, but you could also unload VSAM, edit the unloaded dataset and delete records, and then upload.*

HVPDYNA

This is the DYNALLOC macro required by HVPVS01:

```
* HVPDYNA :  DYNAMIC ALLOCATION  FUNCTIONS
* USAGE NOTES:
* -ALLOCATING MEMBER:
*      DSN IS ALLOCATED
*      MEMBER IS ALLOCATED.
*       IF PERMA-ATTRIBUTE:DDN CANNOT BE REUSED FOR MEMBER
* -ALLOCATING DSN ON TAPE:
*   TO PREVENT MOUNTING/DEMOUNTING OF VOLSER:
*    MOUNT UNIT,VOL=(SL,BROLØ1),USE=PRIVATE
*     EXECUTE JOB
*    UNLOAD UNIT
* DOEL  :
* INPUT :
*  PARMS:  RØ1 POINTS TO HVRDYNAA
*  FILES:
* OUTPUT:
*  FILES  :
*   LAYOUT:
* DOCUM:
*  MVS UPDATE 1Ø4 MAY 1995 P. 3   P. 42
*  MVS UPDATE 12Ø SEP 1996 P. 12Ø S99RBX EXTENDED REASON CODES
* ABENDS     :
* | OPM |    :
* REGISTER USAGE   :
* R13   : ADDRESS OF SAVE AREA FROM CALLER
*       : ADDRESS OF OWN SAVE AREA
* R14   : RETURN ADDRESS TO CALLER
* R15   : ENTRY ADDRESS OF CALLED PROGRAM / RETURN-CODE
* REG EQUS
SVCREGØ  EQU  RØØ
SVCREG1  EQU  RØ1
WORKØ2   EQU  RØ2
SVCREG2  EQU  RØ2
R_BASE   EQU  RØ3     BASE REGISTER
SP       EQU  RØ4     STACK POINTER
HVPDYNAPR EQU RØ5     POINTER TO HVRDYNAA
R_S99RBPTR EQU RØ6
R_S99RB   EQU RØ7
R_S99TUPL EQU RØ8
R_S99TUNIT EQU RØ9
WORKØ1    EQU R1Ø
R_DCB     EQU R11
PLITCAR  EQU  R12     PL/1 TCA
* OTHER EQUS
RBLEN    EQU (S99RBEND-S99RB)
YES          EQU C'Y'
BLANKO       EQU C' '
```

```
CMDALLOC     EQU C'A'
CMDUNALLOC   EQU C'U'
CMDINFORM    EQU C'I'        C1
RC_NO_PARMS EQU  16
RC_INV_HO   EQU  17
RC_INSUFF_STACK  EQU 18
RC_INV_PRMLEN EQU  19
RC_INV_IPO    EQU  2Ø
RC_INV_CMD   EQU  21
RC_INV_STATS EQU  22
RC_NO_DSORGPO EQU  23
RC_OPEN_ABEND  EQU  24  DYNAPERRC:ABEND CODE . DYNAPINFC: REASON CODE
RC_INSUFF_AUTH  EQU  25
RC_INV_LABEL EQU  26
RC_INV_NDISP EQU  27
RC_BASE_OPEN EQU  5ØØ
RC_BASE_CLOSE EQU  6ØØ
RC_BASE_BLDL  EQU  7ØØ
RC_BASE_DYNA EQU  1ØØØ
         COPY HVREGS     REGISTER EQUATES
         COPY HVRMACS    EIGEN MACROS
*   DSECTS
         DCBD DSORG=PO,DEVD=DA
         COPY HVRDYNAA
HVPDYNA  CSECT
HVPDYNA  AMODE 31
HVPDYNA  RMODE ANY
         SAVE  (14,12),,HVPDYNA.&SYSTIME..&SYSDATE
         LR    R_BASE,R15
         USING HVPDYNA,R_BASE
* GETMAIN WORKING STORAGE
         LR    HVPDYNAPR,RØ1 SAVE PARM ADDRESS
         L     RØ6,WS_FIX_LEN
*        DCB   SHOULD BE BELOW
         STORAGE OBTAIN,LENGTH=(RØ6),ADDR=(RØ1),LOC=BELOW
         ST    R13,4(Ø,1)    SAVE 13 IN OWN SAVE
         ST    RØ1,8(Ø,13)   SAVE ADR. OF SAVE IN PREV. SAVE
         LR    R13,RØ1
         USING WS,R13
         ST    HVPDYNAPR,PARMADR
* END GETMAIN WORKING STORAGE
         LA    SP,STACK
         XR    SVCREGØ,SVCREGØ
         ST    SVCREGØ,RC
BEGIN    EQU   *
*        CHECK PARMS
         LTR   HVPDYNAPR,HVPDYNAPR
         BZ    L_NO_PARMS          NO PARMS
         L     HVPDYNAPR,Ø(Ø,HVPDYNAPR)  ONLY 1 PARM
         LTR   HVPDYNAPR,HVPDYNAPR
         BNM   L_INV_HO             HO BIT SHOULD BE ON
```

```
         USING HVRDYNAA,HVPDYNAPR
         LA    WORKØ1,DYNAPEND-HVRDYNAA
         CH    WORKØ1,DYNAPL
         BNE   L_INV_PRMLEN
         CLC   DYNAPIPO,=C'ØØØØØØØØØ'
         BNE   L_INV_IPO
*        CHECK  STATS
         LA    RØØ,DYNAPSTATS
         LA    RØ1,L'DYNAPSTATS  //-1  DYNAPSTATS IS 1 TOO BIG
         LA    RØ2,$DALSTATS
         LA    R15,L'$DALSTATS
         BAS   R14,S_INDEX
*        R15   CONTAINS INDEX
         LTR   R15,R15
         BZ    L_INV_STATS
         LA    WORKØ2,$DALSTATS1
         IC    R15,Ø(R15,WORKØ2)
         STC   R15,STATSKEYW
*        CHECK  LABEL
         XC    LABELKEYW,LABELKEYW
         CLI   DYNAPLABEL,C' '
         BE    L_CHKLABELE
         CLI   DYNAPLABEL,X'ØØ'
         BE    L_CHKLABELE
         LA    RØØ,DYNAPLABEL
         LA    RØ1,L'DYNAPLABEL
         LA    RØ2,$DALLABEL
         LA    R15,L'$DALLABEL
         BAS   R14,S_INDEX
*        R15   CONTAINS INDEX
         LTR   R15,R15
         BZ    L_INV_LABEL
         LA    WORKØ2,$DALLABEL1
         IC    R15,Ø(R15,WORKØ2)
         STC   R15,LABELKEYW
L_CHKLABELE EQU *
*        CHECK  NDISP
         XC    NDISPKEYW,NDISPKEYW
         CLI   DYNAPNDISP,C' '
         BE    L_CHKNDISPE
         CLI   DYNAPNDISP,X'ØØ'
         BE    L_CHKNDISPE
         LA    RØØ,DYNAPNDISP
         LA    RØ1,L'DYNAPNDISP
         LA    RØ2,$DALNDISP
         LA    R15,L'$DALNDISP
         BAS   R14,S_INDEX
*        R15   CONTAINS INDEX
         LTR   R15,R15
         BZ    L_INV_NDISP
         LA    WORKØ2,$DALNDISP1
```

```
         IC    R15,Ø(R15,WORKØ2)
         STC   R15,NDISPKEYW
L_CHKNDISPE EQU *
*        INIT  S99RB
         LA    R_S99RBPTR,IEFRBPTR
         USING S99RBP,R_S99RBPTR
         LA    R_S99RB,IEFRB
         USING S99RB,R_S99RB
         ST    R_S99RB,S99RBPTR
         OI    S99RBPTR,S99RBPND       TURN ON HO BIT
         XC    S99RB(RBLEN),S99RB      ZERO OUT
         MVI   S99RBLN,RBLEN
         LA    R_S99TUPL,IEFTUPL
         USING S99TUPL,R_S99TUPL
         ST    R_S99TUPL,S99TXTPP
         LA    R_S99TUNIT,IEFTUNIT
         USING S99TUNIT,R_S99TUNIT
L_MAYBE_INFORM   EQU  *   C1
         CLI   DYNAPCMD,CMDINFORM   C1
         BNE   L_MAYBE_UNALLOC    C1
         BAS   R14,S_INFORM
         LTR   R15,R15
         BZ    RETURN
         LA    R15,RC_BASE_DYNA(Ø,R15)
         ST    R15,RC
         B     RETURN
L_MAYBE_UNALLOC  EQU  *   C1
         CLI   DYNAPCMD,CMDUNALLOC
         BNE   L_MAYBE_ALLOC
         MVC   BL_MBR,DYNAPMBR
         BAS   R14,S_UNALLOC
         LTR   R15,R15
         BZ    RETURN
         LA    R15,RC_BASE_DYNA(Ø,R15)
         ST    R15,RC
         B     RETURN
L_MAYBE_ALLOC  EQU  *
         CLI   DYNAPCMD,CMDALLOC
         BNE   L_INV_CMD
L_ALLOC      EQU *
         CLI   DYNAPMBR,BLANKO
         BE    L_ALLOC_Ø1
         BAS   R14,S_ALLOC_MBR
         B     RETURN
L_ALLOC_Ø1   EQU *
         MVC   BL_MBR,DYNAPMBR
         BAS   R14,S_ALLOC
         LTR   R15,R15
         BZ    RETURN
         LA    R15,RC_BASE_DYNA(Ø,R15)
         ST    R15,RC
```

```
         B     RETURN
*
RETURN   EQU   *
         L  R15,RC      LOAD RC BEFORE RESTORING R13
         L  WORKØ1,SAVEAREA+4    PRECEDING SAVE AREA
         ST R15,12+4(Ø,WORKØ1)   STORE R15 IN PRECEDING SA
         L  WORKØ2,WS_FIX_LEN
         STORAGE RELEASE,LENGTH=(WORKØ2),ADDR=(R13)
         LR R13,WORKØ1
         RETURN (14,12)   RC=ALREADY STORED IN PREV SA
L_NO_PARMS  EQU *
         LA     R15,RC_NO_PARMS
         ST     R15,RC
         B      RETURN
L_INV_HO     EQU *
         LA     R15,RC_INV_HO
         ST     R15,RC
         B      RETURN
L_INV_IPO    EQU *
         LA     R15,RC_INV_IPO
         ST     R15,RC
         B      RETURN
L_INV_PRMLEN EQU *
         LA     R15,RC_INV_PRMLEN
         ST     R15,RC
         B      RETURN
L_INV_CMD    EQU *
         LA     R15,RC_INV_CMD
         ST     R15,RC
         B      RETURN
L_INV_STATS  EQU *
         LA     R15,RC_INV_STATS
         ST     R15,RC
         B      RETURN
L_INV_LABEL  EQU *
         LA     R15,RC_INV_LABEL
         ST     R15,RC
         B      RETURN
L_INV_NDISP  EQU *
         LA     R15,RC_INV_NDISP
         ST     R15,RC
         B      RETURN
*SUBROUTINES
S_INFORM        EQU  *
         LA    SP,4(Ø,SP)
         ST    R14,Ø(,SP)
*
         MVI   S99VERB,S99VRBIN
         LA    R_S99TUPL,IEFTUPL
         LA    R_S99TUNIT,IEFTUNITE
         BCTR  R_S99TUNIT,Ø          -1
```

```
*         BY DDNAME
          LA    WORKØ1,6+L'DYNAPDDN   RESERVE SPACE
          SLR   R_S99TUNIT,WORKØ1
          LA    WORKØ1,DINDDNAM       INFORM BY DDNAME
          STH   WORKØ1,S99TUKEY
          LA    WORKØ1,1
          STH   WORKØ1,S99TUNUM       1 PARM
          LA    WORKØ1,L'DYNAPDDN
          STH   WORKØ1,S99TULNG
          MVC   S99TUPAR(L'DYNAPDDN),DYNAPDDN
          ST    R_S99TUNIT,S99TUPTR
*         RETURN DSNAME
          LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
          LA    WORKØ1,6+44           RESERVE SPACE
          SLR   R_S99TUNIT,WORKØ1
          LA    WORKØ1,DINRTDSN
          STH   WORKØ1,S99TUKEY
          LA    WORKØ1,1
          STH   WORKØ1,S99TUNUM         1 PARM
          LA    WORKØ1,44
          STH   WORKØ1,S99TULNG
          ST    R_S99TUNIT,S99TUPTR
          ST    R_S99TUNIT,DALDSNAMTU@    SAVE @
*         RETURN  DSORG           C2
          LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
          LA    WORKØ1,8           RESERVE SPACE
          SLR   R_S99TUNIT,WORKØ1
          LA    WORKØ1,DINRTORG
          STCM  WORKØ1,B'ØØ11',S99TUKEY
          LA    WORKØ1,1
          STCM  WORKØ1,B'ØØ11',S99TUNUM
          LA    WORKØ1,2
          STCM  WORKØ1,B'ØØ11',S99TULNG
          ST    R_S99TUNIT,S99TUPTR
          ST    R_S99TUNIT,DALRTORGTU@    C2
L_DO_INFORM    EQU *
          OI    S99TUPTR,S99TUPLN     TURN ON HO BIT ON LAST PARM
          LR    RØ1,R_S99RBPTR
          DYNALLOC
          MVC   DYNAPERRC,S99ERROR
          MVC   DYNAPINFC,S99INFO
          LTR   R15,R15
          BNZ   S_INFORM_RET
* COPY BACK DSNAME
          L     R_S99TUNIT,DALDSNAMTU@
          XR    RØ2,RØ2
          ICM   RØ2,B'ØØ11',S99TULNG
          BCTR  RØ2,Ø
          EX    RØ2,L_COPY_DSN
* COPY BACK DSORG
          L     R_S99TUNIT,DALRTORGTU@
```

```
*        LOOK UP DSORG
         LA    RØØ,S99TUPAR
         LA    RØ1,2
         LA    RØ2,$DALRTORG_BEGIN
         LA    R15,$DALRTORG_END-$DALRTORG_BEGIN
         BAS   R14,S_INDEX
*        R15   CONTAINS INDEX   Ø IF NOT FOUND
         LA    WORKØ2,$DSORG_BEGIN
         SLL   R15,2              X4
         LA    WORKØ2,Ø(R15,WORKØ2)
         MVC   DYNAPDSORG,Ø(WORKØ2)
         XR    R15,R15
S_INFORM_RET   EQU *
         L     R14,Ø(,SP)
         S     SP,$FW4
         BR    R14
L_COPY_DSN     MVC   DYNAPDSN(Ø),S99TUPAR
S_UNALLOC       EQU  *
         LA    SP,4(Ø,SP)
         ST    R14,Ø(,SP)
*        SET UP TUPL TUNIT
         MVI   S99VERB,S99VRBUN
         LA    R_S99TUPL,IEFTUPL
         LA    WORKØ1,4
         SR    R_S99TUPL,WORKØ1      SET AT TUPL-1
         LA    R_S99TUNIT,IEFTUNITE  SET 1 BEYOND END.
L_DUNREMOV     EQU  *   REMOVE IN-USE ATTRIBUTE?
         CLI   DYNAPUNOPT,C'R'
         BNE   L_DUNREMOVE
         LA    WORKØ1,4              RESERVE SPACE
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DUNREMOV
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,Ø
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         ST    R_S99TUNIT,S99TUPTR
L_DUNREMOVE    EQU *
L_DUNUNALC     EQU  *   DEALLOC EVEN PERMANENTLY ALLOCATED?
         CLI   DYNAPUNOPT,C'U'
         BNE   L_DUNUNALCE
         LA    WORKØ1,4                  RESERVE SPACE
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DUNUNALC
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,Ø
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         ST    R_S99TUNIT,S99TUPTR
L_DUNUNALCE    EQU *
*        UNALLOCATE BY DDNAME
```

```
         LA    WORKØ1,6+L'DYNAPDDN       MAKE ROOM
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DUNDDNAM
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1                  1 PARM
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,L'DYNAPDDN
         STCM  WORKØ1,B'ØØ11',S99TULNG
         MVC   S99TUPAR(L'DYNAPDDN),DYNAPDDN
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         ST    R_S99TUNIT,S99TUPTR
*        CLC   BL_MBR,BLANKO_8
         CLI   BL_MBR,C' '
         BE    L_DO_UNALLOC
         CLI   BL_MBR,X'ØØØØ'
         BE    L_DO_UNALLOC
L_DUNMEMBR   EQU  *
         LA    WORKØ1,6+L'BL_MBR   MAKE ROOM
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DUNMEMBR
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1                1 PARM
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,L'BL_MBR
         STCM  WORKØ1,B'ØØ11',S99TULNG
         MVC   S99TUPAR(L'BL_MBR),BL_MBR
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         ST    R_S99TUNIT,S99TUPTR
L_DUNDSNAM   EQU  *        IF DUNMEMBR -> DUNDSNAM
         LA    WORKØ1,6+L'DYNAPDSN MAKE ROOM
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DUNDSNAM
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1                1 PARM
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,L'DYNAPDSN
         STCM  WORKØ1,B'ØØ11',S99TULNG
         MVC   S99TUPAR(L'DYNAPDSN),DYNAPDSN
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         ST    R_S99TUNIT,S99TUPTR
L_DO_UNALLOC   EQU *
         OI    S99TUPTR,S99TUPLN     TURN ON HO BIT ON LAST PARM
         LR    RØ1,R_S99RBPTR
         DYNALLOC
         MVC   DYNAPERRC,S99ERROR
         MVC   DYNAPINFC,S99INFO
S_UNALLOC_RT   EQU  *
         L     R14,Ø(,SP)
         S     SP,$FW4
         BR    R14
S_ALLOC       EQU  *
```

```
         LA    SP,4(Ø,SP)
         ST    R14,Ø(,SP)
         MVI   S99VERB,S99VRBAL
         LA    R_S99TUPL,IEFTUPL
         LA    WORKØ1,4
         SR    R_S99TUPL,WORKØ1      SET AT TUPL-1
*        SET 1 BEYOND END.
         LA    R_S99TUNIT,IEFTUNITE
*        BY    DSNAME
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         LA    WORKØ1,6+L'DYNAPDSN   RESERVE SPACE
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALDSNAM
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1                1 PARM
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,L'DYNAPDSN
         STCM  WORKØ1,B'ØØ11',S99TULNG
         MVC   S99TUPAR(L'DYNAPDSN),DYNAPDSN
         ST    R_S99TUNIT,S99TUPTR
****     LA    R_S99TUNIT,2+2+2(WORKØ1,R_S99TUNIT) POINT TO NEXT TU
         CLC   BL_MBR,BLANKO_8
         BE    L_DALSTATS
*        MEMBER NAME
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         LA    WORKØ1,6+L'BL_MBR     RESERVE SPACE
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALMEMBR
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1                1 PARM
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,L'BL_MBR
         STCM  WORKØ1,B'ØØ11',S99TULNG
         MVC   S99TUPAR(L'BL_MBR),BL_MBR
         ST    R_S99TUNIT,S99TUPTR
***      LA    R_S99TUNIT,2+2+2(WORKØ1,R_S99TUNIT) POINT TO NEXT TU
L_DALSTATS     EQU   *
*        DATA  SET STATUS
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         LA    WORKØ1,6+1            RESERVE SPACE
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALSTATS
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         STCM  WORKØ1,B'ØØ11',S99TULNG
* KEYVALUE WAS STORED IN STATSKEYW
         MVC   S99TUPAR(1),STATSKEYW
         ST    R_S99TUNIT,S99TUPTR
****     LA    R_S99TUNIT,2+2+2(WORKØ1,R_S99TUNIT) POINT TO NEXT TU
*        IF DDN=' '  SYSTEM WILL RETURN DDNAME
```

```
         CLC   DYNAPDDN,BLANKO_8
         BE    L_DALRTDDN
L_DALDDNAM     EQU    *
         LA    WORKØ2,DALDDNAM
         B     L_RT_DD
L_DALRTDDN     EQU    *
         LA    WORKØ2,DALRTDDN
L_RT_DD        EQU    *
*        DDNAME
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         LA    WORKØ1,6+8            RESERVE SPACE
         SLR   R_S99TUNIT,WORKØ1
         STCM  WORKØ2,B'ØØ11',S99TUKEY
         LA    WORKØ1,1
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,8
         STCM  WORKØ1,B'ØØ11',S99TULNG
         MVC   S99TUPAR(L'DYNAPDDN),DYNAPDDN
         ST    R_S99TUNIT,S99TUPTR
         ST    R_S99TUNIT,DALDDNAMTU@
**       LA    R_S99TUNIT,2+2+2(WORKØ1,R_S99TUNIT) POINT TO NEXT TU
L_DALRTORG     EQU  *   RETURN DSORG
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         LA    WORKØ1,6+2            RESERVE SPACE
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALRTORG
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,2
         STCM  WORKØ1,B'ØØ11',S99TULNG
         ST    R_S99TUNIT,S99TUPTR
         ST    R_S99TUNIT,DALRTORGTU@
L_DALVLSER     EQU  *   VOLUME SERIAL
         CLI   DYNAPVLSER,C' '
         BE    L_DALVLSERE
         CLI   DYNAPVLSER,X'ØØ'
         BE    L_DALVLSERE
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         LA    WORKØ1,6+6            RESERVE SPACE
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALVLSER
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,6
         STCM  WORKØ1,B'ØØ11',S99TULNG
         MVC   S99TUPAR(6),DYNAPVLSER
         ST    R_S99TUNIT,S99TUPTR
L_DALVLSERE    EQU  *
L_DALUNIT      EQU  *   UNIT
```

```
         CLI   DYNAPUNIT,C' '
         BE    L_DALUNITE
         CLI   DYNAPUNIT,X'ØØ'
         BE    L_DALUNITE
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         LA    WORKØ1,6+L'DYNAPUNIT
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALUNIT
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,L'DYNAPUNIT
         STCM  WORKØ1,B'ØØ11',S99TULNG
         MVC   S99TUPAR(L'DYNAPUNIT),DYNAPUNIT
         ST    R_S99TUNIT,S99TUPTR
L_DALUNITE     EQU  *
L_DALDEFER     EQU  *
         CLI   DYNAPDEFER,C'Y'
         BNE   L_DALDEFERE
         LA    WORKØ1,4           MAKE ROOM
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALDEFER
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,Ø
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         ST    R_S99TUNIT,S99TUPTR
L_DALDEFERE    EQU  *
L_DALDSSEQ     EQU  *   DSSEQ FSEQN
         LH    WORKØ1,DYNAPDSSEQ
         LTR   WORKØ1,WORKØ1
         BZ    L_DALDSSEQE
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         LA    WORKØ1,6+2
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALDSSEQ
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,2
         STCM  WORKØ1,B'ØØ11',S99TULNG
         MVC   S99TUPAR(2),DYNAPDSSEQ
         ST    R_S99TUNIT,S99TUPTR
L_DALDSSEQE    EQU  *
L_DALLABEL     EQU  *
         CLI   LABELKEYW,X'ØØ'
         BE    L_DALLABELE
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         LA    WORKØ1,6+1
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALLABEL
```

```
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,1
         STCM  WORKØ1,B'ØØ11',S99TULNG
         MVC   S99TUPAR(1),LABELKEYW
         ST    R_S99TUNIT,S99TUPTR
L_DALLABELE    EQU  *
L_DALNDISP     EQU  *
         CLI   NDISPKEYW,X'ØØ'
         BE    L_DALNDISPE
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         LA    WORKØ1,6+1
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALNDISP
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,1
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    WORKØ1,1
         STCM  WORKØ1,B'ØØ11',S99TULNG
         MVC   S99TUPAR(1),NDISPKEYW
         ST    R_S99TUNIT,S99TUPTR
L_DALNDISPE    EQU  *
L_DALPRIVT     EQU  *
         CLI   DYNAPPRIVT,C'Y'
         BNE   L_DALPRIVTE
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         LA    WORKØ1,4
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALPRIVT
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,Ø
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         ST    R_S99TUNIT,S99TUPTR
L_DALPRIVTE  EQU  *
L_DALPERMA     EQU  *
         CLI   DYNAPPERMA,C'Y'
         BNE   L_DALPERMAE
         LA    WORKØ1,4                  MAKE ROOM
         SLR   R_S99TUNIT,WORKØ1
         LA    WORKØ1,DALPERMA
         STCM  WORKØ1,B'ØØ11',S99TUKEY
         LA    WORKØ1,Ø
         STCM  WORKØ1,B'ØØ11',S99TUNUM
         LA    R_S99TUPL,4(Ø,R_S99TUPL)  POINT TO NEXT TUPL
         ST    R_S99TUNIT,S99TUPTR
L_DALPERMAE  EQU  *
L_DOIT   EQU   *
         CR    R_S99TUPL,R_S99TUNIT
         BL    L_DOITØ5
         LA    R15,RC_INSUFF_STACK
```

```
         B     L_ALLOC_RT
L_DOITØ5 EQU   *
         OI    S99TUPTR,S99TUPLN         TURN ON HO BIT ON LAST PARM
         LR    RØ1,R_S99RBPTR
         DYNALLOC
         MVC   DYNAPERRC,S99ERROR
         MVC   DYNAPINFC,S99INFO
         LTR   R15,R15
         BZ    S_ALLOC_OK
         B     L_ALLOC_RT
S_ALLOC_OK     EQU  *
* COPY BACK DDNAME
         L     R_S99TUNIT,DALDDNAMTU@
         MVC   DYNAPDDN,S99TUPAR
* COPY BACK DSORG
         L     R_S99TUNIT,DALRTORGTU@
*        LOOK UP DSORG
         LA    RØØ,S99TUPAR
         LA    RØ1,2
         LA    RØ2,$DALRTORG_BEGIN
         LA    R15,$DALRTORG_END-$DALRTORG_BEGIN
         BAS   R14,S_INDEX
*        R15   CONTAINS INDEX   Ø IF NOT FOUND
         LA    WORKØ2,$DSORG_BEGIN
         SLL   R15,2              X4
         LA    WORKØ2,Ø(R15,WORKØ2)
         MVC   DYNAPDSORG,Ø(WORKØ2)
         XR    R15,R15
L_ALLOC_RT     EQU  *
         L     R14,Ø(,SP)
         S     SP,$FW4
         BR    R14
S_ALLOC_MBR    EQU  *
         LA    SP,4(Ø,SP)
         ST    R14,Ø(,SP)
*        INIT
         MVI AX_PRMLST,X'ØØ'
         MVC AX_PRMLST+1(L'AX_PRMLST-1),AX_PRMLST
*        ALLOCATE FIRST WITHOUT MEMBER
         MVC    BL_MBR,BLANKO_8
         BAS    R14,S_ALLOC
         LTR    R15,R15
         BZ     S_ALLOC_MBR_Ø2
         LA     R15,RC_BASE_DYNA(Ø,R15)
         ST     R15,RC
         B      S_ALLOC_MBR_RT
S_ALLOC_MBR_Ø2  EQU *
         CLC    DYNAPDSORG(2),DSORGPO
         BE     S_ALLOC_MBR_Ø3
         LA     R15,RC_NO_DSORGPO
         ST     R15,RC
```

```
         BAS    R14,S_UNALLOC
         B      S_ALLOC_MBR_RT
S_ALLOC_MBR_Ø3  EQU *
         LA    R_DCB,DCBPO
*        TO STAY REENTRANT
         MVC   DCBPO,$DCBPO
         MVC   DCBPO_OPELST,$DCBPO_OPELST
         MVC   DCBPO_CLSLST,$DCBPO_CLSLST
         USING IHADCB,R_DCB
         MVC   DCBDDNAM,DYNAPDDN
*        MOVE ABEND EXIT TO LOW STORAGE
         MVC   ABEND_EXIT,$ABEND_EXIT
*        ABEND EXIT LAST AND ONLY EXIT
         LA    WORKØ1,X'8Ø'+X'11'
         STC   WORKØ1,@ABEND_EXIT
         LA    WORKØ1,ABEND_EXIT
         STCM  WORKØ1,B'Ø111',@ABEND_EXIT+1
         LA    WORKØ1,@ABEND_EXIT
         STCM  WORKØ1,B'Ø111',DCBEXLSA   FILL IN EXLST
         DROP  R_DCB
         OPEN  ((R_DCB),INPUT),MF=(E,DCBPO_OPELST),MODE=31
*        INIT ABENDZONES + DID   ABEND OCURR ?
         XR    WORKØ1,WORKØ1
         ICM   WORKØ1,B'ØØØ1',AX_PRMLST+2
         STH   WORKØ1,ABENDREASON
*        LH    WORKØ1,AX_PRMLST
         L     WORKØ1,AX_PRMLST
*        LTR   WORKØ1,WORKØ1
*        BZ    L_TEST_R15
*        SRL   WORKØ1,4            SCC ONLY 12 BITS
         SRL   WORKØ1,2Ø           SCC ONLY 12 BITS
         STH   WORKØ1,ABENDCODE
         LTR   WORKØ1,WORKØ1
         BZ    L_TEST_R15
         LA    R15,RC_OPEN_ABEND
         CLC   ABENDCODE,$ABEND913
         BNE   L_OPEN_ERROR
         CLC   ABENDREASON,$ABEND913RC
         BNE   L_OPEN_ERROR
         LA    R15,RC_INSUFF_AUTH
         B     L_OPEN_ERROR
L_TEST_R15   EQU *
         LTR   R15,R15
         BZ    S_ALLOC_MBR_Ø4
         LA    R15,RC_BASE_OPEN(Ø,R15)
L_OPEN_ERROR   EQU *
         ST    R15,RC
         BAS    R14,S_UNALLOC
*        EVENTUAL ABEND + REASON CODE
         MVC    DYNAPERRC,ABENDCODE
         MVC    DYNAPINFC,ABENDREASON
```

```
         B      S_ALLOC_MBR_RT
S_ALLOC_MBR_Ø4  EQU *
         USING IHADCB,R_DCB
         MVC   DYNAPLRECL,DCBLRECL
         MVC   DYNAPBLKSI,DCBBLKSI
         DROP  R_DCB
*        PREPARE FOR BLDL
         LA    RØØ,1
         STH   RØØ,BL_ENTRY#
         LA    RØØ,BL_1_END-BL_1
         STH   RØØ,BL_ENT_LEN
         MVC    BL_MBR,DYNAPMBR
         BLDL  (R_DCB),BLDL_LIST
         MVC    BL_MBR,BLANKO_8
         LTR   R15,R15
         BZ    S_ALLOC_MBR_Ø5
         LA    R15,RC_BASE_BLDL(Ø,R15)
         ST    R15,RC
S_ALLOC_MBR_Ø5  EQU *
         CLOSE ((R_DCB)),MF=(E,DCBPO_CLSLST),MODE=31
         LTR   R15,R15
         BZ    S_ALLOC_MBR_Ø6
         LA    R15,RC_BASE_CLOSE(Ø,R15)
         ST    R15,RC
S_ALLOC_MBR_Ø6  EQU *
*        UNALLOC  PDS
         BAS   R14,S_UNALLOC
         LTR    R15,R15
         BZ     S_ALLOC_MBR_Ø8
         LA     R15,RC_BASE_DYNA(Ø,R15)
         ST     R15,RC
S_ALLOC_MBR_Ø8  EQU *
*        ALL   OK ?
         L     R15,RC
         LTR   R15,R15
         BNZ   S_ALLOC_MBR_RT
*        ALLOCATE MEMBER
         MVC   BL_MBR,DYNAPMBR
         BAS   R14,S_ALLOC
         LTR    R15,R15
         BZ     S_ALLOC_MBR_RT
         LA     R15,RC_BASE_DYNA(Ø,R15)
         ST     R15,RC
S_ALLOC_MBR_RT EQU  *
         L     R14,Ø(,SP)
         S     SP,$FW4
         BR    R14
S_INDEX        EQU  *
* RØØ:@ELEM RØ1:LEN ELEM RØ2:@STRING R15:LEN STRING
* R15: Ø OR INDEX #
         LA    SP,4(Ø,SP)
```

```
         ST    R14,Ø(,SP)
         LR    R14,RØØ
         LA    RØØ,Ø(RØ2,R15)       EINDADRES+1
         XR    R15,R15
         BCTR  RØ1,Ø                -1 FOR EX
S_INDEX_1  EQU  *
         EX    RØ1,L_IX_CMP
         BE    S_INDEX_FND
         LA    RØ2,1(RØ1,RØ2)       +1 TO COMPENSATE BCTR
         CR    RØ2,RØØ
         BNL   S_INDEX_NOTFND
*        ST    RØ2,Ø(,SP)
         LA    RØ2,Ø(RØ1,RØ2)  CAN IT BE CONTAINED IN STRING ?
         CR    RØ2,RØØ
         BNL   S_INDEX_NOTFND
         SR    RØ2,RØ1
*        L     RØ2,Ø(,SP)
         LA    R15,1(Ø,R15)
         B     S_INDEX_1
S_INDEX_END    EQU   *
*        S     SP,$FW4      SAVE_ZONE
         L     R14,Ø(,SP)
         S     SP,$FW4
         BR    R14
S_INDEX_FND    EQU  *
               LA   R15,1(Ø,R15)
               B    S_INDEX_END
S_INDEX_NOTFND    EQU  *
               XR   R15,R15
               B    S_INDEX_END
L_IX_CMP  CLC  Ø(Ø,RØ2),Ø(R14)
* KONSTANTEN
WS_FIX_LEN DC  A(WS_FIX_END-WS)
BLANKO_8   DC  C'        '
           DS  ØH
$ABEND913  DC  X'Ø913'   ABEND 913-38
$ABEND913RC DC X'ØØ38'
$DALSTATS  DC  C'OLD MOD NEW SHR '
$DALSTATS1 DC  X'ØØØ1Ø2Ø4Ø8'   FIRST ØØ IF NOTFOUND
$DALNDISP  DC  C'UNCATLG CATLG   DELETE  KEEP    '
$DALNDISP1 DC  X'ØØØ1Ø2Ø4Ø8'   FIRST ØØ IF NOTFOUND
$DALLABEL  DC  C'NL  SL  NSL SUL BLP LTM AL  AUL '
$DALLABEL1 DC  X'ØØØ1Ø2Ø4ØA1Ø214Ø48' FIRST ØØ IF NOTFOUND
$DALRTORG_BEGIN EQU *
           DC  X'ØØØØ'    UNKNOWN
           DC  X'ØØØ4'    TR
           DC  X'ØØØ8'    VSAM
           DC  X'ØØ2Ø'    TQ
           DC  X'ØØ4Ø'    TX
           DC  X'ØØ8Ø'    GS
           DC  X'Ø2ØØ'    PO
```

```
           DC  X'0300'    POU
           DC  X'0400'    MQ
           DC  X'0800'    CQ
           DC  X'1000'    CX
           DC  X'2000'    DA
           DC  X'2100'    DAU
           DC  X'4000'    PS
           DC  X'4100'    PSU
           DC  X'8000'    IS
           DC  X'8100'    ISU
$DALRTORG_END  EQU *
$DSORG_BEGIN EQU *
           DC  C'NTFN'    NOT FOUND
           DC  C'UNKN'
           DC  C'TR  '
           DC  C'VSAM'
           DC  C'TQ  '
           DC  C'TX  '
           DC  C'GS  '
DSORGPO    DC  C'PO  '
           DC  C'POU '
           DC  C'MQ  '
           DC  C'CQ  '
           DC  C'CX  '
           DC  C'DA  '
           DC  C'DAU '
           DC  C'PS  '
           DC  C'PSU '
           DC  C'IS  '
           DC  C'ISU '
$DSORG_END  EQU *
$FW4       DC  F'4'
* STATIC STORAGE
$DCBPO    DCB DDNAME=DCBPO,DSORG=PO,MACRF=(R)
$DCBPO_END  EQU *
$DCBPO_OPELST  OPEN  ($DCBPO,INPUT),MF=L,MODE=31
$DCBPO_OPELST_END EQU *
$DCBPO_CLSLST  CLOSE ($DCBPO),MF=L,MODE=31
$DCBPO_CLSLST_END EQU *
* DYNAMIC STORAGE
WS       DSECT
SAVEAREA   DS 18F
PARMADR    DS   A
RC         DS   F
DALRTORGTU@ DS  A   ADDRESS OF DALRTORGTU
DALDDNAMTU@ DS  A   ADDRESS OF DALDDNAMTU
DALDSNAMTU@ DS  A   ADDRESS OF DALDSNAMTU
STATSKEYW  DS   X
LABELKEYW  DS   X
NDISPKEYW  DS   X
IEFRBPTR   DS   A
```

```
IEFRB      DS   CL(RBLEN)
IEFTUPL    DS   49A
IEFTUNIT   DS   CL5ØØ
IEFTUNITE EQU  *        C1
           DS ØA
DCBPO      DS CL($DCBPO_END-$DCBPO)
           DS ØA
DCBPO_OPELST DS CL($DCBPO_OPELST_END-$DCBPO_OPELST)
           DS ØA
DCBPO_CLSLST DS CL($DCBPO_CLSLST_END-$DCBPO_CLSLST)
           DS ØA
BLDL_LIST  DS ØA
BL_ENTRY#  DS H        # ENTRIES IN LIST
BL_ENT_LEN DS H        LEN OF 1 ENTRY
BL_1       DS ØC
BL_MBR     DS CL8
BL_TTR     DS CL3
BL_K       DS CL1
BL_Z       DS CL1
BL_C       DS CL1
BL_USER    DS CL62
BL_1_END   EQU  *    END OF 1 ENTRY
*  ABEND EXIT IN LOW STORAGE
           DS ØA
@ABEND_EXIT    DC  AL1(X'8Ø'+X'11'),AL3(ABEND_EXIT)
ABEND_EXIT DS CL($ABEND_EXIT_END-$ABEND_EXIT)
*
           DS ØA
AX_PRMLST  DS CL16           ABEND EXIT PARMLIST
ABENDCODE  DS H
ABENDREASON DS H
*******   DON'T PUT ANYTHING BEHIND STACK
STACK      DS 18F
********
WS_FIX_END EQU *
HVPDYNA  CSECT
$ABEND_EXIT    EQU *
*          R13  STILL VALID HERE
           MVC  AX_PRMLST,Ø(1)
           LA   WORKØ1,4       IGNORE CODE
           STC  WORKØ1,3(Ø,RØ1)
           BR   R14
$ABEND_EXIT_END    EQU *
         LTORG
         IEFZB4DØ
         IEFZB4D2
         END
```

Herman Vierendeels
Systems Programmer (Belgium)

NEON Systems has integrated its Diplomat product with Shadow Direct, providing application integration between different customer relationship management and help desk applications, as well as direct access to System/390 data and transactions.

Diplomat enables EAI and B2B event-driven integration for different mainframe, Windows NT, and Unix applications in mixed cross-function environments.

With Diplomat, event-driven integration is streamlined among intra-office business units as well as among B2B and B2C interactions.

Shadow Direct provides access to System/390 CICS, IMS, VSAM, ADABAS, and DB2 data and transactions.

For further information contact:
NEON Systems, 14100 Southwest Freeway, #500 Sugarland, TX 77478, USA.
Tel: (281) 491 4200.
URL: http://www.neonsys.com.

* * *

Sybase has begun shipping its Enterprise Portal software for accessing information and business functions. The portal product promises continuous availability and integration of existing applications and data. Besides shipping the Enterprise Portal, the company is also working with a range of application providers, including Vignette, TIBCO, and Autonomy.

It has built-in support for portal systems management, claimed better mainframe integration with CICS and IMS, support for XML and Java, interfaces to key system management packages, support for major data sources and existing applications, such as SAP and Peoplesoft, load balancing, cluster support, and single sign-on security.

It can migrate connections over to back-up servers and it provides tools for content management, content retrieval, and application integration. The security framework protects the complete portal environment and integrates with third-party providers, while the management tools help reduce the resource requirements to support it.

For further information contact:
Sybase, 6475 Christie Avenue, Emeryville, CA 94608-9967, USA.
Tel: (510) 922 3500.
Sybase (UK), Sybase Court, Crown Lane, Maidenhead, Berks, SL6 8QX, UK.
Tel: (01628) 597100.
http://www.sybase.com.