79

DB2

May 1999

In this issue

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.

Recovering tables using image copy dataset

INTRODUCTION

DBAs regularly take an image copy of all tablespaces; however, most do not back-up all tables using DSNTIAUL. Backing-up tables using DSNTIAUL is much more time-consuming than taking an image copy.

Application programmers may sometimes want to see the tables with the previous day's content (or earlier), but not want to overwrite existing data. If the DBA has not taken a DSNTIAUL back-up, he will have to use an image copy dataset using OBID translation, a process that takes significant time and effort. This program automates the process.

PROGRAM CODE

```
IMG2UNL:PROC OPTIONS(MAIN);
EXEC SQL INCLUDE SQLCA;
DCL (ADDR,NULL,LOW,TRANSLATE,SUBSTR,INDEX)  BUILTIN,
    SYSPRINT       FILE  OUTPUT,
    JCLMEM         FILE  OUTPUT,
    SYSIN          FILE  INPUT;
DCL (DBID1,PSID1)                    FIXED BIN(15),
    (DBID2,PSID2)                    FIXED BIN(15),
    (ISOBID1(50),ISOBID2(50))        FIXED BIN(15),
    (ISID1,ISID2)                    FIXED BIN(15),
    (IOBID1(50),IOBID2(50))          FIXED BIN(15),
    (IID1,IID2)                      FIXED BIN(15),
    (OBID1(30),OBID2(30))            FIXED BIN(15),
    (OID1,OID2)                      FIXED BIN(15),
    (TSPQTY,TSSQTY)                  FIXED BIN(31) INIT(0),
    (KEYLEN,MAXKEYLEN,INDEXCNT)      FIXED BIN(31) INIT(0),
    WORKSPACE                        FIXED DEC(15,0) INIT(0),
    (TABCNT,MAXTABUNL)               FIXED BIN(31) INIT(0),
    (SEGSIZE,PARTCNT)                FIXED BIN(15),
    (CRETABSQL,CNTTABSQL)            CHAR(2000) VARYING,
    HLQ                              CHAR(8) VARYING,
    GIVENMAXTS                       CHAR(26) VARYING,
    TBIND                            FIXED BIN(15) INIT(1),
    IXIND                            FIXED BIN(15) INIT(0),
    TSIND                            FIXED BIN(15) INIT(1),
```

```
    (AUTH_ID,TSNAME,DBNAME)          CHAR(8) VARYING,
    (NEWTSNAME,AUTHID,UNITNAME)      CHAR(8) VARYING,
    (III,FIRST_NE)                   FIXED BIN(15),
    UNLIND                           FIXED DEC(2) INIT(1),
    UNLINDCH                         CHAR(8),
    (SSID,VCATNAME)                  CHAR(8) VARYING,
    (TABLE_NAME,TABNAM)              CHAR(18) VARYING;
CALL READ_SYSIN;
CALL CREATE_TABLESPACE;
EXEC SQL SELECT DBID,PSID INTO :DBID2,:PSID2
         FROM SYSIBM.SYSTABLESPACE
         WHERE NAME=:NEWTSNAME AND DBNAME=:DBNAME;
IF SQLCODE < Ø THEN GOTO HATA;
EXEC SQL DECLARE CRS3 CURSOR FOR
         SELECT NAME,CREATOR,OBID FROM SYSIBM.SYSTABLES
         WHERE TSNAME=:TSNAME AND TYPE='T';
IF SQLCODE < Ø THEN GOTO HATA;
EXEC SQL OPEN CRS3;
IF SQLCODE < Ø THEN GOTO HATA;
EXEC SQL FETCH CRS3 INTO :TABNAM,:AUTHID,:OID1;
IF SQLCODE < Ø THEN GOTO HATA;
OBID1(TBIND)=OID1;
DO WHILE (SQLCODE=Ø);
  TABNAM=STRIP_SPACES(TABNAM);
  AUTHID=STRIP_SPACES(AUTHID);
  MAXKEYLEN=Ø;
  CALL CREATE_TABLES;
  EXEC SQL DECLARE KEYS CURSOR FOR
           SELECT SUM(B.LENGTH)
           FROM SYSIBM.SYSKEYS A,SYSIBM.SYSCOLUMNS B,
                SYSIBM.SYSINDEXES C
           WHERE A.COLNAME=B.NAME AND
                 C.NAME=A.IXNAME AND C.CREATOR=A.IXCREATOR AND
                 C.TBNAME=B.TBNAME AND C.TBCREATOR=B.TBCREATOR AND
                 C.TBNAME=:TABNAM AND C.TBCREATOR=:AUTHID
           GROUP BY A.IXNAME,A.IXCREATOR;
  EXEC SQL OPEN KEYS;
  EXEC SQL FETCH KEYS INTO :KEYLEN;
  DO WHILE (SQLCODE=Ø);
    IF KEYLEN > MAXKEYLEN THEN MAXKEYLEN=KEYLEN;
    EXEC SQL FETCH KEYS INTO :KEYLEN;
  END;
  EXEC SQL CLOSE KEYS;
  EXEC SQL DECLARE C1 CURSOR FOR CNTTAB;
  CNTTABSQL='SELECT COUNT(*) FROM '||AUTHID||'.'||TABNAM;
  EXEC SQL PREPARE CNTTAB FROM :CNTTABSQL;
  IF SQLCODE < Ø THEN GOTO HATA;
  EXEC SQL OPEN C1;
  IF SQLCODE < Ø THEN GOTO HATA;
  EXEC SQL FETCH C1 INTO :TABCNT;
```

```
  IF SQLCODE < Ø THEN GOTO HATA;
  IF TABCNT*MAXKEYLEN*INDEXCNT>MAXTABUNL THEN
     MAXTABUNL=TABCNT*MAXKEYLEN*INDEXCNT;
  EXEC SQL CLOSE C1;
  EXEC SQL FETCH CRS3 INTO :TABNAM,:AUTHID,:OID1;
  IF SQLCODE < Ø THEN GOTO HATA;
  TBIND=TBIND+1;
  OBID1(TBIND)=OID1;
END;

TBIND=TBIND-1;
EXEC SQL CLOSE CRS3;
IF SQLCODE < Ø THEN GOTO HATA;
CALL PREPARE_DSN1CP_JCL;
GOTO SON;
CREATE_TABLESPACE:PROC;
  DCL PARTNUM                 FIXED BIN(15),
      STORNAME                CHAR(8) VARYING,
      (CRETSPSQL,CRETSPSQL1)  CHAR(2ØØØ) VARYING;
  EXEC SQL SELECT A.DBID,A.PSID,A.NAME,A.DBNAME,A.SEGSIZE
           INTO :DBID1,:PSID1,:TSNAME,:DBNAME,:SEGSIZE
           FROM SYSIBM.SYSTABLESPACE A,SYSIBM.SYSTABLES B
           WHERE B.TSNAME=A.NAME AND
                 B.DBNAME=A.DBNAME AND
                 B.CREATOR=:AUTH_ID AND
                 B.NAME=:TABLE_NAME;

  IF SQLCODE=1ØØ THEN DO;
    PUT SKIP EDIT('COULD NOT FIND TABLE ',AUTH_ID,'.',TABLE_NAME)
             (A(22),A(8),A(1),A(18));
    STOP;
  END;
  IF SQLCODE < Ø THEN GOTO HATA;

  EXEC SQL SELECT MAX(PARTITION) INTO :PARTCNT
           FROM SYSIBM.SYSTABLEPART
           WHERE TSNAME=:TSNAME AND DBNAME=:DBNAME;
  IF SQLCODE < Ø THEN GOTO HATA;

  IF PARTCNT=Ø THEN DO;
    EXEC SQL SELECT PQTY*4,SQTY*4,VCATNAME,STORNAME
             INTO :TSPQTY,:TSSQTY,:VCATNAME,:STORNAME
             FROM SYSIBM.SYSTABLEPART
             WHERE TSNAME=:TSNAME AND DBNAME=:DBNAME;
    IF SQLCODE < Ø THEN GOTO HATA;

    CRETSPSQL=' IN '||DBNAME||
              ' USING STOGROUP '||STORNAME||
              ' PRIQTY ' || TSPQTY ||
              ' SECQTY ' || TSSQTY ||
```

```
            ' FREEPAGE Ø PCTFREE Ø GBPCACHE  CHANGED'||
            ' BUFFERPOOL BP2 LOCKSIZE ANY CLOSE YES';
  IF SEGSIZE¬=Ø THEN CRETSPSQL=CRETSPSQL||' SEGSIZE '||SEGSIZE;
END;
ELSE DO;
  CRETSPSQL=' IN '||DBNAME||
            ' NUMPARTS '||PARTCNT||' (';

  EXEC SQL DECLARE CRS4 CURSOR FOR
           SELECT PQTY*4,SQTY*4,VCATNAME,STORNAME,PARTITION
           FROM SYSIBM.SYSTABLEPART
           WHERE TSNAME=:TSNAME AND DBNAME=:DBNAME
           ORDER BY PARTITION;
  IF SQLCODE < Ø THEN GOTO HATA;
  EXEC SQL OPEN CRS4;

  EXEC SQL FETCH CRS4
           INTO :TSPQTY,:TSSQTY,:VCATNAME,:STORNAME,:PARTNUM;
  IF SQLCODE < Ø THEN GOTO HATA;

  DO WHILE(SQLCODE=Ø);
    CRETSPSQL= CRETSPSQL||' PART '||PARTNUM||
               ' USING STOGROUP '||STORNAME||
               ' PRIQTY ' || TSPQTY ||
               ' SECQTY ' || TSSQTY ||
               ' FREEPAGE Ø PCTFREE Ø GBPCACHE  CHANGED ';

    IF PARTNUM=PARTCNT THEN CRETSPSQL= CRETSPSQL||')';
                       ELSE CRETSPSQL= CRETSPSQL||',';

    EXEC SQL FETCH CRS4
             INTO :TSPQTY,:TSSQTY,:VCATNAME,:STORNAME,:PARTNUM;
    IF SQLCODE < Ø THEN GOTO HATA;
  END;

  EXEC SQL CLOSE CRS4;
  CRETSPSQL= CRETSPSQL||' BUFFERPOOL BP2 LOCKSIZE ANY CLOSE YES';
END;

LB2:DO III=1 TO 2Ø;
  UNLINDCH=UNLIND;
  NEWTSNAME='UNLOAD'||STRIP_SPACES(UNLINDCH);
  CRETSPSQL1='CREATE TABLESPACE '||NEWTSNAME||CRETSPSQL;
  EXEC SQL EXECUTE IMMEDIATE :CRETSPSQL1;

  IF SQLCODE=-6Ø1 THEN UNLIND=UNLIND+1;
  ELSE IF SQLCODE=Ø THEN LEAVE LB2;
  ELSE IF SQLCODE<Ø THEN DO;
    PUT SKIP LIST(CRETSPSQL1);
    GOTO HATA;
```

```
    END;
  END;
  IF III=21 THEN DO;
    PUT SKIP EDIT('PLEASE DROP ALL UNNECESSARY UNLOADXX TABLESPACES')
                (A(8Ø));
    PUT SKIP EDIT('AND RERUN THIS JCL...') (A(8Ø));
    STOP;
  END;

  PUT SKIP EDIT('TABLESPACE ',NEWTSNAME,' CREATED...')
            (A(11),A(8),A(2Ø));
  EXEC SQL COMMIT;
END CREATE_TABLESPACE;

CREATE_TABLES:PROC;
  CRETABSQL='CREATE TABLE UNL.'||TABNAM||' LIKE '||
            AUTHID||'.'||TABNAM||' IN '||DBNAME||'.'||NEWTSNAME;
  EXEC SQL EXECUTE IMMEDIATE :CRETABSQL;
  IF SQLCODE=-6Ø1 THEN DO;
    PUT SKIP EDIT('UNL.',TABNAM,' ALREADY EXISTS.')(A(4),A(18),A(2Ø));
    PUT SKIP EDIT('PLEASE DROP THE TABLE AND RERUN THIS JCL...')
            (A(8Ø));
  END;
  IF SQLCODE < Ø THEN DO;
    PUT SKIP LIST(CRETABSQL);
    GOTO HATA;
  END;

  PUT SKIP EDIT('TABLE UNL.',TABNAM,' CREATED.')(A(1Ø),A(18),A(9));
  EXEC SQL SELECT OBID INTO :OID2
           FROM SYSIBM.SYSTABLES
           WHERE NAME=:TABNAM AND CREATOR='UNL' AND TYPE='T';
  OBID2(TBIND)=OID2;
  IF SQLCODE < Ø THEN GOTO HATA;
  CALL CREATE_INDEXES;
END CREATE_TABLES;
CREATE_INDEXES:PROC;
  DCL FIRST_COME             FIXED BIN(15),
      (PARTCNT1,PARTNUM)     FIXED BIN(15),
      IXNAME                 CHAR(18) VARYING,
      (IXCREATOR,STORNAME)   CHAR(8) VARYING,
      COLNAME                CHAR(18),
      (ORDERING,UNIQUERULE)  CHAR(1),
      CREINDSQL              CHAR(4ØØØ) VARYING,
      LIMITKEY               CHAR(512) VARYING INIT(' '),
      PQTY                   FIXED BIN(31),
      (COLSEQ,SQTY)          FIXED BIN(15);
  INDEXCNT=Ø;
  EXEC SQL DECLARE CRS1 CURSOR FOR
           SELECT NAME,CREATOR,UNIQUERULE,ISOBID,OBID
```

```
         FROM SYSIBM.SYSINDEXES
         WHERE TBNAME=:TABNAM AND TBCREATOR=:AUTHID;
IF SQLCODE < Ø THEN GOTO HATA;
EXEC SQL OPEN CRS1;
IF SQLCODE < Ø THEN GOTO HATA;
EXEC SQL FETCH CRS1 INTO
         :IXNAME,:IXCREATOR,:UNIQUERULE,:ISID1,:IID1;
IF SQLCODE < Ø THEN GOTO HATA;
IXIND=IXIND+1;
ISOBID1(IXIND)=ISID1;
IOBID1(IXIND)=IID1;
DO WHILE(SQLCODE=Ø);
  INDEXCNT=INDEXCNT+1;
  IXCREATOR=STRIP_SPACES(IXCREATOR);
  IXNAME=STRIP_SPACES(IXNAME);
  IF UNIQUERULE='D' THEN
    CREINDSQL='CREATE TYPE 2 INDEX UNL.';
  ELSE
    CREINDSQL='CREATE TYPE 2 UNIQUE INDEX UNL.';
  CREINDSQL=CREINDSQL||IXNAME||' ON UNL.'||
            TABNAM||' (';
  FIRST_COME=1;
  EXEC SQL DECLARE CRS2 CURSOR FOR
           SELECT COLNAME,ORDERING,COLSEQ
           FROM SYSIBM.SYSKEYS
           WHERE IXNAME=:IXNAME AND IXCREATOR=:IXCREATOR
           ORDER BY COLSEQ;
  IF SQLCODE < Ø THEN GOTO HATA;
  EXEC SQL OPEN CRS2;
  IF SQLCODE < Ø THEN GOTO HATA;
  EXEC SQL FETCH CRS2 INTO :COLNAME,:ORDERING,:COLSEQ;
  IF SQLCODE < Ø THEN GOTO HATA;
  DO WHILE(SQLCODE=Ø);
    IF FIRST_COME=Ø THEN DO;
      CREINDSQL=CREINDSQL||',';
    END;
    FIRST_COME=Ø;
    CREINDSQL=CREINDSQL||COLNAME;
    IF ORDERING='A' THEN CREINDSQL=CREINDSQL||'ASC ';
    EXEC SQL FETCH CRS2 INTO :COLNAME,:ORDERING,:COLSEQ;
  END;
  EXEC SQL SELECT MAX(PARTITION) INTO :PARTCNT1
           FROM SYSIBM.SYSINDEXPART
           WHERE IXNAME=:IXNAME AND IXCREATOR=:IXCREATOR;
  IF SQLCODE < Ø THEN GOTO HATA;
  IF PARTCNT1=Ø THEN DO;
    EXEC SQL SELECT PQTY*4,SQTY*4,STORNAME
             INTO :PQTY,:SQTY,:STORNAME
             FROM SYSIBM.SYSINDEXPART
             WHERE IXNAME=:IXNAME AND IXCREATOR=:IXCREATOR;
```

```
  IF SQLCODE < Ø THEN GOTO HATA;
  CREINDSQL=CREINDSQL||') USING STOGROUP '||STORNAME||
            ' PRIQTY '||PQTY||' SECQTY '|| SQTY||
            ' FREEPAGE Ø PCTFREE Ø GBPCACHE CHANGED'||
            ' BUFFERPOOL BP3 CLOSE YES';
END;
ELSE DO;
  CREINDSQL=CREINDSQL||') CLUSTER (';
  EXEC SQL DECLARE CRS5 CURSOR FOR
           SELECT PQTY*4,SQTY*4,STORNAME,PARTITION
           FROM SYSIBM.SYSINDEXPART
           WHERE IXNAME=:IXNAME AND IXCREATOR=:IXCREATOR;
  IF SQLCODE < Ø THEN GOTO HATA;
  EXEC SQL OPEN CRS5;
  EXEC SQL FETCH CRS5
           INTO :PQTY,:SQTY,:STORNAME,:PARTNUM;
  IF SQLCODE < Ø THEN GOTO HATA;
  DO WHILE(SQLCODE=Ø);
    EXEC SQL SELECT LIMITKEY INTO :LIMITKEY
             FROM SYSIBM.SYSTABLEPART
             WHERE PARTITION=:PARTNUM AND
                   TSNAME=:TSNAME AND
                   DBNAME=:DBNAME AND
                   IXNAME=:IXNAME AND
                   IXCREATOR=:IXCREATOR;
    IF SQLCODE < Ø THEN GOTO HATA;
    LIMITKEY=STRIP_SPACES(LIMITKEY);
    CREINDSQL=CREINDSQL||' PART '||PARTNUM||
              ' VALUES('||LIMITKEY||')'||
              ' USING STOGROUP '||STORNAME||
              ' PRIQTY '||PQTY||' SECQTY '|| SQTY||
              ' FREEPAGE Ø PCTFREE Ø GBPCACHE CHANGED';
    IF PARTNUM=PARTCNT1 THEN
        CREINDSQL=CREINDSQL||') BUFFERPOOL BP3 CLOSE YES';
    ELSE CREINDSQL=CREINDSQL||',';
    EXEC SQL FETCH CRS5
             INTO :PQTY,:SQTY,:STORNAME,:PARTNUM;
    IF SQLCODE < Ø THEN GOTO HATA;
  END;
  EXEC SQL CLOSE CRS5;
  IF SQLCODE < Ø THEN GOTO HATA;
END;
EXEC SQL EXECUTE IMMEDIATE :CREINDSQL;
IF SQLCODE < Ø THEN DO;
  PUT SKIP LIST(CREINDSQL);
  GOTO HATA;
END;
PUT SKIP EDIT('INDEX UNL.',IXNAME,' CREATED.')
         (A(1Ø),A(18),A(9));
EXEC SQL CLOSE CRS2;
```

```
    IF SQLCODE < Ø THEN GOTO HATA;
    EXEC SQL SELECT ISOBID,OBID INTO :ISID2,:IID2
             FROM SYSIBM.SYSINDEXES
             WHERE NAME=:IXNAME AND CREATOR='UNL';
    IF SQLCODE < Ø THEN GOTO HATA;
    ISOBID2(IXIND)=ISID2;
    IOBID2(IXIND)=IID2;
    EXEC SQL FETCH CRS1 INTO
             :IXNAME,:IXCREATOR,:UNIQUERULE,:ISID1,:IID1;
    IF SQLCODE < Ø THEN GOTO HATA;
    IXIND=IXIND+1;
    ISOBID1(IXIND)=ISID1;
    IOBID1(IXIND)=IID1;
  END;
  EXEC SQL CLOSE CRS1;
  IF SQLCODE < Ø THEN GOTO HATA;
  IXIND=IXIND-1;
END CREATE_INDEXES;
PREPARE_DSN1CP_JCL:PROC;
  DCL JCL_LINE(1ØØØ)      CHAR(8Ø) VARYING,
      DSNAME              CHAR(44) VARYING,
      MAXTS               CHAR(26) INIT(''),
      WORKSPACEC          CHAR(15) VARYING,
      PARTCNTC            CHAR(15) VARYING,
      (III,LNCN)          FIXED BIN(15);

  VCATNAME=STRIP_SPACES(VCATNAME);
  PARTCNTC=PARTCNT;
  PARTCNTC=STRIP_SPACES(PARTCNTC);
  EXEC SQL SELECT MAX(TIMESTAMP) INTO :MAXTS
           FROM SYSIBM.SYSCOPY
           WHERE TSNAME=:TSNAME AND
                 DBNAME=:DBNAME AND
                 ICTYPE='F' AND
                 TIMESTAMP<:GIVENMAXTS;
  PUT SKIP EDIT('IMAGE COPY TAKEN AT ',MAXTS,' WILL BE USED.')
              (A(21),A(24),A(14));
  EXEC SQL SELECT DSNAME INTO :DSNAME
           FROM SYSIBM.SYSCOPY
           WHERE TIMESTAMP=:MAXTS AND
                 TSNAME=:TSNAME AND
                 DBNAME=:DBNAME AND
                 ICTYPE='F';
  DSNAME=STRIP_SPACES(DSNAME);
  JCL_LINE(1)='//DSN1COPY JOB (ACCT(tm)),''DSN1COPY'',MSGCLASS=X,';
  JCL_LINE(2)='// MSGLEVEL=(1,1),CLASS=A,NOTIFY=&SYSUID';
  JCL_LINE(3)='//STOPTS  EXEC PGM=IKJEFTØ1,REGION=1Ø24K';
  JCL_LINE(4)='//SYSPRINT DD SYSOUT=*';
  JCL_LINE(5)='//STEPLIB   DD DISP=SHR,DSN='||HLQ||'.SDSNLOAD';
  JCL_LINE(6)='//          DD DISP=SHR,DSN='||HLQ||'.SDSNLOAD';
```

```
JCL_LINE(7)='//SYSTSPRT  DD SYSOUT=*';
JCL_LINE(8)='//SYSPRINT  DD SYSOUT=*';
JCL_LINE(9)='//SYSUDUMP  DD DUMMY';
JCL_LINE(10)='//SYSTSIN   DD *';
JCL_LINE(11)='  DSN SYSTEM('||SSID||')';
JCL_LINE(12)='  -STOP DB('||DBNAME||') SPACENAM('||
            NEWTSNAME||')';
JCL_LINE(13)='  END';
JCL_LINE(14)='/*';
JCL_LINE(15)='//DSN1COPY JOB (ACCT(tm)),''DSN1COPY'',MSGCLASS=X,';
JCL_LINE(16)='// MSGLEVEL=(1,1),CLASS=A,NOTIFY=&SYSUID';
JCL_LINE(17)='//DSN1CP  EXEC PGM=DSN1COPY,REGION=1024K,';
IF PARTCNT=0 THEN
  JCL_LINE(18)='//             PARM=''OBIDXLAT,FULLCOPY,'||
            'RESET''';
ELSE JCL_LINE(18)='//             PARM=''OBIDXLAT,FULLCOPY,'||
            'NUMPARTS('||PARTCNTC||')'||',RESET''';
JCL_LINE(19)='//STEPLIB DD DSN='||HLQ||'.SDSNLOAD,DISP=SHR';
JCL_LINE(20)='//SYSPRINT DD SYSOUT=*';
JCL_LINE(21)='//SYSUT1  DD DSN='||DSNAME||',';
JCL_LINE(22)='//            DISP=OLD,UNIT='||UNITNAME;
JCL_LINE(23)='//SYSUT2  DD DSN='||VCATNAME||'.DSNDBC.'||DBNAME||
            '.'||NEWTSNAME||'.I0001.A001,';
JCL_LINE(24)='//            DISP=OLD';
JCL_LINE(25)='//*';
JCL_LINE(26)='//SYSXLAT DD *';
JCL_LINE(27)='  '||DBID1||','||DBID2;
JCL_LINE(28)='  '||PSID1||','||PSID2;
DO III=1 TO IXIND;
  JCL_LINE(28+III)='  '||ISOBID1(III)||','||ISOBID2(III);
END;
DO III=1 TO IXIND;
  JCL_LINE(28+IXIND+III)='  '||IOBID1(III)||','||IOBID2(III);
END;
DO III=1 TO TBIND;
  JCL_LINE(28+IXIND*2+III)='  '||OBID1(III)||','||OBID2(III);
END;
LNCN=28+IXIND*2+TBIND;
JCL_LINE(LNCN+1)='//DSN1COPY JOB ACCT(tm)),''DSN1COPY'',MSGCLASS=X,';
JCL_LINE(LNCN+2)='// MSGLEVEL=(1,1),CLASS=A,NOTIFY=&SYSUID';
JCL_LINE(LNCN+3)='//STARTTS  EXEC PGM=IKJEFT01,REGION=1024K';
JCL_LINE(LNCN+4)='//SYSPRINT DD SYSOUT=*';
JCL_LINE(LNCN+5)='//STEPLIB   DD DISP=SHR,DSN='||HLQ||'.SDSNLOAD';
JCL_LINE(LNCN+6)='//          DD DISP=SHR,DSN='||HLQ||'.SDSNLOAD';
JCL_LINE(LNCN+7)='//SYSTSPRT  DD SYSOUT=*';
JCL_LINE(LNCN+8)='//SYSPRINT  DD SYSOUT=*';
JCL_LINE(LNCN+9)='//SYSUDUMP  DD DUMMY';
JCL_LINE(LNCN+10)='//SYSTSIN   DD *';
JCL_LINE(LNCN+11)='  DSN SYSTEM('||SSID||')';
JCL_LINE(LNCN+12)='  -START DB('||DBNAME||') SPACENAM('||
```

```
                    NEWTSNAME||')';
  JCL_LINE(LNCN+13)='  END';
  JCL_LINE(LNCN+14)='/*';
  IF MAXTABUNL < 1Ø24 THEN MAXTABUNL=1Ø24;
  WORKSPACEC=MAXTABUNL/1Ø24;
  WORKSPACEC=STRIP_SPACES(WORKSPACEC);
  IF MAXTABUNL > Ø THEN DO;
  JCL_LINE(LNCN+15)='//RECINDX  EXEC PGM=DSNUTILB,REGION=1Ø24K,';
  JCL_LINE(LNCN+16)='//            PARM='''||SSID||',DSNTEX''';
  JCL_LINE(LNCN+17)='//STEPLIB  DD DSN='||HLQ||'.SDSNLOAD,DISP=SHR';
  JCL_LINE(LNCN+18)='//*';
  JCL_LINE(LNCN+19)='//SORTWKØ1 DD DSN=SYSPDBA.PSØ.RECIND.WORK1.TEMP,';
  JCL_LINE(LNCN+2Ø)='//            SPACE=(1Ø24,('||WORKSPACEC||
                    ','||WORKSPACEC||'),,,ROUND)';
  JCL_LINE(LNCN+21)='//SORTWKØ2 DD DSN=SYSPDBA.PSØ.RECIND.WORK2.TEMP,';
  JCL_LINE(LNCN+22)='//            SPACE=(1Ø24,('||WORKSPACEC||
                    ','||WORKSPACEC||'),,,ROUND)';
  JCL_LINE(LNCN+23)='//SYSUT1   DD DSN=SYSPDBA.PSØ.RECIND.SUT1.TEMP,';
  JCL_LINE(LNCN+24)='//            SPACE=(1Ø24,('||WORKSPACEC||
                    ','||WORKSPACEC||'),,,ROUND)';
  JCL_LINE(LNCN+25)='//SYSPRINT DD SYSOUT=*';
  JCL_LINE(LNCN+26)='//UTPRINT  DD SYSOUT=*';
  JCL_LINE(LNCN+27)='//SYSIN    DD *';
  JCL_LINE(LNCN+28)='   RECOVER INDEX ALL TABLESPACE '||DBNAME||'.'||
                    NEWTSNAME;
  JCL_LINE(LNCN+29)='/*';
  DO III=1 TO LNCN+29;
    PUT FILE(JCLMEM) SKIP EDIT(JCL_LINE(III))(A(8Ø));
  END;
  END;
  ELSE DO;
    DO III=1 TO LNCN+14;
      PUT FILE(JCLMEM) SKIP EDIT(JCL_LINE(III))(A(8Ø));
    END;
  END;
END PREPARE_DSN1CP_JCL;
STRIP_SPACES:PROC(TEXT) RETURNS(CHAR(5ØØ) VARYING);
  DCL TEXT CHAR(1ØØØ),
      III,FIRST_NE  FIXED BIN(15);
  FIRST_NE=Ø;
  LB1:DO III=1 TO 1ØØØ;
    IF FIRST_NE¬=Ø & SUBSTR(TEXT,III,1)=' ' THEN DO;
      RETURN(SUBSTR(TEXT,FIRST_NE,III-FIRST_NE));
      LEAVE LB1;
    END;
    IF FIRST_NE=Ø & SUBSTR(TEXT,III,1)¬=' ' THEN DO;
      FIRST_NE=III;
    END;
  END;
END STRIP_SPACES;
```

```
READ_SYSIN:PROC;
  DCL (INPUT_TABNAM,INPUT_AUTHID) CHAR(70),
      (INPUT_SSID,INPUT_DATE)     CHAR(70),
      (INPUT_HLQ,INPUT_UNITNAME)  CHAR(70);
  GET SKIP EDIT(INPUT_AUTHID)(A(70));
  GET SKIP EDIT(INPUT_TABNAM)(A(70));
  GET SKIP EDIT(INPUT_SSID)(A(70));
  GET SKIP EDIT(INPUT_HLQ)(A(70));
  GET SKIP EDIT(INPUT_DATE)(A(70));
  GET SKIP EDIT(INPUT_UNITNAME)(A(70));
  AUTH_ID=STRIP_SPACES(INPUT_AUTHID);
  TABLE_NAME=STRIP_SPACES(INPUT_TABNAM);
  SSID=STRIP_SPACES(INPUT_SSID);
  HLQ=STRIP_SPACES(INPUT_HLQ);
  GIVENMAXTS=STRIP_SPACES(INPUT_DATE);
  UNITNAME=STRIP_SPACES(INPUT_UNITNAME);
END READ_SYSIN;

HATA:
 PUT SKIP LIST(SQLCA);
 EXEC SQL ROLLBACK;
 STOP;
SON:
END IMG2UNL;
```

PROGRAM EXPLANATION

The table to be recovered is entered in the first two lines of SYSIN. This program creates a new tablespace with the same type as the tablespace to which the table belongs. The name of the new tablespace always starts with UNLOAD (UNLOAD1, UNLOAD2, etc). A new table is created, identical to the given table, with the same name but with a different authorization-id (UNL). All other tables in the original tablespace are also created in the new tablespace with authorization-id UNL. All indexes for all tables are created on the newly created tables. For example UNL.CST_PSTG_ENT table is created for table THST.CST_PSTG_ENT. All other tables in the tablespace of THST.CST_PSTG_ENT are created with the authorization-id UNL. All indexes are also created with authorization-id UNL.

DBID, PSID, and ISOBIDs for all indexes, and OBIDs for all tables, are written for current and newly created objects for OBID translation.

If the new tables are not needed any more, you should drop the UNLOAD*xx* tablespaces since this occupies unnecessary space on disk.

The JCLMEM DD of the JCL points to a sequential dataset that will be used for DSN1COPY JCL. Three jobs will be submitted within this JCL. The first stops the new tablespace. The second runs DSN1COPY job with OBID translation, and the third starts the tablespace and reorganizes all indexes of the new tablespace.

The DB2 subsystem-id, the HLQ of the DB2 datasets, and the tape unit name must also be entered from SYSIN DD.

We also give a timestamp in SYSIN. This timestamp is used for defining which image copy will be used. The newest image copy that is older than the given timestamp will be used for the DSN1COPY job.

SAMPLE JCL TO RUN PROGRAM

```
//IMG2UNL1 JOB (ACCTO),,MSGCLASS=X,MSGLEVEL=(1,1),CLASS=A
//SQL EXEC PGM=IKJEFT01,REGION=512K
//STEPLIB   DD DISP=SHR,DSN=TDSN.SDSNLOAD
//JCLMEM    DD DSN=SYSPDBA.PS0.DSN1CP,DISP=OLD
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD DUMMY
//SYSIN     DD *
   THST
   CST_PSTG_ENT
   DBT0
   TDSN
   1998-12-30-11.00.00.000000
   CART
/*
//SYSTSIN   DD *
  DSN SYSTEM(DBT0)
  RUN PROGRAM(IMG2UNL) PLAN(IMG2UNL) LIB('SYSPDBA.LOADLIB')
  END
/*
//RUNJCL  EXEC PGM=IKJEFT01,REGION=512K,COND=(4,LT)
//SYSPRINT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
   SUBMIT 'SYSPDBA.PS0.DSN1CP'
/*
```

Abdullah Ongul
DB2 DBA
Pamukbank (Turkey)

DB2 catalog statistics update REXX EXEC – part 2

This month we conclude the article providing a REXX EXEC that can be used to update DB2 catalog statistics.

```
 /*-------------------------------------------------------------------*/
 /* If we are connected to DB2 ( DB2CON = 1) then updates  */
 /* have been performed, therefore a 'COMMIT' is required. */
 /*-------------------------------------------------------------------*/
  if DB2CON = 1 then do
     address db2 "COMMIT"
     if SQLCODE <> 0 then do
        MESS = 'Error on DB2 Commit sqlcode ' SQLCODE
        return
     end
     MESS = 'Table/Index statistics updated'
  end
  else
     MESS = 'No updates required for Table/Index statistics'
 /*-------------------------------------------------------------------- */
 /* Set DUPD flag to show that new number of rows is active */
 /*-------------------------------------------------------------------- */
  DUPD = 1
return

 /*-------------------------*/
 /* Process Index Columns  */
 /*-------------------------*/
E_Columns:
  if DB2CON = 0 then do
     address db2 "SIGNON" DB2S
     if RC > 0 then do
        MESS = "Error - Unable to connect to " DB2S
        RETURN16
     end
  end
  DB2CON = 1
 /*---------------------------------------------- */
 /* Define ISPF Table T£CLM to hold column data */
 /*---------------------------------------------- */
  address ispexec "TBCREATE T£CLM",
          "NAMES(CNAME CFLD NCDATAD NCDATAE CCDATAE CCDATAD CTYPE",
          " CLEN)",
          "NOWRITE REPLACE"
  CNAME  = ""
  CCFLD  = ""
  CTYPE  = ""
```

```
 NCDATAD= ""
 NCDATAE= ""
 CCDATAE= ""
/*-------------------------------------------------------------------- */
/* Set up SQL statement to retrieve column data from syscolumns, */
/* using correlated subquery on sysindexes and syskeys to ensure */
/* returned values are for index columns.                        */
/* Declare and open cursor and fetch required data.              */
/*-------------------------------------------------------------------- */
 SLCT =      "SELECT A.TBCREATOR,A.TBNAME,A.NAME,",
             "A.COLCARD,HEX(A.LOW2KEY),HEX(A.HIGH2KEY)",
             ",A.COLTYPE,A.LENGTH FROM SYSIBM.SYSCOLUMNS A"
 WHRCLS =    "WHERE A.TBCREATOR='"CRTR"' AND A.TBNAME='"TBNAM"'"
 SUBSLCT =   "AND EXISTS (SELECT * FROM SYSIBM.SYSINDEXES B",
             ",SYSIBM.SYSKEYS C WHERE B.TBCREATOR = A.TBCREATOR",
             "AND B.TBNAME=A.TBNAME AND B.CREATOR=C.IXCREATOR",
             "AND B.NAME=C.IXNAME AND A.NAME=C.COLNAME)"
 ORDBY  =    " ORDER BY A.TBCREATOR,A.TBNAME,A.NAME"
 address db2 "DECLARE RXCSR4 CURSOR FOR",
             SLCT,
             WHRCLS,
             SUBSLCT,
             ORDBY
 if RC ¬= Ø then
    MESS = 'Error declaring cursor - RC ' RC
 else do
    address db2 "OPEN RXCSR4"
    do A = 1 by 1 until SQLCODE ¬= Ø
       address db2 "FETCH RXCSR4"
       if SQLCODE ¬= Ø then iterate A
/*-------------------------------------- */
/* Populate Table T£CLM with column data */
/*-------------------------------------- */
       CNAME  = RXCSR4.3
       CFLD   = 'Colcard'
       CTYPE  = RXCSR4.7
       CLEN = RXCSR4.8
       CCARD  = RXCSR4.4
       CCDATAD= RXCSR4.4
       NCDATAD= RXCSR4.4
       CCDATAE= RXCSR4.4
       NCDATAE= RXCSR4.4
       address ispexec "TBADD T£CLM"
       CNAME  = CTYPE
       if CTYPE = 'CHAR' | CTYPE = 'DECIMAL' then
          CNAME  = CTYPE || ' ' || CLEN
       CFLD   = 'Lo2key'
       CCDATAE= RXCSR4.5
       call EB_DECODE
       NCDATAE= CCDATAE
```

```
      NCDATAD= CCDATAD
      address ispexec "TBADD T£CLM"
      CNAME  = ' '
      CFLD   = 'Hi2key'
      CCDATAE= RXCSR4.6
      call EB_DECODE
      NCDATAE= CCDATAE
      NCDATAD= CCDATAD
      address ispexec "TBADD T£CLM"
   end
   address db2 "CLOSE RXCSR4"
 end
 address db2 "SIGNOFF"
 DB2CON = Ø
 ZCMD = ''
 address ispexec "TBTOP T£CLM"
/*———————————————————————————— */
/* Process column statistics panel */
/*———————————————————————————— */
 do forever
   address ispexec "TBDISPL T£CLM PANEL(CSCLM) AUTOSEL(NO)"
   if rc > 4 then leave
   MESS = ' '
   K = ZTDSELS + Ø
   do until K < 1
      if K > Ø then do I = 1 by 1 to K
         if NCDATAD = '' | NCDATAD = ' ' then do
            NCDATAE = CCDATAE
            NCDATAD = CCDATAD
         end
         else
/*———————————————————————————————*/
/* Process data depending on CFLD */
/*———————————————————————————————*/
            if CFLD = 'Colcard' then
               NCDATAE = NCDATAD
            else
               call EC_ENCODE
         address ispexec "TBPUT T£CLM"
         address ispexec "TBDISPL T£CLM"
         MESS = ' '
      end /* I LOOP */
      K = ZTDSELS + Ø
   end /* K LOOP */
   MESS = ' '
   if rc > 4 then leave
   if ZCMD = exit then exit
   if ZCMD = end  then leave
   if ZCMD = 'UPD'
   then do
```

```
        ZCMD = ''
        call EA_UPDATE
        leave
     end
  end
  return

 /*———————————————————*/
 /* Update column data */
 /*———————————————————*/
EA_UPDATE:
  address ispexec "TBTOP T£CLM"
  do forever
     CFLAG = Ø
     address ispexec "TBSKIP T£CLM"
     if rc > Ø then leave
     if CNAME  = ' ' then error
     UCNAME = CNAME
 /*———————————————————————————————————————— */
 /* Ensure first table entry is for colcard.*/
 /* Three table entries for each update.    */
 /*———————————————————————————————————————— */
     if CFLD <> 'Colcard' then do
        MESS = 'Error first table entry not colcard'
        return
     end
     if CCDATAE <> NCDATAE then CFLAG = 1
     NCCARD = NCDATAD
     address ispexec "TBSKIP T£CLM"
     if CCDATAD <> NCDATAD then CFLAG = 1
     NL2KEY = x2c(NCDATAE)
     address ispexec "TBSKIP T£CLM"
     if CCDATAD <> NCDATAD then  CFLAG = 1
     NH2KEY = x2c(NCDATAE)
 /*————————————————————————————————————————*/
 /* CFLAG set if any updates are required */
 /*————————————————————————————————————————*/
     if CFLAG = 1 then do
        CFLAG = Ø
        if DB2CON = Ø then do
           address db2 "SIGNON" DB2S
           if RC > Ø then do
              MESS = "Error - Unable to connect to " DB2S
              RETURN16
           end
           DB2CON = 1
        end
        UPDT = "UPDATE SYSIBM.SYSCOLUMNS ",
            "SET COLCARD="NCCARD",LOW2KEY='"NL2KEY"',",
            "HIGH2KEY='"NH2KEY"'"
```

```
        WHRCLS = "WHERE TBCREATOR='"CRTR"' AND TBNAME='"TBNAM"'",
                 "AND NAME='"UCNAME"'"
        address db2 UPDT,
                    WHRCLS
        if SQLCODE <> Ø then do
           MESS = 'Error on DB2 update rc ' RC ' sqlcode ' SQLCODE
           return
        end
     end
  end
 /*-------------------------------------------------------------------*/
 /* If we are connected to DB2 ( DB2CON = 1) then updates  */
 /* have been performed, 'SIGNOFF SYNC' required for COMMIT*/
 /*-------------------------------------------------------------------*/
  if DB2CON = 1 then do
     DB2CON = Ø
     address db2 "SIGNOFF SYNC"
     if SQLCODE <> Ø then do
        MESS = 'Error on DB2 SIGNOFF SYNC sqlcode ' SQLCODE
        return
     end
     MESS = 'Column statistics updated'
     EUPD = 1
  end
  else
     MESS = 'No updates required for column statistics'
return

 /*-------------------------*/
 /* Decode Lo2key/Hi2key */
 /*-------------------------*/
EB_DECODE:
 /*------------------------------------------ */
 /* If colcard = -1 set fields to default */
 /*------------------------------------------ */
  if CCARD = -1     then do
     CCDATAD = '        '
     return
  end
  if CCDATAE = '4040404040404040' then do
     CCDATAD = '        '
     return
  end
 /*--------------------*/
 /* Character Decode */
 /*--------------------*/
  if CTYPE = 'CHAR' then do
     CCDATAD = X2C(CCDATAE)
     return
  end
```

```
/*-------------------*/
/* Decimal Decode */
/*-------------------*/
 if CTYPE = 'DECIMAL' then do
    if left(CCDATAE,1) = 'F' then
       DEFLD = '+' || substr(CCDATAE,2)
    else do
       DEFLD = '-'
       do I = 2 BY 1 TO 16
          wkf = x2d(substr(CCDATAE,I,1))
          wkf = 15 - wkf
          wkf = d2x(wkf,1)
          DEFLD = DEFLD || wkf
       end
    end
    CSIZE = CLEN + 1
    CCDATAD = left(DEFLD,CSIZE)
    return
 end
/*-------------------*/
/* Integer Decode */
/*-------------------*/
 if CTYPE = 'INTEGER' then do
    DEFLD = ''
    wkf = left(CCDATAE,8)
    wkf = x2b(wkf)
    if left(wkf,1) = '1' then
       wkf = 'Ø' || substr(wkf,2)
    else
       wkf = '1' || substr(wkf,2)
    defld = b2x(wkf)
    defld = x2d(defld,9)
    CCDATAD = DEFLD
    return
 end
/*------------------- */
/* Smallint Decode */
/*------------------- */
 if CTYPE = 'SMALLINT' then do
    DEFLD = ''
    wkf = left(CCDATAE,4)
    wkf = x2b(wkf)
    if left(wkf,1) = '1' then
       wkf = 'Ø' || substr(wkf,2)
    else
       wkf = '1' || substr(wkf,2)
    defld = b2x(wkf)
    defld = x2d(defld,5)
    CCDATAD = DEFLD
    return
```

```
  end
 /*————————————————————————————————————————————————————*/
 /* Default Decode data to hexadecimal encoded */
 /*————————————————————————————————————————————————————*/
  CCDATAD = CCDATAE
  return

EC_ENCODE:
 /*———————————————————*/
 /* Character Encode */
 /*———————————————————*/
  if CTYPE = 'CHAR' then do
     NCDATAD = left(NCDATAD,CLEN)
     NCDATAE = c2x(NCDATAD) || '4040404040404040'
     NCDATAE = left(NCDATAE,16)
     return
  end
 /*———————————————————*/
 /* Decimal Encode */
 /*———————————————————*/
  if CTYPE = 'DECIMAL' then do
     if CLEN > 15 then CLEN = 15
     CSIZE = CLEN + 1
     DSIGN = left(NCDATAD,1)
 /*———————————————*/
 /* Extract sign */
 /*———————————————*/
     if DSIGN ¬= '+' then
        if DSIGN ¬= '-' then do
           if DSIGN = ' ' then NCDADATD = '+' || substr(NCDATAD,2)
           else NCDATAD = '+' || NCDATAD
           DSIGN = '+'
        end
 /*———————————————— */
 /* Right justify */
 /*———————————————— */
     DLEN = length(NCDATAD)
     ZS   = CSIZE - DLEN
     if ZS > 0 then do until ZS < 1
        NCDATAD = DSIGN || '0' || substr(NCDATAD,2)
        ZS = ZS - 1
        end
     NCDATAD = left(NCDATAD,CSIZE)
 /*————————————*/
 /* Now encode */
 /*————————————*/
     if left(NCDATAD,1) = '-' then do
        DEFLD = 'D'
        do I = 2 BY 1 TO CSIZE
           wkf = x2d(substr(NCDATAD,I,1))
```

```
            wkf = 15 - wkf
            wkf = d2x(wkf,1)
            DEFLD = DEFLD || wkf
         end
         DEFLD = DEFLD || '4040404040404040'
      end
        else
          DEFLD = 'F' || substr(NCDATAD,2,CLEN) || '4040404040404040'
      NCDATAE = left(DEFLD,16)
      return
   end
  /*-------------------------------- */
  /* Smallint/Integer validation */
  /*-------------------------------- */
   if CTYPE = 'INTEGER' then do
      intlen = 9
      if left(NCDATAD,1) = '-' then intlen = intlen + 1
      if left(NCDATAD,1) = '+' then intlen = intlen + 1
      NCDATAD = left(NCDATAD,intlen)
   end
   if CTYPE = 'SMALLINT' then do
      if NCDATAD < -32768 then  NCDATAD = -32768
      if NCDATAD >  32767 then  NCDATAD =  32767
   end
  /*-----------------*/
  /* Integer Encode */
  /*-----------------*/
   if CTYPE = 'INTEGER' then do
      wkf = d2x(NCDATAD,8)
      wkf = x2b(wkf)
      if left(wkf,1) = '1' then
         wkf = '0' || substr(wkf,2)
      else
         wkf = '1' || substr(wkf,2)
      wkf = b2x(wkf)
      NCDATAE = wkf || '40404040'
      return
   end
  /*----------------- */
  /* Smallint Encode */
  /*----------------- */
   if CTYPE = 'SMALLINT' then do
      wkf = d2x(NCDATAD,4)
      wkf = x2b(wkf)
      if left(wkf,1) = '1' then
         wkf = '0' || substr(wkf,2)
      else
         wkf = '1' || substr(wkf,2)
      wkf = b2x(wkf)
      NCDATAE = wkf || '404040404040'
```

```
    return
 end
/*-------------- */
/* Date Encode */
/*-------------- */
 if CTYPE = 'DATE' then do
    NCDATAD = left(NCDATAD,8) || '40404040'
    NCDATAE = NCDATAD
    return
 end
/*------------------------------------------ */
/* Default Endcode data to input decoded */
/*------------------------------------------ */
 NCDATAE = NCDATAD || '4040404040404040'
 NCDATAE = left(NCDATAE,16)
 return
```

Liz Page
Independent Consultant (UK)

Driving and testing FIELDPROC

This article gives a PL/I program that drives and tests a DB2 Assembler FIELDPROC exit. Using this program, you can perform a lot of testing before linking your FIELDPROC to the exits library.

PL/I PROGRAM

```
//TSHVRD JOB (),'PLIXC',CLASS=A,MSGCLASS=X,NOTIFY=TSHVR
//PLI    EXEC IEL1CLG,
//         REGION=5000K,
//         PARM.PLI='OFFSET,INCLUDE,NODECK,LIST,AGGREGATE,ATTRIBUTES',
//         PARM.LKED='XREF,LIST,RENT,AMODE=31,RMODE=ANY',
//         PARM.GO='/'
//PLI.SYSLIB   DD DSN=TSHVR.SOURCE.TEST,DISP=SHR
//             DD DSN=TSHVR.INCLUDE.TOOLS,DISP=SHR
//             DD DSN=TSHVR.SOURCE.TOOLS,DISP=SHR
//PLI.SYSIN    DD *
* PROCESS LANGLVL(OS,SPROG);
* PROCESS SYSTEM(MVS);
 /*TST2UC TEST DBF2UC                                                 */
 /* drives and test DB2 Assembler FIELDPROC exit                      */
 TST2UC: PROC(PARMS) OPTIONS(MAIN REENTRANT) ;
```

```
/***************************/
/* RETURN CODES            */
/***************************/
DCL MYRC BIN FIXED(31) INIT(Ø);
/***************************/
/* CONSTANTS               */
/***************************/
DCL   $TRUE  BIT(1) STATIC INIT('1'B);
DCL   $FALSE BIT(1) STATIC INIT('Ø'B);
DCL   $ERROR   CHAR(8) STATIC INIT('*ERROR*');
DCL   $DEBUG CHAR(8)  STATIC INIT('*DEBUG*');
DCL   $INFO  CHAR(8)  STATIC INIT('*INFO *');
DCL   $PROG CHAR(8) STATIC INIT('TST2UC');
DCL   $HOMELIB CHAR(44) STATIC INIT('TSHVR.SOURCE.TEST');
/***************************/
/* EXTERNAL ENTRIES  ENTRY */
/***************************/
DCL DBF2UC    ENTRY OPTIONS(INTER,ASM,RETCODE);
DCL HVPC2X    ENTRY OPTIONS(INTER,ASM,RETCODE);
DCL HVPDMPX ENTRY(
   POINTER,
   BIN FIXED(31),
   FILE)
  RETURNS(BIN FIXED(31));
/***************************/
/* PASSED PARM PARMS PRM   */
/***************************/
DCL PARMS CHAR(*) VARYING;
/***************************/
/* BUILTIN                 */
/***************************/
%INCLUDE BUILTIN;
/***************************/
/* FILES                   */
/***************************/
DCL SYSPRINT FILE STREAM OUTPUT PRINT ;
/***************************/
/* WORK        VARIABLES   */
/***************************/
DCL TRCLVL BIN FIXED(31) INIT(Ø);
dcl loper bin fixed(15);
dcl loper_chars(2) char(1) based(addr(loper));
DCL PRMTST1 CHAR(14);
DCL PRMABEND CHAR(5) INIT('ABEND');
DCL PVDL BIN FIXED(15);
/**/
/* DSN41Ø.SDSNMACS(DSNDFPPB) */
DCL  FPBFENC BIN FIXED(15) STATIC INIT(Ø);
DCL  FPBFDEC BIN FIXED(15) STATIC INIT(4);
DCL  FPBFDEF BIN FIXED(15) STATIC INIT(8);
DCL  FPBFINV BIN FIXED(15) STATIC INIT(12);
```

```
DCL  FPBWKLND BIN FIXED(15) STATIC INIT(512);
DCL  FPVDTINT BIN FIXED(15) STATIC INIT(Ø);
DCL  FPVDTSMA BIN FIXED(15) STATIC INIT(4);
DCL  FPVDTFLT BIN FIXED(15) STATIC INIT(8);
DCL  FPVDTDEC BIN FIXED(15) STATIC INIT(12);
DCL  FPVDTCHR BIN FIXED(15) STATIC INIT(16);
DCL  FPVDTVCH BIN FIXED(15) STATIC INIT(2Ø);
DCL  FPVDTGRA BIN FIXED(15) STATIC INIT(24);
DCL  FPVDTVRA BIN FIXED(15) STATIC INIT(28);
DCL FPPL@ POINTER;
DCL Ø1 FPPL UNALIGNED,
 Ø2 FPPWORK  POINTER,
 Ø2 FPPFPIB  POINTER,
 Ø2 FPPCVD   POINTER,
 Ø2 FPPFVD   POINTER,
 Ø2 FPPPVL   POINTER;
DCL Ø1 FPIB UNALIGNED,
 Ø2 FPBFCODE BIN FIXED(15) INIT(Ø),
 Ø2 FPBWKLN  BIN FIXED(15) INIT(Ø),
 Ø2 FPBSORC  BIN FIXED(15) INIT(Ø),
 Ø2 FPBRTNC  CHAR(2) INIT(''),
 Ø2 FPBRSNC  CHAR(4) INIT(''),
 Ø2 FPBTOKP  POINTER INIT(SYSNULL),
 Ø2 RSRVD    CHAR(4) INIT(LOW(4));
DCL FPBTOK CHAR(4Ø) BASED(FPBTOKP);
DCL Ø1 FPVD UNALIGNED BASED,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE BIN FIXED(15),
  Ø3 FPVDVLEN BIN FIXED(15),
 Ø2 FPVDVALE CHAR(1);
DCL Ø1 FPVD_DEC UNALIGNED CTL,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE BIN FIXED(15),
  Ø3 FPVDPREC BIT(8),
  Ø3 FPVDSCAL BIT(8),
 Ø2 FPVDVALE CHAR(*);
DCL Ø1 CVD_DEF UNALIGNED ,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE BIN FIXED(15),
  Ø3 FPVDVLEN BIN FIXED(15);
DCL Ø1 CVD  UNALIGNED CTL,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE BIN FIXED(15),
  Ø3 FPVDVLEN BIN FIXED(15),
 Ø2 FPVDVALE CHAR(*);
DCL Ø1 CVD_VL  UNALIGNED CTL,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE BIN FIXED(15),
  Ø3 FPVDVLEN BIN FIXED(15),
 Ø2 FPVDVALE CHAR(*) VARYING;
DCL Ø1 FVD_DEF UNALIGNED ,
```

```
 Ø2 FPVDHD,
  Ø3 FPVDTYPE BIN FIXED(15),
  Ø3 FPVDVLEN BIN FIXED(15);
DCL Ø1 FVD  UNALIGNED CTL,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE BIN FIXED(15),
  Ø3 FPVDVLEN BIN FIXED(15),
 Ø2 FPVDVALE CHAR(*);
DCL Ø1 FVD_VL  UNALIGNED CTL,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE BIN FIXED(15),
  Ø3 FPVDVLEN BIN FIXED(15),
 Ø2 FPVDVALE CHAR(*) VARYING;
/* FIELDPROC PARAMETER VALUE LIST  */
DCL FPPVVDS@ POINTER;
DCL Ø1 FPPVL UNALIGNED,
 Ø2 FPPVLEN BIN FIXED(15),
 Ø2 FPPVAREA,
  Ø3 FPPVCNT BIN FIXED(15),
  Ø3 FPPVVDS  CHAR(254);    /* PVD'S */  /*254=max.length*/
DCL WA_DEF CHAR(512);
DCL WA_WORK CHAR(*) CTL;
DCL WA_WORK_LEN BIN FIXED(15);
/***************************/
/* CONDITIONS              */
/***************************/
/***************************/
/*       MAIN              */
/***************************/
PUT SKIP LIST('TEST@ DBF2UC NO PARMS ->ERROR');
CALL DBF2UC;
MYRC=PLIRETV();
PUT SKIP DATA(MYRC);
PUT SKIP LIST('TEST@ DBF2UC NO PARMS ->ERROR END');
/**/
PUT SKIP LIST('TEST@ DBF2UC INV CODE ->ERROR');
WA_DEF='';
FPBFCODE=FPBFINV;
FPBWKLN=CSTG(WA_DEF);
FPBTOKP=SYSNULL;
CALL DBF2UC(WA_DEF,FPIB,SYSNULL,SYSNULL,SYSNULL);
MYRC=PLIRETV();
PUT SKIP DATA(MYRC);
PUT SKIP DATA(FPIB);
PUT SKIP LIST('TEST@ DBF2UC INV CODE ->ERROR END');
/**/
PUT SKIP LIST('TEST@ DBF2UC INV TYPE ->ERROR');
WA_DEF='';
FPBFCODE=FPBFDEF;
FPBWKLN=CSTG(WA_DEF);
FPBTOKP=SYSNULL;
```

```
CVD_DEF.FPVDTYPE=255;  /*INVALID*/
CVD_DEF.FPVDVLEN=255;
CALL DBF2UC(WA_DEF,FPIB,CVD_DEF,FVD_DEF,FPPVL);
MYRC=PLIRETV();
PUT SKIP DATA(MYRC);
IF FPBTOKP¬=SYSNULL THEN
 PUT SKIP EDIT('FPBTOK=',FPBTOK)(A);
PUT SKIP DATA(FPIB);
PUT SKIP LIST('TEST@ DBF2UC INV TYPE ->ERROR END');
/**/
PUT SKIP LIST('TEST@ DBF2UC DEFINE WITH PRMABEND->NOK');
WA_DEF='';
FPBFCODE=FPBFDEF;
FPBWKLN=CSTG(WA_DEF);
FPBTOKP=SYSNULL;
CVD_DEF.FPVDTYPE=FPVDTCHR;
CVD_DEF.FPVDVLEN=255;
/**/
FPPVLEN=2;
FPPVCNT=Ø;
FPPVVDS@=ADDR(FPPVVDS);
PVDL=PVD_ADD(FPPVVDS@,FPVDTCHR,CSTG(PRMABEND),PRMABEND);
FPPVLEN=FPPVLEN+PVDL;
FPPVCNT=FPPVCNT+1;
FPPVVDS@=POINTERADD(FPPVVDS@,PVDL);
MYRC=HVPDMPX(ADDR(FPPVL),BINARY(fppvlen+2,31),SYSPRINT);
/**/
CALL DBF2UC(WA_DEF,FPIB,CVD_DEF,FVD_DEF,FPPVL);
MYRC=PLIRETV();
PUT SKIP DATA(MYRC);
IF FPBTOKP¬=SYSNULL THEN
 PUT SKIP EDIT('FPBTOK=',FPBTOK)(A);
PUT SKIP DATA(FPIB);
PUT SKIP DATA(FVD_DEF);
PUT SKIP LIST('TEST@ DBF2UC DEFINE WITH PRMABEND->NOK END');
/**/
PUT SKIP LIST('TEST@ DBF2UC DEFINE OK');
WA_DEF='';
FPBFCODE=FPBFDEF;
FPBWKLN=CSTG(WA_DEF);
FPBTOKP=SYSNULL;
CVD_DEF.FPVDTYPE=FPVDTCHR;
CVD_DEF.FPVDVLEN=255;
/**/
FPPVLEN=2;
FPPVCNT=Ø;
FPPVVDS@=ADDR(FPPVVDS);
PRMTST1='PRM 1 2 3';
PVDL=PVD_ADD(FPPVVDS@,FPVDTCHR,CSTG(PRMTST1),PRMTST1);
FPPVLEN=FPPVLEN+PVDL;
FPPVCNT=FPPVCNT+1;
```

```
FPPVVDS@=POINTERADD(FPPVVDS@,PVDL);
MYRC=HVPDMPX(ADDR(FPPVL),BINARY(fppvlen+2,31),SYSPRINT);
/**/
CALL DBF2UC(WA_DEF,FPIB,CVD_DEF,FVD_DEF,FPPVL);
MYRC=PLIRETV();
PUT SKIP DATA(MYRC);
IF FPBTOKP¬=SYSNULL THEN
 PUT SKIP EDIT('FPBTOK=',FPBTOK)(A);
PUT SKIP DATA(FPIB);
PUT SKIP DATA(FVD_DEF);
PUT SKIP LIST('TEST@ DBF2UC DEFINE OK END');
/**/
WA_WORK_LEN=FPIB.FPBWKLN;
/**/
PUT SKIP LIST('TEST@ DBF2UC ENCODE ');
ALLOC WA_WORK CHAR(WA_WORK_LEN);
WA_WORK='';
FPBFCODE=FPBFENC;
FPBWKLN=WA_WORK_LEN;
FPBTOKP=SYSNULL;
ALLOCATE
Ø1 CVD,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(16);
ALLOCATE
Ø1 FVD,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(16);
CVD.FPVDTYPE=FPVDTCHR;
CVD.FPVDVLEN=16;
cvd.fpvdvale='abcdefghijklmnops';
fvD.FPVDTYPE=FPVDTCHR;
fVD.FPVDVLEN=16;
fvd.fpvdvale='';
CALL DBF2UC(WA_DEF,FPIB,CVD,FVD,FPPVL);
MYRC=PLIRETV();
PUT SKIP DATA(MYRC);
IF FPBTOKP¬=SYSNULL THEN
 PUT SKIP EDIT('FPBTOK=',FPBTOK)(A);
PUT SKIP DATA(FPIB);
PUT SKIP DATA(FVD);
FREE CVD;
FREE FVD;
FREE WA_WORK;
PUT SKIP LIST('TEST@ DBF2UC ENCODE END');
```

```
/**/
PUT SKIP LIST('TEST@ DBF2UC DECODE');
ALLOC WA_WORK CHAR(WA_WORK_LEN);
WA_WORK='';
FPBFCODE=FPBFdec;
FPBWKLN=WA_WORK_LEN;
FPBTOKP=SYSNULL;
ALLOCATE
Ø1 CVD,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(16);
ALLOCATE
Ø1 FVD,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(16);
fvD.FPVDTYPE=FPVDTCHR;
fvd.FPVDVLEN=16;
fvd.fpvdvale='abcdefghijklmnops';
cvd.FPVDTYPE=FPVDTCHR;
cvd.FPVDVLEN=16;
cvd.fpvdvale='';
CALL DBF2UC(WA_DEF,FPIB,CVD,FVD,FPPVL);
MYRC=PLIRETV();
PUT SKIP DATA(MYRC);
IF FPBTOKP¬=SYSNULL THEN
 PUT SKIP EDIT('FPBTOK=',FPBTOK)(A);
PUT SKIP DATA(FPIB);
PUT SKIP DATA(cVD);
FREE CVD;
FREE FVD;
FREE WA_WORK;
PUT SKIP LIST('TEST@ DBF2UC DECODE END');
/**/
ALLOC WA_WORK CHAR(WA_WORK_LEN);
WA_WORK='';
FPBFCODE=FPBFdec;
FPBWKLN=WA_WORK_LEN;
FPBTOKP=SYSNULL;
ALLOCATE
Ø1 CVD,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(16);
ALLOCATE
Ø1 FVD,
```

```
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(16);
fvD.FPVDTYPE=FPVDTCHR;
fvd.FPVDVLEN=16;
fvd.fpvdvale='abcdefghijklmnops';
cvd.FPVDTYPE=FPVDTCHR;
cvd.FPVDVLEN=Ø8;  /*error ! */
cvd.fpvdvale='';
CALL DBF2UC(WA_DEF,FPIB,CVD,FVD,FPPVL);
MYRC=PLIRETV();
PUT SKIP DATA(MYRC);
IF FPBTOKP¬=SYSNULL THEN
 PUT SKIP EDIT('FPBTOK=',FPBTOK)(A);
PUT SKIP DATA(FPIB);
PUT SKIP DATA(cVD);
FREE CVD;
FREE FVD;
FREE WA_WORK;
/**/
PUT SKIP LIST('TEST@ DBF2UC ENCODE LEN > 255 ');
ALLOC WA_WORK CHAR(WA_WORK_LEN);
WA_WORK='';
FPBFCODE=FPBFENC;
FPBWKLN=WA_WORK_LEN;
FPBTOKP=SYSNULL;
ALLOCATE
Ø1 CVD,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(3ØØ);
ALLOCATE
Ø1 FVD,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(3ØØ);
CVD.FPVDTYPE=FPVDTCHR;
CVD.FPVDVLEN=3ØØ;
DO LOPER=1 TO CVD.FPVDVLEN;
 SUBSTR(CVD.FPVDVALE,LOPER,1)=LOPER_CHARS(2);
END;
FVD.FPVDTYPE=FPVDTCHR;
FVD.FPVDVLEN=CVD.FPVDVLEN;
FVD.FPVDVALE='';
CALL DBF2UC(WA_DEF,FPIB,CVD,FVD,FPPVL);
MYRC=PLIRETV();
PUT SKIP DATA(MYRC);
```

```
IF FPBTOKP¬=SYSNULL THEN
 PUT SKIP EDIT('FPBTOK=',FPBTOK)(A);
PUT SKIP DATA(FPIB);
PUT SKIP DATA(CVD);
PUT SKIP DATA(FVD);
FREE CVD;
FREE FVD;
FREE WA_WORK;
PUT SKIP LIST('TEST@ DBF2UC ENCODE LEN > 255 END');
/**/
PUT SKIP LIST('TEST@ DBF2UC ENCODE VAR LEN = Ø ');
ALLOC WA_WORK CHAR(WA_WORK_LEN);
WA_WORK='';
FPBFCODE=FPBFENC;
FPBWKLN=WA_WORK_LEN;
FPBTOKP=SYSNULL;
ALLOCATE
Ø1 CVD_VL,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(257) ;
ALLOCATE
Ø1 FVD_VL,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(257) ;
CVD_VL.FPVDTYPE=FPVDTVCH;
CVD_VL.FPVDVALE='';
CVD_VL.FPVDVLEN=STG(CVD_VL.FPVDVALE)-2;
FVD_VL.FPVDTYPE=CVD_VL.FPVDTYPE;
FVD_VL.FPVDVLEN=CVD_VL.FPVDVLEN;
FVD_VL.FPVDVALE='';
PUT SKIP EDIT('STG CVD:',STG(CVD_VL.FPVDVALE))(A);
PUT SKIP EDIT('LEN CVD:',LENGTH(CVD_VL.FPVDVALE))(A);
PUT SKIP EDIT('STG FVD:',STG(FVD_VL.FPVDVALE))(A);
PUT SKIP EDIT('LEN FVD:',LENGTH(FVD_VL.FPVDVALE))(A);
MYRC=HVPDMPX(ADDR(CVD_VL),
 BINARY(CSTG(CVD_VL),31),SYSPRINT);
CALL DBF2UC(WA_DEF,FPIB,CVD_VL,FVD_VL,FPPVL);
MYRC=PLIRETV();
PUT SKIP DATA(MYRC);
IF FPBTOKP¬=SYSNULL THEN
 PUT SKIP EDIT('FPBTOK=',FPBTOK)(A);
PUT SKIP DATA(FPIB);
PUT SKIP DATA(CVD_vl);
PUT SKIP EDIT('LEN:',LENGTH(CVD_VL.FPVDVALE))(A);
PUT SKIP DATA(FVD_vl);
PUT SKIP EDIT('LEN:',LENGTH(FVD_VL.FPVDVALE))(A);
```

```
MYRC=HVPDMPX(ADDR(FVD_VL),
 BINARY(CSTG(FVD_VL),31),SYSPRINT);
FREE CVD_vl;
FREE FVD_vl;
FREE WA_WORK;
PUT SKIP LIST('TEST@ DBF2UC ENCODE VAR LEN = Ø END');
/**/
PUT SKIP LIST('TEST@ DBF2UC ENCODE VAR LEN     ');
ALLOC WA_WORK CHAR(WA_WORK_LEN);
WA_WORK='';
FPBFCODE=FPBFENC;
FPBWKLN=WA_WORK_LEN;
FPBTOKP=SYSNULL;
ALLOCATE
Ø1 CVD_VL,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(257) ;
ALLOCATE
Ø1 FVD_VL,
 Ø2 FPVDHD,
  Ø3 FPVDTYPE ,
  Ø3 FPVDVLEN ,
 Ø2 FPVDVALE CHAR(257) ;
CVD_VL.FPVDTYPE=FPVDTVCH;
CVD_VL.FPVDVALE='abcd';
CVD_VL.FPVDVLEN=STG(CVD_VL.FPVDVALE)-2;
FVD_VL.FPVDTYPE=CVD_VL.FPVDTYPE;
FVD_VL.FPVDVLEN=CVD_VL.FPVDVLEN;
FVD_VL.FPVDVALE='';
PUT SKIP EDIT('STG CVD:',STG(CVD_VL.FPVDVALE))(A);
PUT SKIP EDIT('LEN CVD:',LENGTH(CVD_VL.FPVDVALE))(A);
PUT SKIP EDIT('STG FVD:',STG(FVD_VL.FPVDVALE))(A);
PUT SKIP EDIT('LEN FVD:',LENGTH(FVD_VL.FPVDVALE))(A);
MYRC=HVPDMPX(ADDR(CVD_VL),
 BINARY(CSTG(CVD_VL),31),SYSPRINT);
CALL DBF2UC(WA_DEF,FPIB,CVD_VL,FVD_VL,FPPVL);
MYRC=PLIRETV();
PUT SKIP DATA(MYRC);
IF FPBTOKP¬=SYSNULL THEN
 PUT SKIP EDIT('FPBTOK=',FPBTOK)(A);
PUT SKIP DATA(FPIB);
PUT SKIP DATA(CVD_vl);
PUT SKIP EDIT('LEN:',LENGTH(CVD_VL.FPVDVALE))(A);
PUT SKIP DATA(FVD_vl);
PUT SKIP EDIT('LEN:',LENGTH(FVD_VL.FPVDVALE))(A);
MYRC=HVPDMPX(ADDR(FVD_VL),
 BINARY(CSTG(FVD_VL),31),SYSPRINT);
FREE CVD_vl;
```

```
 FREE FVD_vl;
 FREE WA_WORK;
 PUT SKIP LIST('TEST@ DBF2UC ENCODE VAR LEN = Ø END');
 /***************************/
 /* END                     */
 /***************************/
 L_RETURN:
 ON ERROR SYSTEM;  /* AVOID ERROR LOOP */
 CALL PLIRETC(MYRC);
 RETURN;
 /***************************/
 /* SUBROUTINES             */
 /***************************/
 PVD_ADD:PROC(FPPVVDS@,TP,LN,V) RETURNS(BIN FIXED(31));
  DCL FPPVVDS@ POINTER;
  DCL TP BIN FIXED(15);
  DCL LN BIN FIXED(15);
  DCL V CHAR(*);
  DCL BF31 BIN FIXED(31) BASED;
  DCL PVDL BIN FIXED(31) INIT(Ø);
  DCL TMP@ POINTER;
  TMP@=FPPVVDS@;
  PVDL=4+2+2+LN;
  TMP@->BF31=PVDL;
  TMP@=POINTERADD(TMP@,4);
  TMP@->FPVD.FPVDTYPE=TP;
  TMP@->FPVD.FPVDVLEN=LN;
  SUBSTR(TMP@->FPVD.FPVDVALE,1,LN)=V;
  RETURN(PVDL);
 END PVD_ADD;
 END TST2UC ;
/*
//LKED.SYSLIB  DD
//             DD DISP=SHR,DSN=TSHVR.PGM.LOAD
//             DD DISP=SHR,DSN=TSHVR.PGM.TOOLS
//LKED.SYSIN   DD *
 INCLUDE TOOLSMOD(DBF2UC)
 INCLUDE TOOLSMOD(HVPC2X)
 INCLUDE TOOLSMOD(HVPDMPX)
/*
//LKED.TOOLSMOD DD DISP=SHR,DSN=TSHVR.TEST.OBJMOD
//              DD DISP=SHR,DSN=TSHVR.TOOLS.OBJMOD
//GO.STEPLIB DD
//           DD DISP=SHR,DSN=TSHVR.PGM.TOOLS
//GO.SYSPRINT DD SYSOUT=X,OUTLIM=3ØØØØ   ,CHARS=(GFC),HOLD=YES
```

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

Herman Vierendeels
Systems Programmer (Belgium)

Verifying start-up parameters – part 2

This month we conclude the program that reads the values of ZPARM in the DB2 control blocks, converts the parameter values to report format, and writes to a report file.

```
*
SPRMØ13Ø DS    ØH
         MVC   PRGFINST,=C' NO'
*
SPRMØ135 DS    ØH
         TM    SPRMREGF,X'4Ø'          RGFDEDPL CHECK
         BNO   SPRMØ14Ø
         MVC   PRGFDEDP,=C'YES'
         B     SPRMØ145
*
SPRMØ14Ø DS    ØH
         MVC   PRGFDEDP,=C' NO'
*
SPRMØ145 DS    ØH
         TM    SPRMREGF,X'2Ø'          RGFFULLQ CHECK
         BNO   SPRMØ15Ø
         MVC   PRGFFULL,=C'YES'
         B     SPRMØ155
*
SPRMØ15Ø DS    ØH
         MVC   PRGFFULL,=C' NO'
*
SPRMØ155 DS    ØH
         TM    SPRMREGF,X'1Ø'          RGFDEFLT CHECK
         BNO   SPRMØ16Ø
         MVC   PRGFDEFL,=C'ACCEPT'
         B     SPRMØ17Ø
*
SPRMØ16Ø DS    ØH
         TM    SPRMREGF,X'Ø1'
         BNO   SPRMØ165
         MVC   PRGFDEFL,=C'  APPL'
         B     SPRMØ17Ø
*
SPRMØ165 DS    ØH
         MVC   PRGFDEFL,=C'REJECT'
*
SPRMØ17Ø DS    ØH
         MVC   PRGFESCP,SPRMREGE
         TM    SPRMTYP,X'8Ø'           SITETYPE CHECK
         BNO   SPRMØ175
         MVC   PSITETYP,=C'   LOCALSITE'
```

```
         B     SPRM0180
*
SPRM0175 DS    0H
         MVC   PSITETYP,=C'RECOVERYSITE'
*
SPRM0180 DS    0H
         SR    R8,R8                  CLEAR WORK REGISTER
         L     R9,SPRMSORP            SRTPOOL
         D     R8,=F'1024'            DIVIDE BY 1024
         CVD   R9,PACKWK01            CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PSRTPOOL,EDWORK10
         MVC   PSYSADM,SPRMSADM       SYSADM
         MVC   PSYSADM2,SPRMADM2      SYSADM2
         MVC   PSYSOPR1,SPRMOPR1      SYSOPR1
         MVC   PSYSOPR2,SPRMOPR2      SYSOPR2
         LH    R7,SPRMUTO             UTIMOUT
         CVD   R7,PACKWK01            CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PUTIMOUT,EDWORK10
         LA    R8,PRTSPRM             SET PRTSPRM ADDR
         MVC   LOOPCNT,=X'0019'       SET LOOPCNT
         LH    R9,LOOPCNT
         BAL   R14,WRTRTN
*
SPRMEXT  DS    0H
         L     R14,SPRMSAVE           SET RETURN ADDR
         BR    R14                    RETURN NEXT INST ADDR
*
SPRMSAVE DS     F'0'
         EJECT
*
WRTRTN   DS    0H
         ST    R14,WRTSAVE            SAVE RETURN ADDR
         MVI   OUTREC+0,C' '          CLEAR OUTREC
         MVC   OUTREC+1(79),OUTREC+0
         PUT   ZPARMDD,OUTREC         WRITE
*
WRT0000  DS    0H
         MVC   OUTREC+0(80),0(R8)     MOVE PRT RECORD
         PUT   ZPARMDD,OUTREC         WRITE
         LA    R8,80(R8)              POINT TO NEXT PRT AREA
         BCT   R9,WRT0000             MORE RECORD TO PRINT
*
WRTEXT   DS    0H
         L     R14,WRTSAVE            SET RETURN ADDR
         BR    R14                    RETURN NEXT INST ADDR
*
WRTSAVE  DS    F'0'
```

```
         EJECT
*
****************************************
* WORK AREA AND CONSTANTS                *
****************************************
         SPACE
         DS    ØF
SAVEAREA DS    18F'Ø'                  MY SAVE AREA
RETCODE  DC    F'8'                    RETURN CODE
PARAM    DC    CL4' '                  DB2 NAME FROM PARM LIST
PACKWKØ1 DS    PL8
PACKWKØ2 DS    PL8
EDWORK1Ø DS    CL1Ø
LOOPCNT  DC    XL2'ØØØØ'
*************************************************************************
*        OUTREC RECORD     -  PARAMETER DSECT                           *
*************************************************************************
OUTREC   DC    CL8Ø' '                 OUTREC AREA
         SPACE
PRTARVP  DS    ØCL8Ø                     ARVP PRINT AREA
         DC    CL4Ø' **** DSN6ARVP PARAMETER LIST ****      '
         DC    CL4Ø' '
         DC    CL2' '
         DC    CL1Ø'ALCUNIT : '
         DC    CL7' '
PALCUNIT DC    CL3' '                  CYL/TRK/BLK
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'ARCWTOR : '
         DC    CL7' '
PARCWTOR DC    CL3' '                  WTOR YES/NO
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'ARCRETN : '
PARCRETN DC    CL1Ø' '                 RETENTION PERIOD
         DC    CL Ø3'DAY'
         DC    CL15' '
         DC    CL2' '
         DC    CL1Ø'BLKSIZE : '
PBLKSIZE DC    CL1Ø' '                 BLKSIZE
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'CATALOG : '
         DC    CL7' '
PCATALOG DC    CL3' '                  CATALOG YES/NO
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'COMPACT : '
         DC    CL7' '
PCOMPACT DC    CL3' '                  COMPACT YES/NO
         DC    CL18' '
```

```
         DC    CL2' '
         DC    CL10'PRIQTY  : '
PPRIQTY  DC    CL10' '                 PRIMARY QUANTITY
         DC    CL18' '
         DC    CL2' '
         DC    CL10'SECQTY  : '
PSECQTY  DC    CL10' '                 SECONDARY QUANTITY
         DC    CL18' '
         DC    CL2' '
         DC    CL10'PROTECT : '
         DC    CL7' '
PPROTECT DC    CL3' '                  PROTECT YES/NO
         DC    CL18' '
         DC    CL2' '
         DC    CL10'QUIESCE : '
PQUIESCE DC    CL10' '                 QUIESCE
         DC    CL03'SEC'
         DC    CL15' '
         DC    CL2' '
         DC    CL10'TSTAMP  : '
         DC    CL7' '
PTSTAMP  DC    CL3' '                  TIMESTAMP YES/NO
         DC    CL18' '
         DC    CL2' '
         DC    CL10'UNIT    : '
         DC    CL6' '
PUNIT    DC    CL8' '                  UNIT1
         DC    CL14' '
         DC    CL2' '
         DC    CL10'UNIT2   : '
         DC    CL6' '
PUNIT2   DC    CL8' '                  UNIT2
         DC    CL54' '
         DC    CL2' '
         DC    CL10'ARCPFX1 : '
PARCPFX1 DC    CL35' '                 PERFIX1 NAME
         DC    CL33' '
         DC    CL2' '
         DC    CL10'ARCPFX2 : '
PARCPFX2 DC    CL35' '                 PERFIX2 NAME
         DC    CL33' '
         SPACE
*
PRTLOGP  DS    ØCL8Ø                   LOGP PRINT AREA
         DC    CL4Ø' **** DSN6LOGP PARAMETER LIST ****      '
         DC    CL4Ø' '
         DC    CL2' '
         DC    CL10'DEALLCT : '
PDEALLC1 DC    CL10' '                 DEALLOCATE TIME : MINUTE
         DC    CLØ3'MIN'
         DC    CL1' '
```

```
PDEALLC2 DC    CL03' '                 DEALLOCATE TIME : SECOND
         DC    CL03'SEC'
         DC    CL8' '
         DC    CL2' '
         DC    CL10'INBUFF  : '
PINBUFF  DC    CL10' '                 INPUT BUFFER SIZE
         DC    CL01'K'
         DC    CL17' '
         DC    CL2' '
         DC    CL10'TWOACTV : '
         DC    CL7' '
PTWOACTV DC    CL03'   '               TWOACTV YES/NO
         DC    CL18' '
         DC    CL2' '
         DC    CL10'TWOARCH : '
         DC    CL7' '
PTWOARCH DC    CL03'   '               TWOARCH YES/NO
         DC    CL18' '
         DC    CL2' '
         DC    CL10'OUTBUFF : '
POUTBUFF DC    CL10' '                 OUTPUT BUFFER SIZE
         DC    CL01'K'
         DC    CL17' '
         DC    CL2' '
         DC    CL10'WRTHRSH : '
PWRTHRSH DC    CL10' '                 WRITE THRESHOLD
         DC    CL01'%'
         DC    CL17' '
         DC    CL2' '
         DC    CL10'MAXARCH : '
PMAXARCH DC    CL10' '                 MAX ARCH COUNT IN BSDS
         DC    CL18' '
         DC    CL2' '
         DC    CL10'MAXRTU  : '
PMAXRTU  DC    CL10' '                 MAX NO. OF UNIT FOR ARCH READ
         DC    CL18' '
*
PRTSYSP  DS    0CL80                   SYSP PRINT AREA
         DC    CL40' **** DSN6SYSP PARAMETER LIST ****      '
         DC    CL40' '
         DC    CL2' '
         DC    CL10'CTHREAD : '
PCTHREAD DC    CL10' '                 CONCURRENT THREAD(LOCAL)
         DC    CL18' '
         DC    CL2' '
         DC    CL10'IDFORE  : '
PIDFORE  DC    CL10' '                 FOREGROUND THREAD
         DC    CL18' '
         DC    CL2' '
         DC    CL10'IDBACK  : '
PIDBACK  DC    CL10' '                 BACKGROUND THREAD
```

```
         DC    CL18' '
         DC    CL2' '
         DC    CL10'CONDBAT : '
PCONDBAT DC    CL10' '                 CONCURR INACT + ACT THD(REMOTE)
         DC    CL18' '
         DC    CL2' '
         DC    CL10'MAXDBAT : '
PMAXDBAT DC    CL10' '                 CONCURR ACT THD(REMOTE)
         DC    CL58' '
         DC    CL2' '
         DC    CL10'DLDFREQ : '
PDLDFREQ DC    CL10' '                 DLDFREQ
         DC    CL18' '
         DC    CL2' '
         DC    CL10'LOGLOAD : '
PLOGLOAD DC    CL10' '                 LOGLOAD
         DC    CL18' '
         DC    CL2' '
         DC    CL10'AUDITST : '
         DC    CL7' '
PAUDITST DC    CL03' '                 AUDITST
         DC    CL18' '
         DC    CL2' '
         DC    CL10'SMFACCT : '
         DC    CL7' '
PSMFACCT DC    CL03' '                 SMFACCT
         DC    CL18' '
         DC    CL2' '
         DC    CL10'SMFSTAT : '
         DC    CL7' '
PSMFSTAT DC    CL03' '                 SMFSTAT
         DC    CL18' '
         DC    CL2' '
         DC    CL10'STATIME : '
PSTATIME DC    CL10' '                 STATIME
         DC    CL3'MIN'
         DC    CL15' '
         DC    CL2' '
         DC    CL10'MON     : '
         DC    CL7' '
PMON     DC    CL03' '                 MON(MONITOR TRACE)
         DC    CL18' '
         DC    CL2' '
         DC    CL10'MONSIZE : '
PMONSIZE DC    CL10' '                 MONSIZE
         DC    CL18' '
         DC    CL2' '
         DC    CL10'TRACSTR : '
         DC    CL7' '
PTRACSTR DC    CL03' '                 TRACSTR
         DC    CL18' '
```

```
         DC    CL2' '
         DC    CL10'TRACTBL : '
PTRACTBL DC    CL10' '                 TRACTBL
         DC    CL01'K'
         DC    CL17' '
         DC    CL2' '
         DC    CL10'RLF     : '
         DC    CL7' '
PRLF     DC    CL03' '                 RLF
         DC    CL18' '
         DC    CL2' '
         DC    CL10'RLFTBL  : '
         DC    CL08' '
PRLFTBL  DC    CL02' '                 RLFTBL
         DC    CL18' '
         DC    CL2' '
         DC    CL10'RLFERR  : '
         DC    CL03' '
PRLFERR  DC    CL07' '                 RLFERR
         DC    CL18' '
         DC    CL2' '
         DC    CL10'RLFAUTH : '
         DC    CL03' '
PRLFAUTH DC    CL07' '                 RLFAUTH
         DC    CL18' '
         DC    CL2' '
         DC    CL10'STORPROC: '
         DC    CL2' '
PSTORPRO DC    CL08' '                 STORPROC NAME
         DC    CL18' '
         DC    CL2' '
         DC    CL10'STORMAXB: '
PSTORMAX DC    CL10' '                 STORMAX
         DC    CL18' '
         DC    CL2' '
         DC    CL10'STORTIME: '
         DC    CL4' '
PSTORTIM DC    CL06' '                 STORTIME
         DC    CL03'SEC'
         DC    CL55' '
*
PRTGRP   DS    0CL80                   GRP PRINT AREA
         DC    CL40' **** DSN6GRP  PARAMETER LIST ****      '
         DC    CL40' '
         DC    CL2' '
         DC    CL10'DSHARE  : '
         DC    CL7' '
PDSHARE  DC    CL03' '                 DSHARE
         DC    CL58' '
         DC    CL2' '
         DC    CL10'GRPNAME : '
```

```
         DC    CL2' '
PGRPNAME DC    CLØ8' '                  GRPNAME
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'MEMBNAME: '
         DC    CL2' '
PMEMBNAM DC    CLØ8' '                  MEMBNAME
         DC    CL18' '
*
PRTFAC   DS    ØCL8Ø                    FAC PRINT AREA
         DC    CL4Ø' **** DSN6FAC  PARAMETER LIST ****      '
         DC    CL4Ø' '
         DC    CL2' '
         DC    CL1Ø'DDF     : '
         DC    CL3' '
PDDF     DC    CLØ7' '                  DDF
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'CMTSTAT : '
         DC    CL2' '
PCMTSTAT DC    CLØ8' '                  CMTSTAT
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'IDTHTOIN: '
PIDTHTOI DC    CL1Ø' '                  IDTHTOIN
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'RESYNC  : '
PRESYNC  DC    CL1Ø' '                  RESYNC
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'RLFERRD : '
         DC    CL3' '
PRLFERRD DC    CLØ7' '                  RLFERRD
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'RLFERRDT: '
PRLFERRT DC    CL1Ø' '                  RLF LIMIT TIME
         DC    CL18' '
*
PRTSPRM  DS    ØCL8Ø                    SPRM PRINT AREA
         DC    CL4Ø' **** DSN6SPRM PARAMETER LIST ****      '
         DC    CL4Ø' '
*
PRESTART DC    CL4Ø' '
         DC    CL4Ø' '
         DC    CL2' '
         DC    CL1Ø'ABEXP   : '
         DC    CL7' '
PABEXP   DC    CLØ3' '                  ABEXP
         DC    CL18' '
```

```
         DC    CL2' '
         DC    CL1Ø'ABIND   : '
         DC    CL7' '
PABIND   DC    CLØ3' '                ABIND
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'AUTH    : '
         DC    CL7' '
PAUTH    DC    CLØ3' '                AUTH
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'AUTHCACH: '
PAUTHCAC DC    CL1Ø' '                AUTHCACH
         DC    CLØ4'BYTE'
         DC    CL14' '
         DC    CL2' '
         DC    CL1Ø'BINDNV  : '
         DC    CL3' '
PBINDNV  DC    CLØ7' '                BINDNV
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'BMPTOUT : '
PBMPTOUT DC    CL1Ø' '                BMPTOUT
         DC    CLØ3'SEC'
         DC    CL15' '
         DC    CL2' '
         DC    CL1Ø'CATALOG : '
         DC    CL2' '
PSPRMCAT DC    CLØ8' '                CATALOG
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'CDSSRDEF: '
         DC    CL7' '
PCDSSRDE DC    CLØ3' '                CDSSRDEF
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'CHGDC   : '
         DC    CL7' '
PCHGDC   DC    CLØ3' '                CHGDC
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'DECDIV3 : '
         DC    CL7' '
PDECDIV3 DC    CLØ3' '                DECDIV3
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'DEFIXTP : '
         DC    CL9' '
PDEFIXTP DC    CLØ1' '                DEFIXTP
         DC    CL18' '
         DC    CL2' '
```

```
         DC    CL10'DEFLTID : '
         DC    CL3' '
PDEFLTID DC    CL07' '                 DEFLTID
         DC    CL18' '
         DC    CL2' '
         DC    CL10'DLITOUT : '
PDLITOUT DC    CL10' '                 DLITOUT
         DC    CL03'SEC'
         DC    CL15' '
         DC    CL2' '
         DC    CL10'DSMAX   : '
PDSMAX   DC    CL10' '                 DSMAX
         DC    CL18' '
         DC    CL2' '
         DC    CL10'EDMPOOL : '
PEDMPOOL DC    CL10' '                 EDMPOOL
         DC    CL01'K'
         DC    CL17' '
         DC    CL2' '
         DC    CL10'EDPROP  : '
         DC    CL7' '
PEDPROP  DC    CL03' '                 EDPROP
         DC    CL18' '
         DC    CL2' '
         DC    CL10'HOPAUTH : '
         DC    CL7' '
PHOPAUTH DC    CL03' '                 HOPAUTH
         DC    CL18' '
         DC    CL2' '
         DC    CL10'IRLMAUT : '
         DC    CL7' '
PIRLMAUT DC    CL03' '                 IRLMAUT
         DC    CL18' '
         DC    CL2' '
         DC    CL10'IRLMPRC : '
         DC    CL2' '
PIRLMPRC DC    CL08' '                 IRLMPRC
         DC    CL18' '
         DC    CL2' '
         DC    CL10'IRLMSID : '
         DC    CL6' '
PIRLMSID DC    CL04' '                 IRLMSID
         DC    CL18' '
         DC    CL2' '
         DC    CL10'IRLMRWT : '
PIRLMRWT DC    CL10' '                 IRLMRWT
         DC    CL03'SEC'
         DC    CL15' '
         DC    CL2' '
         DC    CL10'IRLMSWT : '
PIRLMSWT DC    CL10' '                 IRLMSWT
```

```
         DC    CLØ3'SEC'
         DC    CL15' '
         DC    CL2' '
         DC    CL1Ø'MAXRBLK : '
PMAXRBLK DC    CL1Ø' '                 MAXRBLK
         DC    CL01'K'
         DC    CL17' '
         DC    CL2' '
         DC    CL1Ø'NUMLKTS : '
PNUMLKTS DC    CL1Ø' '                 NUMLKTS
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'NUMLKUS : '
PNUMLKUS DC    CL1Ø' '                 NUMLKUS
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'RECALL  : '
         DC    CL7' '
PRECALL  DC    CLØ3' '                 RECALL
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'RECALLD : '
PRECALLD DC    CL1Ø' '                 RECALLD
         DC    CLØ3'SEC'
         DC    CL15' '
         DC    CL2' '
         DC    CL1Ø'RGFCOLID: '
         DC    CL2' '
PRGFCOLI DC    CLØ8' '                 RGFCOLID
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'RGFDBNAM: '
         DC    CL2' '
PRGFDBNA DC    CLØ8' '                 RGFDBNAM
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'RGFDEDPL: '
         DC    CL7' '
PRGFDEDP DC    CLØ3' '                 RGFDEDPL
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'RGFDEFLT: '
         DC    CL4' '
PRGFDEFL DC    CLØ6' '                 RGFDEFLT
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'RGFESCP : '
         DC    CL9' '
PRGFESCP DC    CL01' '                 RGFESCP
         DC    CL18' '
         DC    CL2' '
```

```
         DC    CL10'RGFFULLQ: '
         DC    CL7' '
PRGFFULL DC    CL03' '                 RGFFULLQ
         DC    CL18' '
         DC    CL2' '
         DC    CL10'RGFINSTL: '
         DC    CL7' '
PRGFINST DC    CL03' '                 RGFINSTL
         DC    CL18' '
         DC    CL2' '
         DC    CL10'RGFNMORT: '
PRGFNMOR DC    CL17' '                 RGFNMORT
         DC    CL11' '
         DC    CL2' '
         DC    CL10'RGFINSTL: '
PRGFNMPR DC    CL17' '                 RGFNMPRT
         DC    CL11' '
         DC    CL2' '
         DC    CL10'RRULOCK : '
         DC    CL7' '
PRRULOCK DC    CL03' '                 RRULOCK
         DC    CL18' '
         DC    CL2' '
         DC    CL10'SEQCACH : '
         DC    CL4' '
PSEQCACH DC    CL06' '                 SEQCACH
         DC    CL18' '
         DC    CL2' '
         DC    CL10'SITETYPE: '
PSITETYP DC    CL12' '                 SITETYPE
         DC    CL16' '
         DC    CL2' '
         DC    CL10'SRTPOOL : '
PSRTPOOL DC    CL10' '                 SRTPOOL
         DC    CL01'K'
         DC    CL17' '
         DC    CL2' '
         DC    CL10'SYSADM  : '
         DC    CL2' '
PSYSADM  DC    CL08' '                 SYSADM
         DC    CL18' '
         DC    CL2' '
         DC    CL10'SYSADM2 : '
         DC    CL2' '
PSYSADM2 DC    CL08' '                 SYSADM2
         DC    CL18' '
         DC    CL2' '
         DC    CL10'SYSOPR1 : '
         DC    CL2' '
PSYSOPR1 DC    CL08' '                 SYSOPR1
         DC    CL18' '
```

```
         DC    CL2' '
         DC    CL1Ø'SYSADM2 : '
         DC    CL2' '
PSYSOPR2 DC    CLØ8' '                 SYSOPR2
         DC    CL18' '
         DC    CL2' '
         DC    CL1Ø'UTIMOUT : '
PUTIMOUT DC    CL1Ø' '                 UTIMOUT
         DC    CLØ5'TIMES'
         DC    CL53' '
*
*****************************************
* DCB = ZPARMDD                         *
*****************************************
         SPACE
ZPARMDD  DCB   DSORG=PS,MACRF=PM,DDNAME=ZPARMDD,LRECL=8Ø
         LTORG
*****************************************
* DB2 CONTROL BLOCK                     *
*****************************************
         YREGS
         DSN6ARVP
         DSN6LOGP
         DSN6SYSP
         DSN6GRP
         DSN6FAC
         DSN6SPRM
         END
```

SAMPLE OUTPUT

```
 **** DSN6ARVP PARAMETER LIST ****
  ALCUNIT :        CYL                  ARCWTOR :         NO
  ARCRETN :         2ØDAY               BLKSIZE :     28,672
  CATALOG :        YES                  COMPACT :         NO
  PRIQTY  :        1ØØ                  SECQTY  :         1Ø
  PROTECT :         NO                  QUIESCE :          5SEC
  TSTAMP  :         NO                  UNIT    :       MCTG
  UNIT2   :
  ARCPFX1 : D2T1.ARCHLOG1
  ARCPFX2 : D2T1.ARCHLOG2

 **** DSN6LOGP PARAMETER LIST ****
  DEALLCT :          ØMIN   ØSEC        INBUFF  :         28K
  TWOACTV :         NO                  TWOARCH :         NO
  OUTBUFF :        4ØØK                 WRTHRSH :         2Ø%
  MAXARCH :        1ØØ                  MAXRTU  :          2

 **** DSN6SYSP PARAMETER LIST ****
  CTHREAD :        5ØØ                  IDFORE  :         7Ø
```

```
 IDBACK  :        100                    CONDBAT :         10
 MAXDBAT :         10
 DLDFREQ :          5                    LOGLOAD :     50,000
 AUDITST :         NO                    SMFACCT :         NO
 SMFSTAT :         NO                    STATIME :        30MIN
 MON     :         NO                    MONSIZE :      8,192
 TRACSTR :         NO                    TRACTBL :        16K
 RLF     :         NO                    RLFTBL  :         01
 RLFERR  :          0                    RLFAUTH :     SYSIBM
 STORPROC:                               STORMAXB:          0
 STORTIME:       180SEC

**** DSN6GRP  PARAMETER LIST ****
 DSHARE  :         NO
 GRPNAME :                               MEMBNAME:

**** DSN6FAC  PARAMETER LIST ****
 DDF     :    COMMAND                    CMTSTAT :   ACTIVE
 IDTHTOIN:          0                    RESYNC  :          2
 RLFERRD :    NOLIMIT                    RLFERRDT:          0

**** DSN6SPRM PARAMETER LIST ****
 RESTART, ALL
 ABEXP   :         NO                    ABIND   :        YES
 AUTH    :        YES                    AUTHCACH:      1,024BYTE
 BINDNV  :    BINDADD                    BMPTOUT :          0SEC
 CATALOG :   D2T1                        CDSSRDEF:        ANY
 CHGDC   :         NO                    DECDIV3 :         NO
 DEFIXTP :          2                    DEFLTID :    IBMUSER
 DLITOUT :          0SEC                 DSMAX   :      1,000
 EDMPOOL :      8,192K                   EDPROP  :         NO
 HOPAUTH :         NO                    IRLMAUT :        YES
 IRLMPRC :   D2T1IRLM                    IRLMSID :       JRLM
 IRLMRWT :         60SEC                 IRLMSWT :        300SEC
 MAXRBLK :      9,168K                   NUMLKTS :      5,000
 NUMLKUS :     20,000                    RECALL  :        YES
 RECALLD :        120SEC                 RGFCOLID:   DSNRGCOL
 RGFDBNAM:   DSNRGFDB                    RGFDEDPL:         NO
 RGFDEFLT:     ACCEPT                    RGFESCP :
 RGFFULLQ:        YES                    RGFINSTL:         NO
 RGFNMORT: DSN_REGISTER_OBJT             RGFINSTL: DSN_REGISTER_APPL
 RRULOCK :        YES                    SEQCACH :        SEQ
 SITETYPE:    LOCALSITE                  SRTPOOL :      1,833K
 SYSADM  :   XDB2403                     SYSADM2 :   XDBJ010
 SYSOPR1 :   SYSOPR                      SYSADM2 :   SYSOPR
 UTIMOUT :          6TIMES
```

Young-Ho Kim
DB2 Systems Programmer
LG-EDS Systems (Korea)

HiT Software has announced HiT JDBC/ DB2 Enterprise, DB2 and DB2/400 Java middleware optimized via native IBM server support. It includes JDBC drivers for DB2 and DB2/400 and the HiT dbProxy+ Server which expands IBM OS/390 and OS/400 server flexibility for Java development. Corporate e-commerce Java developers can now deploy DB2 applets faster and minimize client response time across their networks.

HiT JDBC/DB2 Enterprise provides universal servers and clients with DB2 data access via the JDBC API. It supports IBM Distributed Relational Database Architecture and OS/400 optimized database server protocols. Full Type 4 JDBC functionality allows the Java developer to use HiT middleware within servlets or client applets. A small footprint ensures fast applet deployment and application access. dbProxy+ Server allows Java applets DB2 access when this data resides on a different server to the applet's Web server.

For further information contact:
HiT Software, 1975 Hamilton Avenue, Suite 1, San Jose, CA 95125, USA.
Tel: (408) 369 7290.
HiT Software, PO Box 2, Farnborough, Hants, GU14 0NF, UK.
Tel: (01252) 522995.
URL: htp://www.hit.com.

* * *

Centura Software has announced Version 4.1 of its SQLHost software tool, enabling connectivity from user desktops to enterprise DB2 data on the IBM System/390 platform.

SQLHost 4.1 is geared towards developing high-concurrency client/server and Web applications accessing DB2 and legacy data sources on the mainframe, providing end users with controlled, direct access to corporate data, minimizing installation and configuration time with straightforward set-up procedures, moving DB2 data to and from local database servers, and building GUI-based applications that connect to DB2 data.

For further information contact:
Centura Software, Lunar House, Globe Park, Marlow, Bucks, SL7 1LW, UK.
Tel: (01628) 478333.
http://www.centurasoft.com.

* * *

Xephon has just launched four weekly news services covering the following subject areas:

- Data Centre
- Distributed Systems
- Networks
- Software

Each week, subscribers receive, by e-mail, a short news bulletin consisting of a list of items; each item has a link to the page on our Web site that contains the corresponding article. Each news bulletin also carries links to the main industry news stories of the week.

To subscribe to one or more of these news services, or review recent articles, point your browser at http://www.xephon.com/newz.html.

* * *