80

DB2

June 1999

In this issue

3 Ageing your DB2 data

9 Maximum number of VSAM extents on DSNDB06

20 VSAM to DB2 conversion

24 Analysing the DSNZPARM load module

38 Driving and testing FIELDPROC – part 2

48 DB2 news

update

DB2 Update

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

Contributions

Articles published in *DB2 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*.

DB2 Update on-line

Code from *DB2 Update* can be downloaded from our Web site at http://www.xephon.com; you will need the user-id shown on your address label.

Editor

Robert Burgess

Disclaimer

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

Subscriptions and back-issues

A year's subscription to *DB2 Update*, comprising twelve monthly issues, costs £255.00 in the UK; $380.00 in the USA and Canada; £261.00 in Europe; £267.00 in Australasia and Japan; and £265.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1994 issue, are available separately to subscribers for £22.50 ($33.50) each including postage.

Printed in England.

Ageing your DB2 data

You can age your DB2 data using this REXX procedure. It prepares all the SQL CALL updates from the DB2 CATALOG table SYSIBM.SYSCOLUMNS. It looks for columns that are in DATE, TIME, or TIMESTAMP format. To use this tool, you must have compiled the DSNTEP2 sample program.

During the ageing phase of your DB2 applications, you need to age all the columns containing DATEs or TIMEs. With DB2, the use of a product is unnecessary, unless you have the date in CHAR, DECIMAL, INTEGER, or SMALLINT types.

On your test site, you copy or restore your tables from the production site, after reducing the size if necessary, and then you age all the DATEs. If the dates are in DATE, TIME, or TIMESTAMP format, you can use the SQL CALL update by adding a vector of days, and/or months, and/or years, and/or hours, and/or minutes, and/or seconds. (Time calculation is needed for financial or banking applications.)

If you receive the SQLCode 183 (see Figure 1) when you apply the SQL CALL updates, this is either because you have used a vector that is too large (see the limits shown in Figure 2), or a value that is too large. Therefore, if this message appears, it's because you have a date or a time exceeding the DB2 limits (these limits are different between DB2 versions).

To start this procedure, you must ALTLIB your REXX procedures library, LIBDEF your ISPF panels and tables (DDNAME ISPPLIB ISPTLIB ISPTABL), and type TSO %AGEDB2.

```
SQLCODE  183,
ERROR    AN ARITHMETIC OPERATION ON A DATE OR TIMESTAMP HAS A
           RESULT THAT IS NOT WITHIN THE VALID RANGE OF DATES
```

Figure 1: SQL Code 183

When adding	Limit
DAYS	999999
YEARS	999
MONTHS	9999
HOURS	9999999
MINUTES	999999999
SECONDS	99999999999

Figure 2: Limits when using the DB2 SQL update CALL

REXX PROCEDURE

```
/* REXX AGEDB2*/

"ISPEXEC VGET ZUSER"
PREF = SYSVAR("SYSPREF")
IF PREF = ' ' THEN PREF = ZUSER
"ISPEXEC ADDPOP ROW(2) COLUMN(1Ø)"
"ISPEXEC DISPLAY PANEL(AGINGDB2)"
CCPANEL = RC
"ISPEXEC REMPOP ALL"
IF CCPANEL > Ø THEN EXIT(Ø)
UPPER DAYS MONTHS YEARS HOURS MINUTES SECONDS TABLE CREATOR
CALL DSNTEP2
K = Ø
DO I = 1 TO P.Ø
  X = INDEX(P.I, '_|' )
  IF X > Ø THEN
     DO
        COL = WORD(P.I,2)
        TBC = WORD(P.I,4)
        TBN = WORD(P.I,6)
        TYP = WORD(P.I,8)
        CALL PREPSQL
     END
END
IF K = Ø THEN
  DO
     "ISPEXEC ADDPOP ROW(1Ø) COLUMN(1Ø)"
```

```
     "ISPEXEC DISPLAY PANEL(AGINGDB3)"
     "ISPEXEC REMPOP ALL"
     EXIT(Ø)
  END
SQLFILE = "'"||PREF||".SQL'"
X = MSG('OFF')
"FREE FI(SQL)"
X = SYSDSN(SQLFILE)
IF X <> 'DATASET NOT FOUND' THEN
  "DELETE "||SQLFILE
X = MSG('ON')
"ALLOC FI(SQL) DA("||SQLFILE||") NEW CATALOG UNIT(SYSDA)"
"EXECIO "||K||" DISKW SQL (STEM SQL."
"EXECIO Ø DISKW SQL (FINIS"
"FREE FI(SQL)"
"ISPEXEC EDIT DATASET("||SQLFILE||")"
EXIT(Ø)

DSNTEP2:
  X = MSG('OFF')
  "FREE FI(SYSIN,SYSPRINT)"
  X = MSG('ON')
  "ALLOC FI(SYSIN) NEW CATALOG UNIT(SYSDA)"
  "ALLOC FI(SYSPRINT) NEW CATALOG UNIT(SYSDA)"
  O =3
  S.1 = 'SELECT NAME, TBCREATOR, TBNAME, COLTYPE '
  S.2 = 'FROM SYSIBM.SYSCOLUMNS'
  S.3 = "WHERE COLTYPE IN ('DATE','TIMESTMP','TIME')"
  IF TABLE <> '*' THEN
     DO
        X = INDEX(TABLE, '%')
        IF X > Ø THEN
           DO
              O = O + 1
              S.O = "  AND TBNAME LIKE '"||TABLE||"'"
           END
        ELSE
           DO
              O = O + 1
              S.O = "  AND TBNAME = '"||TABLE||"'"
           END
     END
  IF CREATOR <> '*' THEN
     DO
        X = INDEX(CREATOR, '%')
        IF X > Ø THEN
           DO
              O = O + 1
              S.O = "  AND TBCREATOR LIKE '"||CREATOR||"'"
           END
```

```
         ELSE
            DO
               O = O + 1
               S.O = "  AND TBCREATOR = '"||CREATOR||"'"
            END
      END
   SQL1 = ''
   SQL2 = ''
   SQL3 = ''
   SQL4 = ''
   SQL5 = ''
   DO Q = 1 TO O
      CMD = "SQL"||STRIP(Q,,' ')||'="'||S.Q||'"'
      INTERPRET CMD
   END
   "ISPEXEC CONTROL DISPLAY LOCK"
   "ISPEXEC ADDPOP ROW(1Ø) COLUMN(1Ø)"
   "ISPEXEC DISPLAY PANEL(AGINGDB4)"
   "EXECIO "||O||" DISKW SYSIN (STEM S."
   "EXECIO Ø DISKW SYSIN (FINIS"
   QUEUE "RUN PROGRAM(DSNTEP2) PLAN(DSNTEP2) LIB('Your loadlib library')"
   QUEUE 'END'
   "DSN SYSTEM("||DB2ID||")"
   "ISPEXEC REMPOP ALL"
   IF RC > Ø THEN EXIT(8)
   "EXECIO * DISKR SYSPRINT (STEM P."
   "EXECIO Ø DISKR SYSPRINT (FINIS"
   "FREE FI(SYSIN,SYSPRINT)"
   RETURN(Ø)

PREPSQL:
   IF TYP = 'DATE' | TYP = 'TIMESTMP' | TYP = 'TIME' THEN
      DO
         K = K + 1
         SQL.K = 'UPDATE ' || STRIP(TBC,,' ')||'.'||STRIP(TBN,,' ')
         K = K + 1
         SQL.K = 'SET '||COL||'='||COL
         IF TYP <> 'TIME' THEN
            DO
               IF DAYS > Ø THEN
                  DO
                     K = K + 1
                     SQL.K = '           +'||DAYS|| ' DAYS'
                  END
               IF MONTHS > Ø THEN
                  DO
                     K = K + 1
                     SQL.K =    '           +'||MONTHS|| ' MONTHS'
                  END
               IF YEARS > Ø THEN
```

```
                  DO
                     K = K + 1
                     SQL.K =    '           +'||YEARS|| ' YEARS'
                  END
            END
         IF TYP = 'TIMESTMP' | TYP = 'TIME' THEN
            DO
               IF HOURS > Ø THEN
                  DO
                     K = K + 1
                     SQL.K =    '           +'||HOURS|| ' HOURS'
                  END
               IF MINUTES> Ø THEN
                  DO
                     K = K + 1
                     SQL.K =    '           +'||MINUTES|| ' MINUTES'
                  END
               IF SECONDES > Ø THEN
                  DO
                     K = K + 1
                     SQL.K =    '           +'||SECONDS|| ' SECONDS'
                  END
            END
            SQL.K = SQL.K||' ;'
      END
   RETURN(Ø)
```

ISPF PANELS

AGINGDB2 Panel

```
)ATTR /* PANEL : AGINGDB2 */
 + TYPE(TEXT) COLOR(WHITE)
)BODY WINDOW(46,17)
+TYPE THE DB2 SUBSYSTEM ===> _DB2ID+
+————————————————————————
+
+TYPE THE AGEING NUMBER OF :
+————————————————————————
+              DAYS ===> _DAYS  +
+            MONTHS ===> _MONTHS+
+             YEARS ===> _YEARS+
+             HOURS ===> _HOURS+
+           MINUTES ===> _MINUTES+
+           SECONDS ===> _SECONDS+
+
+             THE NAME OF :
+             ————————
```

```
+             TABLE ===> _TABLE             +
+           CREATOR ===> _CREATOR +
+                        (* POUR ALL)
)INIT
 &TABLE= '*'
 &CREATOR='*'
)PROC
  VER(&DAYS,RANGE,Ø,9999)
  VER(&MONTHS,RANGE,Ø,999)
  VER(&YEARS,RANGE,Ø,99)
  VER(&HOURS,RANGE,Ø,99999)
  VER(&MINUTES,RANGE,Ø,9999999)
  VER(&SECONDES,RANGE,Ø,99999999)
  VER(&TABLE,NB)
  VER(&CREATOR,NB)
)END
```

AGINGDB3 Panel

```
)ATTR /* PANEL : AGINGDB3 */
 + TYPE(TEXT) COLOR(WHITE)
)BODY WINDOW(26,1)
 NO OCCURENCE FOUND !!!
)INIT
)PROC
)END
```

AGINGDB4 Panel

```
)ATTR /* PANEL : AGINGDB4 */
 + TYPE(TEXT) COLOR(WHITE)
)BODY WINDOW(5Ø,8)
+
+ SENDING TO DB2 &DB2ID + THE REQUEST
+AS FOLLOWS :
+&SQL1
+&SQL2
+&SQL3
+&SQL4
+&SQL5
)INIT
)PROC
)END
```

Maximum number of VSAM extents on DSNDB06

SQL statements such as ‘ALTER INDEX index-name PRIQTY’ and ‘ALTER TABLE table-space-name PRIQTY’ are not allowed on the DB2 catalog for IBM-defined indexes or tablespaces. The method to enlarge a dataset that has too many extents, or is already fully extended, differs from the user-defined method.

THE PROBLEM

When an insert or update requires additional space, and the space is not available in the current tablespace or indexspace, DB2 issues the following message:

```
DSNP007I - DSNPmmmm - EXTEND FAILED FOR
           data-set-name.  RC=rrrrrrrr
           CONNECTION-ID=xxxxxxxx,
           CORRELATION-ID=yyyyyyyyyyyy
           LUWID-ID=logical-unit-of-work-id=token
```

ENLARGING A (FULLY) EXTENDED DATASET ON DSNDB06

For tablespaces, be sure there is a recent image copy of the DB2 catalog, to allow for recovery in case of failure during this procedure.

My quick procedure has the following steps:

1 Using the access method services, delete and define a shadow linear dataset with the characteristics of the active (original) dataset, but enlarging it with values of LISTCAT information (HI-A-RBA, HI-U-RBA, and EXTENT-NUMBER).

   For example, the name of the first index (DSNKDX01) on SYSIBM.SYSPACKDEP is:

   ```
   catname.DSNDBC.DSNDB06.DSNKDX01.I0001.A001
   ```

   and a shadow dataset name is:

   ```
   catname.DSNDBC.DSNDB06.DSNKDX01.S0001.A001
   ```

2 The DSN1COPY utility copies the dataset for a tablespace or indexspace to a shadow dataset.

3 The DB2 command ‘STOP DATASET(DSNDB06) SPACENAM(spacenam)’ is issued.

4 Using AMS, rename the original dataset name to the old dataset name. For example:

```
catname.DSNDBC.DSNDB06.DSNKDX01.O0001.A001
```

5 Using AMS, rename the shadow dataset name to the original dataset name.

6 The DB2 command ‘START DATASET(DSNDB06) SPACENAM(spacenam)’ is issued.

7 Using AMS, delete the old dataset name.

8 REORG and RUNSTATS table or indexspace (optional).

The procedure works when the subsystem is both active and inactive.

REXX PROCEDURE DB06

All of the above steps are included in the REXX procedure DB06. Starting DB06 is simple – you can find the DB2 datasets in the ISPF 3.4 environment. A scroll right shows the tracks and extents (XT) of these datasets (see Figure 1).

In my example, an index DSNKSX01 has 14 extents. From this position, type DB06 and press Enter. Figure 2 shows the parameter entry panel with the subsystem-id name. If you want reorg and runstats you must enter YES.

The DB06 procedure uses the JCL skeleton DB06S and shows the following JCL stream:

```
EDIT       SYSADM.DB06.TEMP                      Columns 00001 00072
Command ===>                                     Scroll ===> CSR
****************************** Top of Data ******************************
//SYSADMX JOB (ACCT#),'CATALOG TS/IX',
//             NOTIFY=SYSADM,REGION=4M,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*********************************************************************
//* Redefime TS/IX space on DB2 Catalog
//*                                                  Date: 4 Dec 1998
//* Description:                                Time: 1:07pm
//* ————————————————————————————
```

```
DSLIST - Datasets matching DSNCATN.DSNDBD.DSNDBØ6.DSN*          Row 37 of 62
Command ===>                                                 Scroll ===> PAGE

Command - Enter "/" to select action                Tracks %Used XT  Device
------------------------------------------------------------------------------
         DSNCATN.DSNDBD.DSNDBØ6.DSNKAXØ2.IØØØ1.AØØ1       2Ø    ?   2  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKAXØ3.IØØØ1.AØØ1       2Ø    ?   2  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKCXØ1.IØØØ1.AØØ1        4    ?   1  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKDXØ1.IØØØ1.AØØ1      156    ?   1  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKDXØ2.IØØØ1.AØØ1        7    ?   1  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKKXØ1.IØØØ1.AØØ1       16    ?   4  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKKXØ2.IØØØ1.AØØ1       12    ?   5  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKLXØ1.IØØØ1.AØØ1        2    ?   1  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKLXØ2.IØØØ1.AØØ1        8    ?   7  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKPXØ1.IØØØ1.AØØ1        1    ?   1  339Ø
db Ø6    DSNCATN.DSNDBD.DSNDBØ6.DSNKSXØ1.IØØØ1.AØØ1      225    ?  14  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKYXØ1.IØØØ1.AØØ1        2    ?   1  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNPPHØ1.IØØØ1.AØØ1        3    ?   1  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNSCXØ1.IØØØ1.AØØ1       45    ?   1  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNSDXØ1.IØØØ1.AØØ1       45    ?   1  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNSSHØ1.IØØØ1.AØØ1        1    ?   1  339Ø
```

Figure 1: Finding the DB2 datasets – ISPF 3.4

```
//* DB2 TS/IX: DSNCATN.DSNDBD.DSNDBØ6.DSNKSXØ1.IØØØ1.AØØ1
//* Volume   : MVSDB3
//* Type     : CYLINDER
//* Priqty   : 2
//* Records  : 2568
//* Secqty   : 1
//* Records  : 1284
//* Extent   : 14
//* ------------------------------------
//DELDEF  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//* ********************************************************************
//* DELETE/DEFINE SHADOW TS/IX SPACE
//* ********************************************************************
//SYSIN DD *
DELETE -
    (DSNCATN.DSNDBC.DSNDBØ6.DSNKSXØ1.SØØØ1.AØØ1/DBADMIN) -
    CATALOG(DSNCATN/DSNDEFPW)

    SET MAXCC = Ø
```

```
DSLIST - Datasets matching DSNCATN.DSNDBD.DSNDBØ6.DSN*          Row 37 of 62
Command ===>                                                 Scroll ===> PAGE

Command - Enter "/" to select action                 Tracks %Used XT Device
-------------------------------------------------------------------------------
         DSNCATN.DSNDBD.DSNDBØ6.DSNKAXØ2.IØØØ1.AØØ1       2Ø    ?   2  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKAXØ3.IØØØ1.AØØ1       2Ø    ?   2  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKCXØ1.IØØØ1.AØØ1        4    ?   1  339Ø
         DSNCATN.DSNDBD.DSNDBØ6.DSNKDXØ1.IØØØ1.AØØ1      156    ?   1  339Ø
         DSNCATN +-------------------------------------+    7    ?   1  339Ø
         DSNCATN |                                     |   16    ?   4  339Ø
         DSNCATN |                                     |   12    ?   5  339Ø
         DSNCATN |   Subsystem id:  DSNN               |    2    ?   1  339Ø
         DSNCATN |                                     |    8    ?   7  339Ø
         DSNCATN |   Reorg       :  YES                |    1    ?   1  339Ø
DBØ6     DSNCATN |   Runstat     :  YES                |  225    ?  14  339Ø
         DSNCATN |                                     |    2    ?   1  339Ø
         DSNCATN |   Enter:Continue         PF3:End    |    3    ?   1  339Ø
         DSNCATN |                                     |   45    ?   1  339Ø
         DSNCATN |                                     |   45    ?   1  339Ø
         DSNCATN |                                     |    1    ?   1  339Ø
                 +-------------------------------------+
```

Figure 2: The main parameter entry panel

```
 DEFINE CLUSTER -
       ( NAME(DSNCATN.DSNDBC.DSNDBØ6.DSNKSXØ1.SØØØ1.AØØ1) -
         LINEAR -
         REUSE -
         VOLUMES(MVSDB3) -
         MASTERPW(DBADMIN) -
         CONTROLPW(DBADMIN) -
         RECORDS(2568 1284) -
         SHAREOPTIONS(3 3) ) -
    DATA -
       ( NAME(DSNCATN.DSNDBD.DSNDBØ6.DSNKSXØ1.SØØØ1.AØØ1) -
         MASTERPW(DBADMIN) -
         CONTROLPW(DBADMIN) -
                           ) -
    CATALOG(DSNCATN/DSNDEFPW)
/*
//*-------------------------------------
//*-- DSN1COPY: ORIGINAL TS/IX IN SHADOW VSAM TS/IX
```

```
//*-----------------------------------
//DSN1CX EXEC PGM=DSN1COPY,COND=(4,LT)
//STEPLIB  DD DSN=DSN510.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=DSNCATN.DSNDBD.DSNDB06.DSNKSX01.I0001.A001,
//            DISP=SHR
//SYSUT2   DD DSN=DSNCATN.DSNDBD.DSNDB06.DSNKSX01.S0001.A001,
//            DISP=SHR
/*
//*-- STOP TS/IX SPACE ----------------
//STOPTS  EXEC PGM=IKJEFT01,COND=(4,LT)
//STEPLIB  DD DSN=DSN510.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  DSN SYSTEM(DSNN)

  -STOP DATABASE(DSNDB06) SPACENAM(DSNKSX01)
/*
//RENAMEO EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//* ********************************************************************
//* RENAME ORIGINAL TS/IX SPACE IN OLD TS/IX SPACE
//* ********************************************************************
//SYSIN DD *
  ALTER   -
  DSNCATN.DSNDBC.DSNDB06.DSNKSX01.I0001.A001 -
  NEWNAME -
     (DSNCATN.DSNDBC.DSNDB06.DSNKSX01.O0001.A001)

   ALTER   -
      DSNCATN.DSNDBD.DSNDB06.DSNKSX01.I0001.A001 -
   NEWNAME -
     (DSNCATN.DSNDBD.DSNDB06.DSNKSX01.O0001.A001)
/*
//RENAMES EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//* ********************************************************************
//* RENAME SHADOW TS/IX SPACE IN ORIGINAL TS/IX SPACE
//* ********************************************************************
//SYSIN DD *

  ALTER   -
     DSNCATN.DSNDBC.DSNDB06.DSNKSX01.S0001.A001 -
  NEWNAME -
    (DSNCATN.DSNDBC.DSNDB06.DSNKSX01.I0001.A001)

  ALTER   -
     DSNCATN.DSNDBD.DSNDB06.DSNKSX01.S0001.A001 -
```

```
  NEWNAME -
    (DSNCATN.DSNDBD.DSNDBØ6.DSNKSXØ1.IØØØ1.AØØ1)
/*
//*── START TS/IX SPACE ──────────────
//STARTS  EXEC PGM=IKJEFTØ1,COND=(4,LT)
//STEPLIB  DD DSN=DSN51Ø.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
    DSN SYSTEM(DSNN)

   -START DATABASE(DSNDBØ6) SPACENAM(DSNKSXØ1)
/*
//DELETE  EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//* ****************************************************
//* DELETE OLD TS/IX SPACE
//* ****************************************************
//SYSIN DD *
DELETE -
(DSNCATN.DSNDBC.DSNDBØ6.DSNKSXØ1.OØØØ1.AØØ1)
/*
//*── REORG TS/IX ──────────────────
//REORG   EXEC DSNUPROC,SYSTEM=DSNN,
//        UID='SYSADM.REORG',UTPROC=''
//STEPLIB DD DSN=DSN51Ø.SDSNLOAD,DISP=SHR
//SORTOUT DD DISP=(NEW,DELETE,CATLG),
//        UNIT=339Ø,
//        DSN=SYSADM.SORTOUT.D338Ø787,
//        SPACE=(CYL,(5Ø,5Ø),,,ROUND)
//SYSUT1  DD DISP=(NEW,DELETE,CATLG),
//        UNIT=339Ø,
//        DSN=SYSADM.SORTIN.D338Ø787,
//        SPACE=(CYL,(5Ø,5Ø),,,ROUND)
//SYSIN   DD *
REORG     INDEX  SYSIBM.DSNKSXØ1
           SORTDEVT 339Ø
           SORTNUM  5
 RUNSTATS  INDEX (SYSIBM.DSNKSXØ1)
/*
******************************** Bottom of Data **********************
```

COMPONENTS OF DB06

DB06 comprises:

- DB06 – the driver procedure
- DB06M – the main menu

- DB0600 – DB06 message
- DB06S – JCL skeleton.

DB06

```
/* REXX */
/* trace r */
parse upper arg dsn
db =word(translate(dsn,' ','.'),3)
if db ¬='DSNDBØ6'then do
   address ispexec
   zedsmsg = "See long message"
   zedlmsg = "This utility is valid only for DSNDBØ6 database"
   "setmsg msg(isrzØØ1)"
   Exit
end
address ispexec 'addpop row(1Ø) column(15)'
re='no'
ru='NO'
address ispexec "display panel(dbØ6m)"
if rc=8 then Exit
address ispexec 'rempop'
y=listdsi(dsn)
is = '('||dsn||')'
x=outtrap('var.')
address tso "listc" entries is allocation
typ=word(translate(word(var.8,1),' ','-'),3)
pri=word(translate(word(var.9,1),' ','-'),3)
sec=word(translate(word(var.1Ø,1),' ','-'),3)
ext=word(translate(word(var.12,4),' ','-'),3)
x=outtrap('off')
IF TYP='TRACK'
then do
   recp=(pri+sec*ext)*1Ø.7
   recp=recp%1
   recs=recp%2
end
IF TYP='CYLINDER'
then do
   recp=((pri*15)+(sec*15)*ext)*1Ø.7
   recp=recp%1
   recs=recp%2
end
idsn = sysdsname
sdsn = sysdsname
sdsn1= OVERLAY('DSNDBC',sdsn,POS('DSNDBD',sdsn))
sdsnd= OVERLAY('SØØØ1',sdsn,POS('IØØØ1',sdsn))
```

```
sdsnc= OVERLAY('DSNDBC',sdsnd,POS('DSNDBD',sdsnd))
odsnd= OVERLAY('O0001',sdsn,POS('I0001',sdsn))
odsnc= OVERLAY('DSNDBC',odsnd,POS('DSNDBD',odsnd))
cat=word(translate(idsn,' ','.'),1)
db =word(translate(idsn,' ','.'),3)
sp =word(translate(idsn,' ','.'),4)
svol = sysvolume
title='Redefine TS/IX space on DB2 Catalog'
date=date()
time=time(c)
x=msg("off")
user=userid()
suf='D'||right(date('D'),3,'0')||right(time('M'),4,'0')
tempfile=userid()||'.DB06.TEMP'
address tso
"delete '"tempfile"'"
"free dsname('"tempfile"')"
"free ddname(ispfile)"
"free attrlist(formfile)"
"attrib formfile blksize(800) lrecl(80) recfm(f b) dsorg(ps)"
"alloc ddname(ispfile) dsname('"tempfile"')",
       "new using (formfile) unit(3390) space(1 1) cylinders"
address ispexec
"ftopen"
"ftincl db06s"
"ftclose"
zedsmsg = "JCL shown"
zedlmsg = "JCL for redefine ts/ix shown"
"setmsg msg(isrz001)"
"edit dataset('"tempfile"')"
x=msg("on")
exit
```

DB06M

```
)attr default(%+_)
   / type(input)  color(yellow)    hilite(uscore)  intens(high)
   ] type(text)   color(turquoise)                 intens(high)
   | type(text)   color(white)     hilite(reverse) intens(high)
   # type(text)   color(green)                     intens(high)
   @ type(text)   color(red)                       intens(high)
)body window(35,11)
|
| +                              |
| #Subsystem id: /db2 +          |
| +                              |
| +Reorg       : /re +           |
| +Runstat     : /ru +           |
| +                              |
```

```
| ]Enter:Continue         PF3:End|
| +                              |
|
)init
)proc
  ver(&db2,nb,msg=db0600l)
  if (.pfkey = pf03) &pf3 = exit
  vput (db2t) profile
)end
```

DB0600

```
DB0600l 'Empty field'                    .WINDOW=NORESP .ALARM = YES
'Field must be filled.'
```

DB06S

```
)TBA 72
)CM ------------------------------------------------------------------------
)CM SKELETON TO GENERATE JCL TO REDEFINE TS/IX SPACE ON DB2 CATALOG  -
)CM ------------------------------------------------------------------------
//&user.X JOB (ACCT#),'CATALOG TS/IX',
//             NOTIFY=&user,REGION=4M,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//* ********************************************************************
//* &title
//*                                                    Date: &date
//* Description:                                       Time: &time
//* --------------------------------------------------------------------
//* DB2 TS/IX: &idsn
//* Volume   : &svol
//* Type     : &typ
//* Priqty   : &pri
//* Records  : &recp
//* Secqty   : &sec
//* Records  : &recs
//* Extent   : &ext
//* --------------------------------------------------------------------
//DELDEF  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//* ********************************************************************
//* DELETE/DEFINE SHADOW TS/IX SPACE
//* ********************************************************************
//SYSIN DD *
 DELETE -
    (&sdsnc/DBADMIN) -
    CATALOG(&cat/DSNDEFPW)
```

```
)BLANK 1
    SET MAXCC = Ø
)BLANK 1
 DEFINE CLUSTER -
       ( NAME(&sdsnc) -
         LINEAR -
         REUSE -
         VOLUMES(&svol) -
         MASTERPW(DBADMIN) -
         CONTROLPW(DBADMIN) -
         RECORDS(&recp &recs) -
         SHAREOPTIONS(3 3) ) -
    DATA -
       ( NAME(&sdsnd) -
         MASTERPW(DBADMIN) -
         CONTROLPW(DBADMIN) -
                              ) -
    CATALOG(&cat/DSNDEFPW)
/*
//*-------------------------------------------------------------------
//*-- DSN1COPY: ORIGINAL TS/IX IN SHADOW VSAM TS/IX
//*-------------------------------------------------------------------
//DSN1CX EXEC PGM=DSN1COPY,COND=(4,LT)
//STEPLIB  DD DSN=DSN51Ø.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=&idsn,
//            DISP=SHR
//SYSUT2   DD DSN=&sdsnd,
//            DISP=SHR
/*
//*-- STOP TS/IX SPACE ----------------
//STOPTS  EXEC PGM=IKJEFTØ1,COND=(4,LT)
//STEPLIB  DD DSN=DSN51Ø.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
   DSN SYSTEM(&db2)
)BLANK 1
   -STOP DATABASE(&db) SPACENAM(&sp)
/*
//RENAMEO EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//* ********************************************************************
//* RENAME ORIGINAL TS/IX SPACE IN OLD TS/IX SPACE
//* ********************************************************************
//SYSIN DD *
 ALTER   -
    &sdsn1 -
 NEWNAME -
   (&odsnc)
```

```
)BLANK 1
 ALTER   -
    &sdsn -
 NEWNAME -
   (&odsnd)
/*
//RENAMES EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//* ********************************************************************
//* RENAME SHADOW TS/IX SPACE IN ORIGINAL TS/IX SPACE
//* ********************************************************************
//SYSIN DD *
 ALTER   -
    &sdsnc -
 NEWNAME -
   (&sdsn1)
)BLANK 1
 ALTER   -
    &sdsnd -
 NEWNAME -
   (&sdsn)
/*
//*-- START TS/IX SPACE ---------------
//STARTS  EXEC PGM=IKJEFTØ1,COND=(4,LT)
//STEPLIB  DD DSN=DSN51Ø.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
   DSN SYSTEM(&db2)
)BLANK 1
   -START DATABASE(&db) SPACENAM(&sp)
/*
//DELETE  EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//* ****************************************************
//* DELETE OLD TS/IX SPACE
//* ****************************************************
//SYSIN DD *
 DELETE -
    (&odsnc)
/*
)SEL &re = YES
//*-- REORG TS/IX ----------------------
//REORG   EXEC DSNUPROC,SYSTEM=&db2,
//        UID='&user..REORG',UTPROC=''
//STEPLIB DD DSN=DSN51Ø.SDSNLOAD,DISP=SHR
//SORTOUT DD DISP=(NEW,DELETE,CATLG),
//        UNIT=339Ø,
//        DSN=&user..SORTOUT.&suf,
```

```
//        SPACE=(CYL,(3Ø,3Ø),,,ROUND)
//SYSUT1  DD DISP=(NEW,DELETE,CATLG),
//        UNIT=339Ø,
//        DSN=&user..SORTIN.&suf,
//        SPACE=(CYL,(3Ø,3Ø),,,ROUND)
//SYSIN   DD *
REORG     INDEX  SYSIBM.&sp
          SORTDEVT 339Ø
          SORTNUM  5
)SEL &ru = YES
RUNSTATS  INDEX (SYSIBM.&sp)
)ENDSEL
/*
)ENDSEL
```

Check the MVS catalog control password (DSNDEFPW) and DB2 directory and catalog password (DBADMIN). I use default passwords from installation panel DSNTIPP.

Bernard Zver
Database Administrator
Informatika Maribor (Slovenia)

VSAM to DB2 conversion

This article shows how to use DFSORT, REXX, and DB2-load syntax such as NULLIF to help convert VSAM to DB2 simply and at low cost.

```
//TSHVRA  JOB (),,TIME=144Ø,NOTIFY=&SYSUID,REGION=45ØØK,
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//* USE |REXX    TO EASE VSAM-DB2 CONVERSION
//*     |NULLIF
//*
//* STRUCTURE VSAMFILE
//*   Ø1 VSAM_RECORD,
//*    Ø2 OTHERA          CHAR(..)
//*    Ø2 LEGISMV         PIC'(    4)9',
//*     EG 8Ø81 SHOULD BECOME NJSESS1=198Ø  NJSESS2=1981
//*     EG ØØ81 SHOULD BECOME NJSESS1=ØØØØ  NJSESS2=1981
//*     EG 96ØØ SHOULD BECOME NJSESS1=1996  NJSESS2=ØØØØ
//*    Ø2 OTHERB          CHAR(..)
//*    Ø2 DVERGAD         PIC'(    6)9',
//*     EG 8ØØ5Ø6 SHOULD BECOME 198Ø-Ø5-Ø6
```

```
//*          ØØØØØØ SHOULD BECOME NULL
//*     (ACTIVE DB2 LOCAL DATE EDIT ROUTINE DSNXVDTX
//*       CONVERTS 6 DIGIT DATE TO ISO-DATE )
//*    Ø3 OTHERC  CHAR(..);
//DBCRT EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
 DSN SYSTEM(DSNT)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA41) -
      LIB('DSN41Ø.RUNLIB.LOAD')
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 DROP TABLE  TSHVR.BROL;
 COMMIT;
 CREATE TABLE TSHVR.BROL
  (
   OTHERA                          CHAR(6),
   NJSESS1                         SMALLINT NOT NULL,
   NJSESS2                         SMALLINT NOT NULL,
   OTHERB                          CHAR(6),
   DVERGAD                         DATE,
   OTHERC                          CHAR(6)
  ) ;
 COMMIT;
/*
//*
//RXCRT    EXEC PGM=IEBGENER,COND=(Ø,LT)
//SYSIN    DD DUMMY
//SYSUT2   DD DSN=&&REXX(MEM1),
//         DISP=(,PASS),UNIT=WORK,
//         DCB=(LRECL=8Ø,RECFM=FB,BLKSIZE=8ØØØ),
//         SPACE=(TRK,(1,1,1))
//SYSPRINT  DD  SYSOUT=*
//SYSUT1   DD *,DLM=##
 /* REXX */
 DEBUG=255
 RC=Ø
 MYRC=Ø
 NULL=X2C('ØØ')
 ADDRESS MVS "EXECIO 1 DISKR VSAMFILE (STEM LINES. "
 DO WHILE RC=Ø
  IF DEBUG>Ø THEN SAY LINES.1
  PARSE VAR LINES.1 1 REST1 7 NJSESS1 9 NJSESS2 11 REST2
  IF NJSESS1='' THEN NJSESS1=Ø
  IF NJSESS2='' THEN NJSESS2=Ø
  IF NJSESS1¬=Ø THEN NJSESS1=NJSESS1+19ØØ
  IF NJSESS2¬=Ø THEN NJSESS2=NJSESS2+19ØØ
  NJSESS1=D2X(NJSESS1)
  NJSESS2=D2X(NJSESS2)
  NJSESS1=RIGHT(X2C(NJSESS1),2,NULL)
```

```
  NJSESS2=RIGHT(X2C(NJSESS2),2,NULL)
  IF LENGTH(NJSESS1)¬=2 THEN DO
   SAY  NJSESS1 LENGTH(NJSESS1)
   RC=16
   SIGNAL L_EXIT
  END
  IF LENGTH(NJSESS2)¬=2 THEN DO
   SAY  NJSESS2 LENGTH(NJSESS2)
   RC=16
   SIGNAL L_EXIT
  END
  OUTLINES.1=REST1||NJSESS1||NJSESS2||REST2
  IF DEBUG>Ø THEN SAY OUTLINES.1
  ADDRESS MVS "EXECIO 1 DISKW OUT      (STEM OUTLINES. "
  ADDRESS MVS "EXECIO 1 DISKR VSAMFILE (STEM LINES. "
 END /*DO WHILE RC=Ø*/
 IF RC=2 THEN RC=Ø /*EOF*/
 L_EXIT:
 EXIT(RC)
##
//*
//TRNSFRM  EXEC PGM=IKJEFTØ1,PARM='MEM1',COND=(Ø,LT)
//STEPLIB  DD DISP=SHR,DSN=SYS2.CEE.SCEERUN
//         DD DSN=IPO97.KVVCDR.LOAD,DISP=SHR
//SYSPROC  DD DSNAME=*.RXCRT.SYSUT2,DISP=(OLD,DELETE)
//SYSEXEC  DD DISP=SHR,DSN=BFS.LS12Ø.SBFSPROC
//*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   DUMMY
//VSAMFILE DD *
OTHERA0088OTHERBØØØØØØOTHERC
OTHERA88ØØOTHERB88Ø616OTHERC
OTHERAØØØØOTHERBØØØØØØOTHERC
/*
//OUT DD DSN=&&OUT,
//        DISP=(NEW,PASS),
//        RECFM=FB,
//        LRECL=28,
//        BLKSIZE=Ø,
//        DSORG=PS,
//        SPACE=(TRK,(1,1),RLSE)
//*
//LOAD    EXEC DSNUPROC,PARM='DSNT,TSHVRA',COND=(Ø,LT)
//SYSUT1   DD  UNIT=SYSDA,SPACE=(4ØØØ,(2Ø,2Ø),,,ROUND)
//SORTLIB  DD  DSN=SYS2.SORTLIB,DISP=SHR
//SORTOUT  DD  UNIT=SYSDA,SPACE=(4ØØØ,(2Ø,2Ø),,,ROUND)
//SORTWKØ1 DD  UNIT=SYSDA,SPACE=(4ØØØ,(2Ø,2Ø),,,ROUND)
//SORTWKØ2 DD  UNIT=SYSDA,SPACE=(4ØØØ,(2Ø,2Ø),,,ROUND)
//SORTWKØ3 DD  UNIT=SYSDA,SPACE=(4ØØØ,(2Ø,2Ø),,,ROUND)
//SORTWKØ4 DD  UNIT=SYSDA,SPACE=(4ØØØ,(2Ø,2Ø),,,ROUND)
//DSNTRACE DD  SYSOUT=*
```

```
//SYSERR   DD  UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)
//SYSDISC  DD  SYSOUT=X
//SYSMAP   DD  SYSOUT=X
//LOADIN   DD  DSN=&&OUT,DISP=(OLD,DELETE)
/*
//SYSIN    DD  *
 LOAD DATA INDDN(LOADIN) RESUME(YES) LOG(NO)
  INTO TABLE TSHVR.BROL
    (
 OTHERA  POSITION(001) CHAR(06),
 NJSESS1 POSITION(007) SMALLINT           ,
 NJSESS2 POSITION(009) SMALLINT           ,
 OTHERB  POSITION(011) CHAR(06),
 DVERGAD POSITION(017) DATE EXTERNAL(08) NULLIF((17:22)='000000'),
 OTHERB  POSITION(023) CHAR(06)
     )
  ENFORCE CONSTRAINTS MAPDDN SYSMAP
/*
//*
//SHOW     EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//* HVNTEP2 IS ADAPTED DSNTEP2 SAMPLE PROGRAM
//STEPLIB DD DISP=SHR,DSN=SYS2.CEE.SCEERUN
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD *
  DSN SYSTEM(DSNT)
  RUN  PROGRAM(HVNTEP2) PLAN(HVNTEP2) -
       PARMS('/ALIGN(LHS)')           -
       LIB('PRJSGT.DB2.LOAD')
  END
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 SELECT * FROM TSHVR.BROL;
/*
//
//
```

OUTPUT FROM HVNTEP2

```
    +-------------------------------------------------------------------------+
    | OTHERA |  NJSESS1  |  NJSESS2  | OTHERB |  DVERGAD   | OTHERC |
    +-------------------------------------------------------------------------+
  1_| OTHERA |         0 |      1988 | OTHERC | ?          | ?      |
  2_| OTHERA |      1988 |         0 | OTHERC | 1988-06-16 | ?      |
  3_| OTHERA |         0 |         0 | OTHERC | ?          | ?      |
    +-------------------------------------------------------------------------+
```

Herman Vierendeels
Systems Programmer (Belgium)

Analysing the DSNZPARM load module

ZPARMREE ASSEMBLY is a program developed for DB2 for OS/390, Version 4 and Version 5. It analyses the DSNZPARM load module and creates the originating assembly macro input. It is designed to analyse the content of DSNZPARM should the creating job be lost.

To work properly, the source must be assembled using the same version of DB2 as the DSNZPARM load module.

The original source of ZPARMREE ASSEMBLY was taken from *Reverse engineering the ZPARM*, by Larry Prestosa, *DB2 Update*, Issue 38, December 1995 and Issue 39, January 1996.

SOURCE CODE

```
// JOB ,MSGCLASS=T,NOTIFY=,COND=(0,NE)
//A      EXEC  PGM=ASMA90,PARM='OBJECT,NODECK,ESD,NORLD,FLAG(SUBSTR)'
ZPARMV4  TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               RECONSTRUCT TSO INSTALL INPUT'
* ZPARMV4 :  LIST DSNZPARM MACRO VALUES
* FUNCTION : THIS PROGRAM READS THE DSNZPARM LOAD MODULE AND
*            GENERATES SOURCE DSNZPARM MACRO SIMILAR TO
*            SAMPLE JOB DSNTIJUZ WHICH ASSEMBLES/LINKS THE ZPARM
*            The program must be compiled with the same macro library
*            version as the DSNZPARM to be analysed.
* VERSION  : DB2 VERSION 4,5
* JCL      : SAMPLE JCL TO X THIS PROGRAM IS SHOWN BELOW
*    //X       EXEC PGM=ZPARMV4,COND=(4,LT),
*    //        PARM='DSNZPARM'             <- NAME OF ZPARM
*    //STEPLIB  DD DSN=MY.PROGRAM.LOAD,    <- SYSLMOD OF THIS PGM
*    //         DD DISP=SHR
*    //DSNLOAD  DD DSN=MY.TSO.DSNEXIT,     <- YOUR SHOP'S DB2 EXIT
*    //         DD DISP=SHR
*    //SYSPRINT DD SYSOUT=*
*    //SNAPDUMP DD SYSOUT=*
* PSEUDOCODE:
*    INITIALIZATION
*      - GET ZPARM NAME FROM PARMLIST
*      - OPEN FILES
*      - PRINT HEADER LINES
*    MAINLINE
*      - LOAD DSNZPARM LOAD MODULE
*      - FORMAT DSN6SPRM CONTROL BLOCK
*      - FORMAT DSN6ARVP CONTROL BLOCK
```

```
*      - FORMAT DSN6LOGP CONTROL BLOCK
*      - FORMAT DSN6SYSP CONTROL BLOCK
*      - FORMAT DSN6FAC  CONTROL BLOCK
*    FINALIZATION
*      - CLOSE FILES
*        TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               MACRO DSN6ARVP               '
*        COPY  DSN6ARVP
*        TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               MACRO DSN6ENV                '
*        COPY  DSN6ENV
*        TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               MACRO DSN6FAC                '
*        COPY  DSN6FAC
*        TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               MACRO DSN6LOGP               '
*        COPY  DSN6LOGP
*        TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               MACRO DSN6SPRM               '
*        COPY  DSN6SPRM
*        TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               MACRO DSNDSPRM               '
*        COPY  DSNDSPRM
*        TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               MACRO DSN6SYSP               '
*        COPY  DSN6SYSP
*        TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               MACRO DSN6GRP                '
*        COPY  DSN6GRP
         TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               ANALYSE STARTUP              '
         LCLA  &ZPRMLNE
ZPARMV4 START , X'6A10'
&ZPRMLNE SETA  133
         USING ZPARMV4,R15
ZPARMV4 AMODE 31
ZPARMV4 RMODE 24
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5                          BASE FOR CONSTANTS
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
```

```
R15      EQU   15
         STM   R14,R12,12(R13)    STANDARD LINKAGE CONVENTION
         LR    R12,R15
         LR    R11,R15
         A     R11,=A(4Ø96)
         DROP  R15
         USING ZPARMV4,R12,R11
         LR    R1Ø,R13
         L     R13,=A(SAVEAREA)
         USING SAVEAREA,R13
         ST    R13,8(,R1Ø)
         ST    R1Ø,4(,R13)
         B     INITIALZ           GO AROUND EYECATCHER/SAVEAREA
         DS    ØD
EYECATCH DC    CL9'ZPARMV4'
         DC    CL9'&SYSDATE.'
         DC    CL9'&SYSTIME.'
         DC    CL6'LOEBEN'
         LTORG
DSNZPARM DSECT
         PRINT GEN
         TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.1Ø.98               *
               DSNZPARM EXAMPLE COMPILE '
    DSN6ENV    MVS=XA
    DSN6SPRM   RESTART,ALL,                                          X
               ABEXP=YES,                                            X
               ABIND=YES,                                            X
               AUTH=YES,                                             X
               AUTHCACH=1Ø24,                                        X
               BINDNV=BINDADD,                                       X
               BMPTOUT=77,                                           X
               CATALOG=DBAF,                                         X
               CDSSRDEF=1,                                           X
               CHGDC=NO,                                             X
               DBCHK=YES,                                            X
               DECDIV3=NO,                                           X
               DEFIXTP=2,                                            X
               DEFLTID=NOUSER,                                       X
               DESCSTAT=NO,                                          X
               DLITOUT=Ø,                                            X
               DSMAX=6ØØØ,                                           X
               EDMPOOL=28ØØØ,                                        X
               EDPROP=NO,                                            X
               HOPAUTH=YES,                                          X
               IRLMAUT=YES,                                          X
               IRLMPRC=DØØ1IRLM,                                     X
               IRLMSID=DJØ1,                                         X
               IRLMRWT=6Ø,                                           X
               IRLMSWT=3ØØ,                                          X
               MAXRBLK=1ØØØØ,                                        X
               NUMLKTS=1ØØØ,                                         X
               NUMLKUS=1ØØØØ,                                        X
```

```
            RECALL=YES,                                          X
            RECALLD=12Ø,                                         X
            RETLWAIT=NO,                                         X
            RGFCOLID=DSNRGCOL,                                   X
            RGFDBNAM=DSNRGFDB,                                   X
            RGFDEDPL=NO,                                         X
            RGFDEFLT=ACCEPT,                                     X
            RGFESCP=,                                            X
            RGFFULLQ=YES,                                        X
            RGFINSTL=NO,                                         X
            RGFNMORT=DSN_REGISTER_OBJT,                          X
            RGFNMPRT=DSN_REGISTER_APPL,                          X
            RRULOCK=NO,                                          X
            SEQCACH=BYPASS,                                      X
            SEQPRES=NO,                                          X
            SITETYP=LOCALSITE,                                   X
            SRTPOOL=5ØØØ,                                        X
            SYSADM=S1ØØ447,                                      X
            SYSADM2=S1ØØ447,                                     X
            SYSOPR1=SYSOPR,                                      X
            SYSOPR2=SYSOPR,                                      X
            UTIMOUT=6
DSN6ARVP    ALCUNIT=CYL,                                         X
            ARCWRTC=(1,3,4),                                     X
            ARCWTOR=NO,                                          X
            ARCPFX1=DØØ1.ARCHLOG1,                               X
            ARCPFX2=DØØ1.ARCHLOG2,                               X
            ARCRETN=21,                                          X
            BLKSIZE=28672,                                       X
            CATALOG=YES,                                         X
            COMPACT=NO,                                          X
            PRIQTY=5Ø,                                           X
            PROTECT=NO,                                          X
            QUIESCE=5,                                           X
            SECQTY=5Ø,                                           X
            TSTAMP=NO,                                           X
            MSVGP=SVGP1,                                         X
            MSVGP2=,                                             X
            UNIT=AC1Ø,                                           X
            UNIT2=AC1Ø
DSN6LOGP    DEALLCT=(Ø),                                         X
            INBUFF=28,                                           X
            MAXARCH=5ØØ,                                         X
            MAXRTU=2,                                            X
            OUTBUFF=4ØØ,                                         X
            TWOACTV=YES,                                         X
            TWOARCH=YES,                                         X
            WRTHRSH=2Ø
DSN6SYSP    AUDITST=NO,                                          X
            CONDBAT=2Ø,                                          X
            CTHREAD=8Ø,                                          X
            DLDFREQ=5,                                           X
```

```
                IDBACK=40,                                              X
                IDFORE=50,                                              X
                LOGLOAD=90000,                                          X
                MAXDBAT=20,                                             X
                MON=NO,                                                 X
                MONSIZE=8192,                                           X
                RLF=NO,                                                 X
                RLFTBL=01,                                              X
                RLFERR=NOLIMIT,                                         X
                RLFAUTH=SYSIBM,                                         X
                ROUTCDE=(1),                                            X
                SMFACCT=(1,2,3,7,8),                                    X
                SMFSTAT=YES,                                            X
                STATIME=30,                                             X
                STORMXAB=0,                                             X
                STORPROC=D001SPAS,                                      X
                STORTIME=180,                                           X
                TRACSTR=NO,                                             X
                TRACTBL=16
    DSN6FAC     DDF=AUTO,                                               X
                CMTSTAT=ACTIVE,                                         X
                IDTHTOIN=0,                                             X
                RESYNC=2,                                               X
                RLFERRD=NOLIMIT
    DSN6GRP     DSHARE=YES,                                             X
                GRPNAME=DBAF,                                           X
                MEMBNAME=D001
          TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
                ANALYSE DSNZPARM            '
ZPARMV4  CSECT
******** I N I T I A L I Z E
INITIALZ DS    0H
         BAS   R14,GETPRTN                GET PARMLIST
         BAS   R14,INITRTN                DO INITIALIZE VALUES
         BAS   R14,HDRLRTN              DO PRINT HEADER LINES
******** M A I N L I N E
MAINLINE DS    0H
** LOAD DSNZPARM IN VIRTUAL STORAGE
         LOAD  EPLOC=PARMVAL,LOADPT=ZPARMPTR,DCB=LOAD
         LTR   R15,15
         BNZ   ABEND100
         LR    R15,R0
         LA    R15,0(,R15)
         ST    R15,ZPARMPTR
         LR    R7,R15
*        N     R1,=A(X'00FFFFFF')      REMOVE HIGH ORDER BYTE
         LA    R1,0(,R1)               REMOVE HIGH ORDER BYTE
         ST    R1,ZPARMPTR+4           SAVE LENGTH
         TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               ANALYSE DSN6SPRM            '
         L     R7,ZPARMPTR
         USING DSN6SPRM,R7               ESTABLISH ADDRESSABILITY
```

```
         LA    RØ,4
         LA    R1,255(,R7)
         CLC   =CL8'DSN6SPRM',4(R7)
         BE    *+12
         BXLE  R7,RØ,*-1Ø
         B     ABEND19Ø
         L     R7,Ø(,R7)
         L     R2,=A(DSN6SPRM)         SECTION TO BE ANALYSED
         CLC   SPRMID,SPRMID-DSN6SPRM(R2)
         BNE   ABEND1Ø1
         CLC   SPRMEYE,SPRMEYE-DSN6SPRM(R2)
         BNE   ABEND1Ø2                SECTION DSN6SPRM NOT FOUND
** DISPLAY ZPARM NAME AND ASSEMBLY DATE
         MVC   ZPRMLINE(13),=C'MODULE NAME: '
         MVC   ZPRMLINE+13(8),PARMVAL
         MVC   ZPRMLINE+3Ø(8),=CL8'VERSION'
         MVC   ZPRMLINE+38(L'SPRMLVL),SPRMLVL
         MVC   ZPRMLINE+5Ø(15),=C'ASSEMBLY DATE: '
         MVC   ZPRMLINE+65(L'SPRMDATE),SPRMDATE
         BAS   R14,ZWRTRTN             DO PRINT LINE
         MVC   ZPRMLINE,=(&ZPRMLNE)C'-'
         BAS   R14,WRITRTN
*> FORMAT DSN6ENV ****************************************************
         MVC   ZPRMCLØ5(Ø8),=C'DSN6ENV '
*> MVS       - MVS 37Ø OR XA
         MVC   ZPRMCL16(Ø4),=C'MVS='       FIELD LITERAL
         MVC   ZPRMCL16+Ø4(Ø3),SPRMMVS     GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE            FIND FIRST BLANK
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> FORMAT DSN6SPRM ***************************************************
         MVC   ZPRMCLØ5(Ø8),=C'DSN6SPRM'
*> RESTART   - AUTOSTARTED DATABASE/TABLESPACE
         MVC   ZPRMCL16(Ø7),=C'RESTART'    FIELD LITERAL
         TRT   ZPRMCL16,TRTABLE            FIND FIRST BLANK
         MVI   Ø(1),C','                   PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ALL       - DATABASE LIST
         MVC   ZPRMCL16(Ø3),=C'ALL'        FIELD LITERAL
         TRT   ZPRMCL16,TRTABLE            FIND FIRST BLANK
         MVI   Ø(1),C','                   PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ABEXP     - ALLOW/DISALLOW EXPLAIN DURING AUTOBIND
         MVC   ZPRMCL16(Ø6),=C'ABEXP='
         TM    SPRMMISZ,B'1ØØØØØØØ'
         BNO   NOMISZ1
         MVC   ZPRMCL16+Ø6(Ø3),=C'YES'
         B     YESMISZ1
NOMISZ1  MVC   ZPRMCL16+Ø6(Ø3),=C'NO '
YESMISZ1 TRT   ZPRMCL16,TRTABLE            FIND FIRST BLANK
         MVI   Ø(1),C','                   PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ABIND     - AUTOBIND ACTIVATED/DEACTIVATED
```

```
         MVC   ZPRMCL16(Ø6),=C'ABIND='
         CLI   SPRMABN,C'E'
         BNE   NOABIND
         MVC   ZPRMCL16+Ø6(Ø3),=C'YES'
         B     YESABIND
NOABIND  MVC   ZPRMCL16+Ø6(Ø3),=C'NO '
YESABIND TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ALPOOLX  - ALLOCATION POOL EXTENSION SIZE
         MVC   ZPRMCL16(Ø8),=C'ALPOOLX='  FIELD LITERAL
         ICM   R9,B'1111',SPRMTXS
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D
         OI    ZPRMCL16+22,X'FØ'
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> AUTH      - AUTHORIZATION ENABLED/DISABLED
         MVC   ZPRMCL16(Ø5),=C'AUTH='
         CLI   SPRMAUTH,C'E'
         BNE   NOAUTH
         MVC   ZPRMCL16+Ø5(Ø3),=C'YES'
         B     YESAUTH
NOAUTH   MVC   ZPRMCL16+Ø5(Ø3),=C'NO '
YESAUTH  TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> AUTHCACH - AUTHORIZATION CACHE
         MVC   ZPRMCL16(Ø9),=C'AUTHCACH=' FIELD LITERAL
         SR    R9,R9
         ICM   R9,B'ØØ11',SPRMAUCA
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(15),D
         OI    ZPRMCL16+23,X'FØ'
         MVC   ZEROHOLD,ZPRMCL16+Ø9       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø9(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> BINDNV -    BIND OR BINDADD AUTHORITY  FOR DIFFERENT VERSION
         MVC   ZPRMCL16(Ø7),=C'BINDNV='
         MVC   ZPRMCL16+Ø7(Ø8),SPRMBNVA
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> BMPTOUT - IMS BMP TIMEOUT FACTOR
         MVC   ZPRMCL16(Ø8),=C'BMPTOUT='  FIELD LITERAL
```

```
         SR    R9,R9
         ICM   R9,B'ØØ11',SPRMBMP
         CVD   R9,D                      CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D
         OI    ZPRMCL16+22,X'FØ'
         MVC   ZEROHOLD,ZPRMCL16+Ø8      MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN              DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD  MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
         MVI   Ø(1),C','                 PLUG COMMA HERE
         BAS   R14,ZWRTRTN            DO PRINT LINE
*> CATAGALOG - VSAM CATALOG NAME
         MVC   ZPRMCL16(Ø8),=C'CATALOG='
         LA    R8,SPRM
         A     R8,SPRMVCOF
         MVC   ZPRMCL16+Ø8(Ø8),12(R8)
         TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
         MVI   Ø(1),C','                 PLUG COMMA HERE
         BAS   R14,ZWRTRTN            DO PRINT LINE
*> CHGDC    - CHANGE DATA CAPTURE ACT/DEACT
         MVC   ZPRMCL16(Ø6),=C'CHGDC='
         TM    SPRMMISC,B'ØØØØ1ØØØ'
         BNO   NOMISC5
         MVC   ZPRMCL16+Ø6(Ø3),=C'YES'
         B     YESMISC5
NOMISC5  MVC   ZPRMCL16+Ø6(Ø3),=C'NO '
YESMISC5 TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
         MVI   Ø(1),C','                 PLUG COMMA HERE
         BAS   R14,ZWRTRTN            DO PRINT LINE
*> CDSSRDEF -  CURRENT DEGREE SPECIAL REGISTER
         MVC   ZPRMCL16(Ø9),=C'CDSSRDEF='
         MVC   ZPRMCL16+Ø9(Ø3),SPRMCDEG
         TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
         MVI   Ø(1),C','                 PLUG COMMA HERE
         BAS   R14,ZWRTRTN            DO PRINT LINE
*> DBCHK    -  CHECK DB FOR CONSISTENCY
         MVC   ZPRMCL16(1Ø),=C'DBCHK=NO  '
         TM    SPRMDBCK,X'8Ø'
         BZ    *+1Ø
         MVC   ZPRMCL16+Ø7(Ø4),=C'YES'
         TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
         MVI   Ø(1),C','                 PLUG COMMA HERE
         BAS   R14,ZWRTRTN            DO PRINT LINE
*> DECDIV   - DECIMAL DIVIDE OPTION
         MVC   ZPRMCL16(Ø8),=C'DECDIV3='
         TM    SPRMMISC,B'Ø1ØØØØØØ'
         BNO   NOMISC2
         MVC   ZPRMCL16+Ø8(Ø3),=C'YES'
         B     YESMISC2
NOMISC2  MVC   ZPRMCL16+Ø8(Ø3),=C'NO '
YESMISC2 TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
         MVI   Ø(1),C','                 PLUG COMMA HERE
```

```
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DEFIXTP  - DEFAULT INDEX TYPE
         MVC   ZPRMCL16(Ø9),=C'DEFIXTP=2'
         CLI   SPRMDXTP,2
         BE    *+8
         MVI   ZPRMCL16+Ø8,C'1'
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DEFLTID  - SYSTEM DEFAULT USER ID
         MVC   ZPRMCL16(Ø8),=C'DEFLTID='
         MVC   ZPRMCL16+Ø8(Ø8),SPRMDFID
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DESCSTAT -  DESCRIBE STATIC SQL - YES/NO
         MVC   ZPRMCL16(12),=C'DESCSTAT=YES'
         TM    SPRMMIS2,X'8Ø'
         BO    *+1Ø
         MVC   ZPRMCL16+9(3),=C'NO '
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DLITOUT - CLI TIMEOUT FACTOR
         MVC   ZPRMCL16(Ø8),=C'DLITOUT='  FIELD LITERAL
         SR    R9,R9
         ICM   R9,B'ØØ11',SPRMDLI
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D
         OI    ZPRMCL16+22,X'FØ'
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSMAX    - MAX NUMBER OF DATASETS CONCURRENLY IN USE
         MVC   ZPRMCL16(Ø6),=C'DSMAX='    FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         LH    R9,SPRMDSMX                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø6(7),D           PACK TO ZONE
         OI    ZPRMCL16+12,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø6       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø6(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> EDMPOOL  - EDMPOOL SIZE
         MVC   ZPRMCL16(Ø8),=C'EDMPOOL='  FIELD LITERAL
```

```
         SR    R9,R9                     ZERO REGISTER
         L     R8,SPRMEDPL               GET ZPARM VALUE
         SRDA  R8,32(Ø)                  SHIFT RIGHT 32BITS
         D     R8,=F'1Ø24'               DIVIDE BY 1Ø24
         CVD   R9,D                      CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D         PACK TO ZONE
         OI    ZPRMCL16+22,X'FØ'         FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø8      MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN              DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD  MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
         MVI   Ø(1),C','                 PLUG COMMA HERE
         BAS   R14,ZWRTRTN            DO PRINT LINE
*> EDPROP   -
         MVC   ZPRMCL16(Ø7),=C'EDPROP='
         TM    SPRMMISC,B'ØØØØØ1ØØ'      BIT 6
         BNO   NOMISC6
         MVC   ZPRMCL16+Ø7(Ø3),=C'YES'
         B     YESMISC6
NOMISC6  MVC   ZPRMCL16+Ø7(Ø3),=C'NO '
YESMISC6 TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
         MVI   Ø(1),C','                 PLUG COMMA HERE
         BAS   R14,ZWRTRTN            DO PRINT LINE
*> HOPAUTH  - PKG/RUNNER AUTH TO HOP SITE
         MVC   ZPRMCL16(Ø8),=C'HOPAUTH='
         TM    SPRMMISZ,B'Ø1ØØØØØØ'      BIT 2
         BNO   NOMISZ2
         MVC   ZPRMCL16+Ø8(Ø3),=C'YES'
         B     YESMISZ2
NOMISZ2  MVC   ZPRMCL16+Ø8(Ø3),=C'NO '
YESMISZ2 TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
         MVI   Ø(1),C','                 PLUG COMMA HERE
         BAS   R14,ZWRTRTN            DO PRINT LINE
*> IRLAUT  - IRLM AUTO START
         MVC   ZPRMCL16(11),=C'IRLMAUT=YES' FIELD LITERAL
         TM    SPRMIAUT,X'8Ø'
         BO    *+1Ø
         MVC   ZPRMCL16+Ø8(Ø3),=C'NO '    IRLMAUT=NO
         TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
         MVI   Ø(1),C','                 PLUG COMMA HERE
         BAS   R14,ZWRTRTN            DO PRINT LINE
*> IRLMPRC  - IRLM STARTED PROC
         MVC   ZPRMCL16(Ø8),=C'IRLMPRC='  FIELD LITERAL
         MVC   ZPRMCL16+Ø8(Ø8),SPRMIPRC   GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
         MVI   Ø(1),C','                 PLUG COMMA HERE
         BAS   R14,ZWRTRTN            DO PRINT LINE
*> IRLMSID  - IRLM SUBSYSTEM ID
         MVC   ZPRMCL16(Ø8),=C'IRLMSID='  FIELD LITERAL
         MVC   ZPRMCL16+Ø8(Ø4),SPRMISID   GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE          FIND FIRST BLANK
```

```
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> IRLMRWT  - IRLM TIMEOUT VALUE
         MVC   ZPRMCL16(Ø8),=C'IRLMRWT='  FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         L     R9,SPRMTOUT
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D
         OI    ZPRMCL16+22,X'FØ'
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> IRLMSWT  - IRLM TIME TO WAIT FOR START
         MVC   ZPRMCL16(Ø8),=C'IRLMSWT='  FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         L     R9,SPRMISWT                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE
         OI    ZPRMCL16+22,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> MAXRBLK  - MAX RID BLOCK
         MVC   ZPRMCL16(Ø8),=C'MAXRBLK='  FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         L     R9,SPRMRMAX                GET ZPARM VALUE
         M     R8,=F'16'                  MULTIPLY BY 16
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE
         OI    ZPRMCL16+22,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> MINRBLK  - LEAST # OF RIDLISTS FOR EACH RIDMAP
         MVC   ZPRMCL16(Ø8),=C'MINRBLK='  FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         ICM   R9,B'1111',SPRMRMIN        GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE
         OI    ZPRMCL16+22,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
```

```
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> NUMLKTS  - MAX PAGE LOCKS PER TABLESPACE
         MVC   ZPRMCL16(Ø8),=C'NUMLKTS='  FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         L     R9,SPRMLKTS
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D
         OI    ZPRMCL16+22,X'FØ'
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> NUMLKUS  - MAX PAGE LOCKS PER APPLICATION
         MVC   ZPRMCL16(Ø8),=C'NUMLKUS='  FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         L     R9,SPRMLKUS                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE
         OI    ZPRMCL16+22,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RECALL   - HSM AUTO RECALL
         MVC   ZPRMCL16(Ø7),=C'RECALL='
         CLI   SPRMHRCL,X'8Ø'
         BNE   NORECALL
         MVC   ZPRMCL16+Ø7(Ø3),=C'YES'
         B     YESRECAL
NORECALL MVC   ZPRMCL16+Ø7(Ø3),=C'NO '
YESRECAL TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RECALLD  - HSM AUTO DELAY TIME
         MVC   ZPRMCL16(Ø8),=C'RECALLD='  LITERAL
         SR    R9,R9                      ZERO REGISTER
         LH    R9,SPRMHRCD
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(7),D
         OI    ZPRMCL16+14,X'FØ'
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
```

```
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RETLWAIT -  IRLM WAIT FOR INCOMPATIBLE RETAINED LOCKS
         MVC   ZPRMCL16(12),=C'RETLWAIT=YES'
         TM    SPRMMSC2,B'ØØ1ØØØØØ'
         BO    *+1Ø
         MVC   ZPRMCL16+9(3),=C'NO '
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFCOLID - DDL REGISTRATION TABLE OWNER LID
         MVC   ZPRMCL16(Ø9),=C'RGFCOLID='
         MVC   ZPRMCL16+Ø9(Ø8),SPRMREGC
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFDBNAM - DDL REGISTRATION DATABASE NAME
         MVC   ZPRMCL16(Ø9),=C'RGFDBNAM='
         MVC   ZPRMCL16+Ø9(Ø8),SPRMREGN
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFDEDPL - DDL REGISTRATION DEDICATED APPLS
         MVC   ZPRMCL16(Ø9),=C'RGFDEDPL='
         TM    SPRMREGF,B'Ø1ØØØØØØ'       BIT 6
         BNO   NOREGF2
         MVC   ZPRMCL16+Ø9(Ø3),=C'YES'
         B     YESREGF2
NOREGF2  MVC   ZPRMCL16+Ø9(Ø3),=C'NO '
YESREGF2 TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFDEFLT - DDL REGISTRATION DEFAULT
         MVC   ZPRMCL16(16),=C'RGFDEFLT=ACCEPT '
         TM    SPRMREGF,B'ØØØ1ØØØØ'
         BO    REGF45
         MVC   ZPRMCL16+Ø9(Ø6),=C'APPL  '
         TM    SPRMREGF,B'ØØØØ1ØØØ'
         BO    REGF45
         MVC   ZPRMCL16+Ø9(Ø6),=C'REJECT'
REGF45   TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFESCP  - DDL REGISTRATION ESCAPE CHAR
         MVC   ZPRMCL16(Ø8),=C'RGFESCP='  FIELD LITERAL
         MVI   ZPRMCL16+Ø8,X'4Ø'          GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFFULLQ - DDL REGISTRATION FULLY QUALIFIED NAME
         MVC   ZPRMCL16(Ø9),=C'RGFFULLQ='
```

```
         TM    SPRMREGF,B'ØØ1ØØØØØ'       BIT 3
         BNO   NOREGF3
         MVC   ZPRMCL16+Ø9(Ø3),=C'YES'
         B     YESREGF3
NOREGF3  MVC   ZPRMCL16+Ø9(Ø3),=C'NO '
YESREGF3 TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFINSTL - DDL REGISTRATION INSTALLED
         MVC   ZPRMCL16(Ø9),=C'RGFINSTL='
         TM    SPRMREGF,B'1ØØØØØØØ'       BIT 1
         BNO   NOREGF1
         MVC   ZPRMCL16+Ø9(Ø3),=C'YES'
         B     YESREGF1
NOREGF1  MVC   ZPRMCL16+Ø9(Ø3),=C'NO '
YESREGF1 TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFNMORT - DDL REGISTRATION ORT NAME
         MVC   ZPRMCL16(Ø9),=C'RGFNMORT='
         MVC   ZPRMCL16+Ø9(17),SPRMREGO
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFNMORT - DDL REGISTRATION ART NAME
         MVC   ZPRMCL16(Ø9),=C'RGFNMPRT='
         MVC   ZPRMCL16+Ø9(17),SPRMREGA
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RRULOCK  - RR U-LOCK FOR CURRENT PAGE
         MVC   ZPRMCL16(12),=C'RRULOCK=YES '
         TM    SPRMMISZ,B'ØØØØØ1ØØ'       BIT 5
         BO    *+1Ø
         MVC   ZPRMCL16+Ø8(Ø3),=C'NO '
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SEQCACHE - SEQ MODE/BYPASS 339Ø CACHE
         MVC   ZPRMCL16(Ø9),=C'SEQCACHE='
         TM    SPRMMISZ,B'ØØØ1ØØØØ'       BIT 4
         BNO   NOREGF4
         MVC   ZPRMCL16+Ø9(Ø3),=C'YES'
         B     YESREGF4
NOREGF4  MVC   ZPRMCL16+Ø9(Ø3),=C'NO '
YESREGF4 TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
```

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

Rolf Loeben (Germany) © Xephon 1999

Driving and testing FIELDPROC – part 2

This month we conclude the article giving a PL/I program that drives and tests a DB2 Assembler FIELDPROC exit.

ASSEMBLER EXIT

```
//TSHVRA JOB (),'DBF2UC',CLASS=A,MSGCLASS=X,NOTIFY=TSHVR
//STEP1  EXEC ASMAC,PARM.C='NODECK,OBJECT,RENT'
//C.SYSLIB DD
//         DD
//         DD
//         DD DSN=DSN41Ø.SDSNMACS,DISP=SHR
//         DD DSN=TSHVR.SOURCE.TOOLS,DISP=SHR
//C.SYSLIN DD DSN=TSHVR.TEST.OBJMOD(DBF2UC),DISP=SHR
//C.SYSIN  DD *
*DB2 FIELD TO UPPERCASE FIELD PROCEDURE FIELDPROC DBF2UC
* SEE ALSO DSN41Ø.SDSNSAMP(DSN8FPRC)
*          DSN41Ø.SDSNMACS
*DSN MSG WHEN RC<>Ø:
*  DSNT4Ø8I SQLCODE = -681,
*   ERROR:  COLUMN NAME IN VIOLATION OF INSTALLATION
*    DEFINED FIELD PROCEDURE  RT: 2Ø, RS: ØØ24,
*    MSG: Ø8    1Ø        ØØØ4ØØFEØØØ1ØØØØØØØ8
*
         COPY HVREGS  register equates
$RC_ENC  EQU Ø4
$RC_DEC  EQU Ø8
$RC_DEF  EQU 12
$RC_INV  EQU 16
$RC_MAIN EQU 2Ø
$RSNC_LEN  EQU Ø4
$RSNC_TYPE EQU Ø8
$RSNC_INV  EQU 12
$RSNC_TEST EQU 24
FPPLREG  EQU RØ1                       ; EXTERNAL PARAM LIST
BASEREG  EQU RØ3                       ; EXTERNAL PARAM LIST
* WAREG FPPVLREG : SHOULD STAY TOGETHER. BEGIN WITH EVEN REGISTER
WAREG    EQU RØ4
FPIBREG  EQU RØ5                       ; COMMON INFO BLOCK
CVDREG   EQU RØ6                       ; COLUMN VALUE DESCRIPTOR
FVDREG   EQU RØ7                       ; FIELD VALUE DESCRIPTOR
FPPVLREG EQU RØ8                       ; INTERNAL PARAM LIST
* WAREG FPPVLREG END
WORKØ1 EQU RØ9
WORKØ2 EQU R1Ø
WORKØ3 EQU R11
WORKØ4 EQU R12
```

```
FPBRSNØ   EQU   X'FØ4Ø4Ø4Ø'         SUCCESSFUL
FPBRSN8   EQU   X'F84Ø4Ø4Ø'
*
DBF2UC CSECT
DBF2UC AMODE 31
DBF2UC RMODE ANY
         USING DBF2UC,BASEREG
         STM R14,R12,12(R13)              ; SAVE REGISTERS
         LR  BASEREG,R15
         USING WA,WAREG
         USING FPIB,FPIBREG
         USING FPPVL,FPPVLREG
         LTR FPPLREG,FPPLREG
         BNZ L_FPPLREG_OK
         LA  R15,$RC_MAIN
         B   L_RETURN
L_FPPLREG_OK EQU *
         USING FPPL,FPPLREG
         L  WAREG,FPPWORK
         L  FPIBREG,FPPFPIB
         L  CVDREG,FPPCVD
         L  FVDREG,FPPFVD
         L  FPPVLREG,FPPPVL
         DROP       FPPLREG
         XC WARSNC,WARSNC
         XC FPBTOK,FPBTOK
*        SAVEAREA
         XR    RØØ,RØØ
         ST    RØØ,WASA
         ST    R13,WASA+4
         LA    RØØ,WASA
         ST    RØØ,8(,R13)
         LR    R13,RØØ       SAVEAREA NOW
*        TEST FPPVL
         LR    WORKØ1,R14
         CALL  HVPC2X,(FPBFCODE+1,TKFC,$FW1,                          *
               FPPVLEN,TKVLEN,$FW2,                                   *
               FPPVCNT,TKVCNT,$FW2),                                  *
               VL,MF=(E,CALLLST)
         LR    R14,WORKØ1
         LTR   FPPVLREG,FPPVLREG
         BZ    L_CHKFC
         LH    WORKØ1,FPPVCNT
         LTR   WORKØ1,WORKØ1
         BZ    L_CHKFC
         LA    WORKØ2,FPPVVDS
         LR    WORKØ1,R14
         CALL  HVPC2X,((WORKØ2),TKFW,$FW4),                           *
               VL,MF=(E,CALLLST)
         LR    R14,WORKØ1
         LA    WORKØ2,4(Ø,WORKØ2)     FIRST PVD
*
```

```
         LR    WORKØ1,R14
         CALL  HVPC2X,(FPVDVLEN-FPVD(WORKØ2),TKLN3,$FW2,               *
               FPVDTYPE+1-FPVD(WORKØ2),TKTP3,$FW1),                    *
               VL,MF=(E,CALLLST)
         LR    R14,WORKØ1
*TEST
         CLC   =C'ABEND',FPVDVALE-FPVD(WORKØ2)
         BNE   L_CHKFC
         LA    R15,$RSNC_TEST
         ST    R15,WARSNC
         LA    R15,$RC_MAIN
         B     L_NEXT
*
L_CHKFC  DS    ØH
         XR    R15,R15
         CLI FPBFCODE+1,FPBFDEF
         BE  DEFINE
         CLI FPBFCODE+1,FPBFENC
         BE  ENCODE
         CLI FPBFCODE+1,FPBFDEC
         BE  DECODE
         B   INVFC
ENCODE   DS    ØH
         BAS   R14,S_ENCODE
         B     L_NEXT
DECODE   DS    ØH
         BAS   R14,S_DECODE
         B     L_NEXT
DEFINE   DS    ØH
         BAS   R14,S_DEFINE
         B     L_NEXT
INVFC    DS    ØH
         BAS   R14,S_INVFC
         B     L_NEXT
L_NEXT   EQU *
         CVD   R15,CVDAREA
         UNPK  FPBRTNC,CVDAREA
         OI    FPBRTNC+L'FPBRTNC-1,C'Ø'
         L  WORKØ1,WARSNC
         CVD   WORKØ1,CVDAREA
         UNPK  FPBRSNC,CVDAREA
         OI    FPBRSNC+L'FPBRSNC-1,C'Ø'
         LA  WORKØ1,FPBTOK
         ST  WORKØ1,FPBTOKP
         L     R13,4(,R13)            CALLER'S SAVE AREA ADDRESS
L_RETURN EQU *
         ST R15,16(Ø,R13)
         LM R14,R12,12(R13)
         BR R14
*
S_ENCODE  EQU *
*
```

```
         LR    WORKØ1,R14
         CALL  HVPC2X,(FPVDVLEN-FPVD(CVDREG),TKLN1,$FW2,                *
               FPVDVLEN-FPVD(FVDREG),TKLN2,$FW2,                        *
               FPVDVALE-FPVD(CVDREG),TKLN3,$FW2),                       *
               VL,MF=(E,CALLLST)
         LR    R14,WORKØ1
*
         CLC FPVDVLEN-FPVD(,FVDREG),FPVDVLEN-FPVD(CVDREG)
         BE  L_LENSE_OK
         LA  R15,$RSNC_LEN
         ST  R15,WARSNC
         LA  R15,$RC_ENC
         B   L_ENCODE_EXIT
L_LENSE_OK  EQU *
         CLC FPVDTYPE-FPVD(,FVDREG),FPVDTYPE-FPVD(CVDREG)
         BE  L_TPSE_OK
         LA  R15,$RSNC_TYPE
         ST  R15,WARSNC
         LA  R15,$RC_ENC
         B   L_ENCODE_EXIT
L_TPSE_OK  EQU *
         LA    RØØ,FPVDVALE-FPVD(,FVDREG)
         LH    RØ1,FPVDVLEN-FPVD(,CVDREG)
         LA    WORKØ2,FPVDVALE-FPVD(,CVDREG)
         CLI FPVDTYPE+1-FPVD(CVDREG),FPVDTCHR
         BE  L_MVCLE
*        VARYING LENGTH STRING
         LH    RØ1,FPVDVALE-FPVD(,CVDREG) 1ST HW CONTAINS ACTUAL LENGTH
         LA    RØ1,2(Ø,RØ1)    MAYBE RØ1 CONTAINS X'ØØØØØØØ2'
L_MVCLE  EQU *
         LR    WORKØ3,RØ1
         MVCL  RØØ,WORKØ2
*
L_TRANSLATE EQU *
         CLI FPVDTYPE+1-FPVD(CVDREG),FPVDTCHR
         BNE  L_TR_SETUPV
         LA WORKØ1,FPVDVALE-FPVD(,FVDREG)
         LH WORKØ4,FPVDVLEN-FPVD(,FVDREG)
         B  L_TR_LOOP
L_TR_SETUPV   EQU *
* IF VARYING STRING: @+2 #-2 NOG DOEN
         LA WORKØ1,FPVDVALE+2-FPVD(,FVDREG) SKIP HW LEN FIELD
         LH WORKØ4,FPVDVALE-FPVD(,FVDREG)
L_TR_LOOP EQU *
         C  WORKØ4,$FW256
         BNH L_LT256
         LA  WORKØ3,256
         B   L_DO_TR
L_LT256  EQU *
         LR  WORKØ3,WORKØ4
         LTR WORKØ3,WORKØ3
         BZ  L_TRANSLATE_END
```

```
L_DO_TR  EQU *
         BCTR WORKØ3,Ø    -1
         EX  WORKØ3,L_TR
         LA  WORKØ3,1(Ø,WORKØ3)  +1
         SR   WORKØ4,WORKØ3
         LA  WORKØ1,Ø(WORKØ3,WORKØ1)
         B   L_TR_LOOP
L_TR     TR  Ø(Ø,WORKØ1),$TRUC
L_TRANSLATE_END EQU *
* WORKØ1 POINTS BEYOND BUFFER NOW
*
         XR  R15,R15
L_ENCODE_EXIT EQU *
         BR R14
*
S_DECODE  EQU *
*
         LR    WORKØ1,R14
         CALL  HVPC2X,(FPVDVLEN-FPVD(FVDREG),TKLN1,$FW2,                *
               FPVDVLEN-FPVD(CVDREG),TKLN2,$FW2),                       *
               VL,MF=(E,CALLLST)
         LR    R14,WORKØ1
         CLC FPVDVLEN-FPVD(,FVDREG),FPVDVLEN-FPVD(CVDREG)
         BE  L_LENSD_OK
         LA  R15,$RSNC_LEN
         ST  R15,WARSNC
         LA  R15,$RC_DEC
         B   L_DECODE_EXIT
L_LENSD_OK  EQU *
         CLC FPVDTYPE-FPVD(,FVDREG),FPVDTYPE-FPVD(CVDREG)
         BE  L_TPSD_OK
         LA  R15,$RSNC_TYPE
         ST  R15,WARSNC
         LA  R15,$RC_DEC
         B   L_DECODE_EXIT
L_TPSD_OK  EQU *
*
         LA    RØØ,FPVDVALE-FPVD(,CVDREG)
         LH    RØ1,FPVDVLEN-FPVD(,FVDREG)
         LA    WORKØ2,FPVDVALE-FPVD(,FVDREG)
         CLI FPVDTYPE+1-FPVD(CVDREG),FPVDTCHR
         BE  L_MVCLD
*        VARYING LENGTH STRING
         LH    RØ1,FPVDVALE-FPVD(,FVDREG) 1ST HW CONTAINS ACTUAL LENGTH
         LA    RØ1,2(Ø,RØ1)    MAYBE RØ1 CONTAINS X'ØØØØØØØ2'
L_MVCLD  EQU *
         LR    WORKØ3,RØ1
         MVCL  RØØ,WORKØ2
         XR  R15,R15
L_DECODE_EXIT EQU *
         BR R14
*
```

```
S_DEFINE  EQU *
*
* WHEN NO PARMS ON FIELDPROC:
*  FPPVLEN=Ø X ØØFE = 254
*  FPPVCNT=Ø X ØØØØ
*  FPPVVDS 1ST FW=Ø X ØØØØØØØØ
*   FIRST FPVDTYPE=ØXFF
*   FIRST FPVDVLEN=Ø X ØØØØ
* WHEN 1  PARM  ON FIELDPROC:
*  FPPVLEN=Ø X ØØFE = 254
*  FPPVCNT=Ø X ØØØ1
*  FPPVVDS 1ST FW=Ø X ØØØØØØØ8
*   FIRST FPVDTYPE=Ø X 1Ø
*   FIRST FPVDVLEN=Ø X ØØØ4
*
         LR    WORKØ1,R14
         CALL  HVPC2X,(FPVDTYPE+1-FPVD(CVDREG),TKTP1,$FW1),           *
               VL,MF=(E,CALLLST)
         LR    R14,WORKØ1
         CLI FPVDTYPE+1-FPVD(CVDREG),FPVDTCHR
         BE  L_TYPE_OK
         CLI FPVDTYPE+1-FPVD(CVDREG),FPVDTVCH
         BE  L_TYPE_OK
         LA  R15,$RSNC_TYPE
         ST  R15,WARSNC
         LA  R15,$RC_DEF
         B   L_DEFINE_EXIT
L_TYPE_OK EQU *
*
         MVC FPBWKLN,=Y(WAEND-WA)
*
         MVC FPVDTYPE-FPVD(,FVDREG),FPVDTYPE-FPVD(CVDREG)
         MVC FPVDVLEN-FPVD(,FVDREG),FPVDVLEN-FPVD(CVDREG)
*** TEST XC FPPVLEN,FPPVLEN            ; NO FIELD PROC PARAMETERS
         XR    R15,R15
         B   L_DEFINE_EXIT
L_DEFINE_EXIT EQU *
         BR R14
*
S_INVFC   EQU *
         LA  R15,$RSNC_INV
         ST  R15,WARSNC
         LA  R15,$RC_INV
         BR R14
*
$FW1     DC F'1'
$FW2     DC F'2'
$FW4     DC F'4'
$FW256   DC F'256'
$TRUC    EQU *
         DC    X'ØØØ1Ø2Ø3Ø4Ø5Ø6Ø7Ø8Ø9ØA ØB ØC ØD ØE ØF'
```

```
         DC    X'101112131415161718191A1B1C1D1E1F'
         DC    X'202122232425262728292A2B2C2D2E2F'
         DC    X'303132333435363738393A3B3C3D3E3F'
         DC    X'404162636465666768694A4B4C4D4E4F'
         DC    X'507172737475767778595A5B5C5D5E5F'
         DC    X'606162636465666768696A6B6C6D6E6F'
         DC    X'807172737475767778797A7B7C7D7E7F'
         DC    X'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'
         DC    X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'
         DC    X'A0A1E2E3E4E5E6E7E8E9AAABACADAEAF'
         DC    X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'
         DC    X'C0C1C2C3C4C5C6C7C8C9CAEBECEDEEEF'
         DC    X'D0D1D2D3D4D5D6D7D8D9DAFBFCFDFEDF'
         DC    X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'
         DC    X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'
*
         DSNDFPPB                        FIELD PROCEDURE DEFINITIONS
WA       DSECT                           WORK AREA
WASA     DS    18F                       SAVE AREA
WARSNC   DS    F
FPBTOK   DS    0CL40
TKFC     DS    CL2
TKTP1    DS    CL2
TKTP2    DS    CL2
TKTP3    DS    CL2
TKLN1    DS    CL4
TKLN2    DS    CL4
TKLN3    DS    CL4
TKVLEN   DS    CL4
TKVCNT   DS    CL4
TKFW     DS    CL8
         ORG   FPBTOK+40
CVDAREA  DS    D
CALLLST  CALL  ,(0,0,0,0,0,0,0,0,0),VL,MF=L
WAEND    EQU   *                         END OF WORK AREA
DBF2UC   CSECT
         LTORG
         END
```

ASSEMBLER ROUTINE TO CONVERT TO HEX

```
* HVPC2X : CONVERT TO HEX
*
* HOW TO CALL:
* ASM:
*        CALL  HVPC2X,(GA@,ERRMSG+GWAHEX-INFMSG1,                    *
*              $FW4),VL,MF=(E,CALLLIST3)
* PLI:
*  DCL HVPC2X ENTRY OPTIONS(INTER,ASM,RETCODE);
```

```
*   CALL HVPC2X(EIBFN,FN_HEX,CSTG(EIBFN));
*               IN,OUT,INLEN
*   CALL HVPC2X(AMBLP,PHEX,BINARY(4,31));
*
* GOAL  : NUMBERS OF BYTES CONVERTED TO HEXADECIMAL PRESENTATION
* INPUT : RØ1->PARAMETERLIST
* OUTPUT: HEXADECIMAL PRESENTATION
*
*  CHANGES: HERMAN  Ø9.Ø7.93  MORE THAN 1 ENTRY
*
* REGISTER USAGE   :
* R13   : ADRESS OF SAVE AREA FROM CALLER
* R14   : RETURN ADRESS TO CALLER
* R15   : ENTRY ADRESS OF CALLED HVPC2X / RETURN-CODE
* EQU'S
         COPY HVREGS     REGISTER EQUATES
R_PARM   EQU RØ1
R_BASE   EQU RØ2
R_LEN    EQU RØ3           =  HIGH(LEN/4)
WORKØ1   EQU RØ4
WORKØ2   EQU RØ5
R_NIBBLE_IX EQU RØ6
R_MOD4    EQU RØ7      R_MOD4= MOD(LEN/4)
R_MASK    EQU RØ8
R_INPUT  EQU RØ9
WORKØ3   EQU R1Ø
R_OUTP_IX EQU R11
*
PARMLIST DSECT
INPUT@    DS A        @ TO A
OUTPUT@   DS A        @ TO A
LENGTE@   DS A        @ TO F
NEXT      DS ØA
HVPC2X CSECT
HVPC2X AMODE 31
HVPC2X RMODE ANY
         SAVE   (14,12),,HVPC2X.&SYSTIME..&SYSDATE
         BALR   R_BASE,Ø                    ESTABLISH BASE
         USING  *,R_BASE
         USING PARMLIST,R_PARM
L_NEXT   EQU   *
         ICM   R_OUTP_IX,B'1111',OUTPUT@
         ICM   R_INPUT,B'1111',INPUT@
         ICM   R_LEN,B'1111',LENGTE@ ADRES
         ICM   R_LEN,B'1111',Ø(R_LEN) LENGTE ZELF
*
         IC    R_MASK,MASK4   MOVED OUT OF LOOP TO HERE
         LR    R_MOD4,R_LEN
         SLL   R_MOD4,3Ø(Ø)
         SRL   R_MOD4,3Ø(Ø)   WITHHOLD 2 LAST BITS
```

```
         SRL   R_LEN,2(Ø)      R_LEN=R_LEN/4
         LTR   R_MOD4,R_MOD4
         BZ    L_MAINLOOP MULTIPLE OF 4
         LA    R_LEN,1(R_LEN)  R_LEN = HIGH(LEN/4)
L_MAINLOOP EQU *
*        R_LEN=HIGH(LEN/4)
**       IC    R_MASK,MASK4   MOVED OUT OF LOOP
         LA    R_NIBBLE_IX,8   4 BYTES 8 NIBBLES
         CH    R_LEN,=H'1' LAATSTE RONDE
         BNE   CONT1
         LTR   R_MOD4,R_MOD4
         BZ    CONT1
         LR    R_NIBBLE_IX,R_MOD4
         SLL   R_NIBBLE_IX,1(Ø)  1 NIBBLE = 2 NIBBLES OUTPUT
         LA    WORKØ3,MASK1
         SH    R_MOD4,=H'1'
         IC    R_MASK,Ø(R_MOD4,WORKØ3) POINTER BY GOOD MASKER
CONT1    EQU   *
         XR    WORKØ2,WORKØ2
         EX    R_MASK,ICM
L_LOOP_1_4 EQU *    MAX 4 NIBBLES TO MAX 8 CHARS
         XR    WORKØ1,WORKØ1
*        IC    WORKØ1,C_ØF
         SLDL  WORKØ1,4(Ø)      4 BITS FROM WORKØ2 TO WORKØ1
         CH    WORKØ1,=H'9'
         BH    L_A_F
         LA    WORKØ3,X'FØ'
         B     L_STORE
L_A_F    EQU   *
         SH    WORKØ1,=H'9'
         LA    WORKØ3,X'CØ'
L_STORE  EQU   *
         OR    WORKØ1,WORKØ3
*        WORKØ1 CONTAINS NOW B'1111????'   B'11ØØ????'
         STC   WORKØ1,Ø(R_OUTP_IX)
         LA    R_OUTP_IX,1(Ø,R_OUTP_IX)
         BCT   R_NIBBLE_IX,L_LOOP_1_4
         LA    R_INPUT,4(R_INPUT)
         BCT   R_LEN,L_MAINLOOP
*
         L     WORKØ1,LENGTE@
         LTR   WORKØ1,WORKØ1
         BM    RETURN              LAST PARM
         LA    R_PARM,NEXT
         B     L_NEXT
RETURN   EQU   *
         RETURN (14,12),RC=Ø
*  EXECUTED INSTRUCTION
ICM      ICM   WORKØ2,B'ØØØØ',Ø(R_INPUT)
         DS    ØF
```

```
MASK1    DC    B'00001000'   ORED WITH INSTRUCTION
MASK2    DC    B'00001100'   ORED WITH INSTRUCTION
MASK3    DC    B'00001110'   ORED WITH INSTRUCTION
MASK4    DC    B'00001111'   ORED WITH INSTRUCTION
         LTORG
         END
```

PL/I ROUTINE TO PRINT HEXDUMP

```
* PROCESS OFFSET;
* PROCESS LANGLVL(OS,SPROG);
 /*HVPDMPX DUMP HEX */
 HVPDMPX:PROC(MSGPTR,MSGLEN,SYSPRINT) OPTIONS(REENTRANT)
        RETURNS(BIN FIXED(31));
  %INCLUDE BUILTIN;
   DCL MSGPTR POINTER;
   DCL MSGLEN BIN FIXED(31);
   DCL SYSPRINT FILE ;
   DCL HVPC2X    ENTRY OPTIONS(INTER,ASM,RETCODE);
   DCL MYRC   BIN FIXED(31) INIT(0);
   DCL HEXBUFFER CHAR(32);
   DCL HEXLOPER BIN FIXED(31);
   DCL HEXAANTAL BIN FIXED(31);
   DCL OFFSET_HEX CHAR(8);
   DCL OVERLAYC9999 CHAR(9999) BASED;
    HEXLOPER=1;
    DO WHILE(HEXLOPER<=MSGLEN);
     HEXAANTAL=MSGLEN-HEXLOPER+1;
     IF HEXAANTAL>16 THEN HEXAANTAL=16;
     CALL HVPC2X(SUBSTR(MSGPTR->OVERLAYC9999,HEXLOPER,HEXAANTAL),
                 HEXBUFFER,
                 HEXAANTAL);
     CALL HVPC2X(HEXLOPER-1,
                 OFFSET_HEX,
                 BINARY(4,31));
     PUT FILE(SYSPRINT)
         SKIP EDIT(OFFSET_HEX,SUBSTR(HEXBUFFER,1,HEXAANTAL*2),
                    SUBSTR(MSGPTR->OVERLAYC9999,HEXLOPER,HEXAANTAL))
                    (A(8),X(1),A(32),X(1),A(16));
     HEXLOPER=HEXLOPER+HEXAANTAL;
    END;/*DO WHILE(HEXLOPER<=MSGLEN)*/
  HVPDMPX_END:
  RETURN(MYRC);
 END HVPDMPX;
```

Herman Vierendeels
Systems Programmer (Belgium)

Forté Software has announced its DB2 Adaptor for OS/390, enabling Forté users to deploy RDBMS-based applications with DB2 for OS/390. Working in conjunction with Forté's Application Server for OS/390, DB2 Adaptor for OS/390 enables users to integrate existing DB2 for OS/390 data into distributed computing applications.

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

* * *

IBM has announced its DB2 Forms application development tool. Using any DB2 host or workstation server, applications can be created by developers, governed by administrators, and run by end users on any Windows machine. Advanced database techniques and commands can be used with little or no programming or SQL knowledge.

There are facilities to connect, using the existing network topology, between DB2 Forms applications and multiple DB2 database platforms over publicly-accessible Internet connections, dedicated dial-up lines, TCP/IP intranets, and closed SNA environments.

Finished applications can be distributed freely to in-house or remote users. DRDA support also allows inclusion of multi-vendor databases including IMS, VSAM, Oracle, and Microsoft SQL Server via the DB2 DataJoiner multi-database gateway. The software is a follow-on from the DBEdit development tool, extending its legacy capabilities to Windows desktop machines and remote users.

For further information contact your local IBM representative.

* * *

Lawson Software has announced the integration of its LAWSON INSIGHT Business Management System with DB2 Universal Database, extending Lawson's current support of DB2/400 database for the AS/400 platform to support across all IBM hardware.

LAWSON INSIGHT incorporates a set of integrated, Web-deployable process suites for financials, human resources, procurement, supply chain, and performance management. The Application Designer Suite utilizes Lawson's SEA interface to allow non-technical users to create secure, Web-accessible customized reports, forms, and charts.

For further information contact:
Lawson Software, 1300 Godward Street, Minneapolis, MN 55413, USA.
Tel: (612) 379 2633.
Lawson Software, 1000 Great West Road, Brentford, Middlesex, TW8 9HR, UK.
Tel: (0181) 5600825.
URL: http://www.lawson.com.

* * *