78

DB2

update

April 1999

In this issue

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.

Creating DB2 statements on-the-fly

As a DBA, from time to time you may need to generate a lot of commands or DDL as a result of a change, error, update, or something else happening in your DB2 system.

Sometimes you will be lucky and your existing routines or tools can handle that exercise, but this little gem of JCL has saved me a lot of time ever since DB2 Version 1.3!

The problem is quite simple – you need to do a change involving a lot of DB2 objects.

STEP ONE

Based on information in the DB2 catalog, you extract the necessary statements via DSNTEP2.

You then filter away everything from the output in step one that is not a valid statement, using COPY with the INCLUDE parameter, and pass the valid statements to the DSN processor (or DSNTEP2 with DDL) for execution.

The sample JCL given below illustrates the concept:

```
//DSNTEP2  EXEC PGM=IKJEFT1A,DYNAMNBR=2Ø
//STEPLIB  DD DSN=DB2.DSNLOAD,DISP=SHR
//         DD DSN=DB2.DSNEXIT,DISP=SHR
//*******************************************************************
//* THIS STEP GENERATES THE VALID STATEMENTS
//*
//* THIS (SIMPLE) EXAMPLE EXTRACTS PACKAGES TO BE "REBINDED" FOLLOWING
//* A TABLE CHANGE
//*
//*******************************************************************
//SYSTSPRT DD SYSOUT=*
//*SYSPRINT DD SYSOUT=*
//SYSPRINT DD DSN=&&SYSPRINT,
//            SPACE=(TRK,(1,5),RLSE),
//            DCB=(RECFM=FB,LRECL=8Ø),
//            DISP=(,PASS),
//            UNIT=SYSDA
//SYSUDUMP DD SYSOUT=U
//SYSTSIN  DD  *
  DSN SYSTEM(DSNT)
```

```
  RUN PROGRAM(DSNTEP2) PLAN(DSNTEP2)
  END
/*
//SYSIN    DD  *
SELECT DISTINCT
' REBIND PACKAGE ('||SUBSTR(DCOLLID,1,8)||'.'||DNAME||')'
|| '                                                    '
FROM SYSIBM.SYSPACKDEP
WHERE BNAME LIKE 'EY00500T%'
AND BQUALIFIER = 'TEST'
ORDER BY 1
/*
//
//*******************************************************************
//* THIS STEP IDENTIFIES VALID STATEMENTS AND EXCLUDES EVERYTHING ELSE
//*******************************************************************
//SORT     EXEC PGM=SORT
//SYSOUT   DD SYSOUT=*
//SORTIN   DD DSN=*.DSNTEP2.SYSPRINT,DISP=(SHR,DELETE)
//*SORTOUT  DD SYSOUT=*
//SORTOUT  DD DSN=&&SORTOUT,
//            SPACE=(TRK,(1,5),RLSE),
//            DCB=(RECFM=FB,LRECL=80),
//            DISP=(,PASS),
//            UNIT=SYSDA
//SYSIN    DD *
 OPTION COPY
 INCLUDE COND=(27,6,BI,EQ,C'REBIND')
 OUTREC FIELDS=(1C' ',27,35,44C' ')
/*
//*************************************************************
//REBIND   EXEC PGM=IKJEFT1A,DYNAMNBR=20
//STEPLIB  DD DSN=DB2.DSNLOAD,DISP=SHR
//         DD DSN=DB2.DSNEXIT,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=U
//SYSTSIN  DD  *
  DSN SYSTEM(DSNT)
//         DD DSN=*.SORT.SORTOUT,DISP=(SHR,DELETE)
//         DD  *
  END
/*
```

COMMENTS ON JCL

In step one, we execute DSNTEP2, providing the relevant SQL statements as input. The resulting statements are written to SYSPRINT, which we allocate as a temporary dataset, thus allowing the output to be passed on to the next step.

The following sample from SYSPRINT from step one shows the statements to be extracted:

```
***INPUT STATEMENT:  SELECT DISTINCT , REBIND PACKAGE
(,||SUBSTR(DCOLLID,1,8)|| )||'.'||DNAME||')' || ,
  , FROM SYSIBM.SYSPACKDEP WHERE BNAME LIKE
  ,EY00500T%' AND BQUALIFIER   = ,TEST' ORDER BY 1 ;
                      +-----------------------------------------
                      |
                      +-----------------------------------------
                    1_|  REBIND PACKAGE (EA0001TP.EE13400E)
                    2_|  REBIND PACKAGE (EE0001TC.EE13400E)
                    3_|  REBIND PACKAGE (EX0001TC.EX05800E)
                    4_|  REBIND PACKAGE (EX0001TC.EX11800E)
                    5_|  REBIND PACKAGE (EX0001TP.EX00900E)
                    6_|  REBIND PACKAGE (EX0001TT.EX00900E)
                    7_|  REBIND PACKAGE (EY0001TC.EY19102E)
                    8_|  REBIND PACKAGE (EY0001TC.EY19103E)
                    9_|  REBIND PACKAGE (EY0001TC.EY23110E)
                      +-----------------------------------------
SUCCESSFUL RETRIEVAL OF  9 ROW(S)
```

STEP TWO

In step two, we use the SYSPRINT file from step one as SORTIN, filtering out any unwanted lines (any non-valid line/character), and pass on valid DB2 statements to step three.

This step can cause some trouble at first, but this is what you need to know:

- The first time you use the JCL procedure, you need to know where your valid statements start in your SYSPRINT dataset. To discover this, you comment out the dataset allocation for SYSPRINT and use ‘SYSPRINT DD SYSOUT=*’, and submit your job.

  In your spool system, you browse the output from step one (in the example above we want the lines starting with REBIND), set column numbering on, and figure out where your statement starts and how many characters long it is. These two numbers are the parameters for the INCLUDE condition in the COPY step.

- For subsequent use of the JCL procedure, instead of doing the above exercise every time you submit the JCL with a new SQL statement, you adjust the number of blanks being concatenated, keeping the resulting length of the statement constant. In this

way, you avoid the ‘random’ placement you will otherwise get from DSNTEP2.

STEP THREE

Step three takes the SORTOUT file as input, in this example REBIND PACKAGE.

The observant reader might have noticed an ‘extra’ ‘//*SYSPRINT’ commented out in step one, and an ‘extra’ ‘//*SORTOUT’ in step two. These will not be needed if everything works; however, if you have trouble for some reason, use these DD statements and comment out the one allocating a temporary file. If you follow the recommendations for the subsequent use of the JCL procedure correctly, there should be little need for these statements.

EXECUTING DDL

If you want to execute DDL in step three (for example, altering indexes to TYPE 2), all you need to do is change step three to an execution of DSNTEP2 (as in step one), and use the following SYSIN statement:

```
//SYSIN DD DSN=*.SORT.SORTOUT,DISP=(SHR,DELETE)
```

Example below :

```
//DSNTEP2  EXEC PGM=IKJEFT1A,DYNAMNBR=20
//STEPLIB  DD DSN=DB2.DSNLOAD,DISP=SHR
//         DD DSN=DB2.DSNEXIT,DISP=SHR
//*************************************************************
//* THIS STEP GENERATES THE VALID STATEMENTS
//*
//* THIS (SIMPLE) EXAMPLE EXTRACTS INDEXES TO BE CONVERTED TO
//* TYPE 2
//*
//*************************************************************
//SYSTSPRT DD SYSOUT=*
//*SYSPRINT DD SYSOUT=*
//SYSPRINT DD DSN=&&SYSPRINT,
//          SPACE=(TRK,(1,5),RLSE),
//          DCB=(RECFM=FB,LRECL=80),
//          DISP=(,PASS),
//DSNTEP2  EXEC PGM=IKJEFT1A,DYNAMNBR=20
//STEPLIB  DD DSN=DB2.DSNLOAD,DISP=SHR
//         DD DSN=DB2.DSNEXIT,DISP=SHR
```

```
//*******************************************************************
//* THIS STEP GENERATES THE VALID STATEMENTS
//*
//* THIS (SIMPLE) EXAMPLE EXTRACTS INDEXES TO BE CONVERTED TO
//* TYPE 2
//*
//*******************************************************************
//SYSTSPRT DD SYSOUT=*
//*SYSPRINT DD SYSOUT=*
//SYSPRINT DD DSN=&&SYSPRINT,
//          SPACE=(TRK,(1,5),RLSE),
//          DCB=(RECFM=FB,LRECL=8Ø),
//          DISP=(,PASS),
//          UNIT=SYSDA
//SYSUDUMP DD SYSOUT=U
//SYSTSIN  DD  *
  DSN SYSTEM(DSNT)
  RUN PROGRAM(DSNTEP2) PLAN(DSNTEP2)
  END
/*
//SYSIN    DD  *
SELECT 'ALTER INDEX TEST.'
|| NAME || ' CONVERT TO TYPE 2;'
|| '                              '
 FROM SYSIBM.SYSINDEXES
 WHERE NAME LIKE 'EY%'
 AND CREATOR = 'TEST'
 ORDER BY 1
/*
//*******************************************************************
//* THIS STEP IDENTIFIES VALID STATEMENTS AND EXCLUDES EVERYTHING ELSE
//*******************************************************************
//SORT     EXEC PGM=SORT
//SYSOUT   DD SYSOUT=*
//SORTIN   DD DSN=*.DSNTEP2.SYSPRINT,DISP=(SHR,DELETE)
//SORTOUT  DD SYSOUT=*
//*ORTOUT  DD DSN=&&SORTOUT,
//*           SPACE=(TRK,(1,5),RLSE),
//*           DCB=(RECFM=FB,LRECL=8Ø),
//*           DISP=(,PASS),
//*           UNIT=SYSDA
//SYSIN    DD *
 OPTION COPY
 INCLUDE COND=(27,5,BI,EQ,C'ALTER')
 OUTREC FIELDS=(1C' ',27,45,34C' ')
/*
//
//*************************************************************
//* REBIND
//*************************************************************
//ALTER   EXEC PGM=IKJEFT1A,DYNAMNBR=2Ø
//STEPLIB  DD DSN=DB2.DSNLOAD,DISP=SHR
```

```
//         DD DSN=DB2.DSNEXIT,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=U
//SYSTSIN  DD  *
  DSN SYSTEM(DSNT)
//         DD DSN=*.SORT.SORTOUT,DISP=(SHR,DELETE)
//         DD  *
  END
/*
```

Kim Hjortholm
Repository administrator
Kommunedata (Denmark)

DB2 catalog statistics update REXX EXEC

CSUPD is a REXX EXEC that can be used to update DB2 catalog statistics.

When updating catalog statistics via QMF or SPUFI, great care is required to ensure that all indexes are identified and updated in line with the table statistics. CSUPD takes away a lot of the drudgery involved in the updating of catalog statistics. After the DB2 subsystem and table have been entered, CSUPD will retrieve tablespace, table, index, and column statistics. This information will be displayed in one of three ISPF panels. Repeating data for indexes and columns is stored in two ISPF tables.

The EXEC will not update the table and tablespace statistics without updating the index statistics, to ensure that these are kept in line.

It is assumed that all indexes are unique, so that SYSINDEXES FULLKEYCARD is always maintained as the same value as SYSTABLES CARD. All index columns are retrieved and updateable. CSUPD will decode HIGH2KEY and LOW2KEY on display, and encode them on update.

Column statistics can be updated independently of table, tablespace, and index statistics.

The REXX/DB2 interface used in this EXEC is RxD2, supplied by Boole & Babbage with their performance monitor InTune. The EXEC will need converting to the REXX/DB2 interface used at your site.

Please read the comments at the beginning of the EXEC. Variables are also described within the EXEC to help with maintenance and modifications.

CSTAB – TABLE STATISTICS ISPF PANEL

```
)ATTR
 _ TYPE(input)  COLOR(green)
 } TYPE(output)
 £ TYPE(output) INTENS(low)
 # TYPE(output) INTENS(high) color(red)
 ¬ TYPE(text)   INTENS(low)  COLOR(yellow)
 + TYPE(text)   INTENS(low)  COLOR(turq)
 % TYPE(text)   INTENS(high)
 | TYPE(text)   INTENS(high) color(red)
 { TYPE(output) INTENS(non)
)BODY
% ------    DB2 Table  -  Catalog Statistics Update   --------
%COMMAND ===>_zcmd      +
¬Enter 'COL' for column stats, 'IND' for index stats
#MESS                                                          +
+
+ DB2 Subsystem:_DB2S+ {PDB2S+     Record length . . .}RECLEN+
+ Creator . . .:_CRTR    +         Variable length . .}VARLEN+
+ Table . . . .:_TBNAM           + Variable len used :_VARUSE+ {PVARUSE+
+ Rows  . . . .:_ROWS    +         Actual Length . . .}LENUSE+
+ {PCRTR   {PTBNAM            +
+ --Statistics Current  ------ Calculated-
+   Cardinality £OCARD   +       }PROWS   + ¬Enter table details,
+   Npages . . .£ONPGES  +       }NPAGES  + ¬then enter number of rows,
+   Pctpages . .£OPCPAGES+       }PCTPAGES+ ¬new stats are calculated
+
+   Database . .£DBNAM   +
+   Tablespace .£TSPACE            +
+   Tables . . .£TBNUM+
+   Nactive  . .£OACTIVE +          }NACTIVE +
+
+
|Indexes must be processed before table & tablespace stats are updated.
)PROC
)END
```

CSIDX – INDEX STATISTICS ISPF PANEL

```
)ATTR
 _ TYPE(input)  COLOR(green)
 } TYPE(output)
 £ TYPE(output) INTENS(low)
 # TYPE(output) INTENS(high) color(red)
```

```
 ¬ TYPE(text)   INTENS(low)  COLOR(yellow)
 + TYPE(text)   INTENS(low)  COLOR(turq)
 % TYPE(text)   INTENS(high)
 { TYPE(output) INTENS(non)
)BODY
% ————————   DB2 Index Statistics Update   ——————————
%COMMAND ===>_zcmd         +
¬Enter 'UPD' to update table/tablespace and index data
¬Enter 'COL' to process columns and bypass table/index update
#MESS                                                                +
+ Creator . . . }CRTR    +          DB2 Subsystem. £DB2S+
+ Table . . . . }TBNAM             +Database . . . £DBNAM   +
+ Rows. . . . . }ROWS   +           Tablespace . . £TSPACE            +
+ Original Rows £OCARD   +
+                                 ¬Index Stats are calculated from Rows
+            C/Ratio  -Firstkeycard-  Full                       Update
+Index       New Curr New    Current  Keycard  Nleaf  Nlvls Len Pending
)MODEL
£Z          _Z  }Z   _Z      £Z      }Z       }Z     }Z    £Z    }Z
)INIT
 .ZVARS = '(INAME NCRIO OCRIO NFSKCRD FSKCRD FLKCRD NLEAF NLVLS IXLEN
IUPD)'
)PROC
)END
```

CSCLM – COLUMN STATISTICS ISPF PANEL

```
)ATTR
 _ TYPE(input)  COLOR(green)
 } TYPE(output)
 £ TYPE(output) INTENS(low)
 # TYPE(output) INTENS(high) color(red)
 ¬ TYPE(text)   INTENS(low)  COLOR(yellow)
 + TYPE(text)   INTENS(low)  COLOR(turq)
 % TYPE(text)   INTENS(high)
 { TYPE(output) INTENS(non)
)BODY
% ————————    DB2 Catalog Statistics Update    ——————————
%COMMAND ===>_zcmd      +
¬Enter 'UPD' to update column statistics
#MESS                                                              +
+ DB2 Subsystem }DB2S+     Table .}TBNAM    +
+ Creator . . . }CRTR    + Rows. .}ROWS    +
+                          —— New Value ———          Current Value
+Column                    Decoded          Encoded/hex      Encoded/hex
)MODEL
£Z              £Z       _Z              }Z               £Z
)INIT
  .ZVARS = '(CNAME CFLD NCDATAD NCDATAE CCDATAE)'
)PROC
)END
```

CSUPD – CATALOG STATISTICS UPDATE REXX EXEC

```
/* REXX  */
/*                                                                  */
/*  This EXEC will retrieve and update catalog statistics           */
/*  for a given DB2 table.                                          */
/*                                                                  */
/*  The following ISPF/PDF panels are used by the EXEC.             */
/*                                                                  */
/*  CSTAB - handling table statistics                               */
/*  CSIDX - handling index statistics                               */
/*  CSCLM - handling columns statistics for index columns           */
/*                                                                  */
/*  Repeating data is held on the following ISPF tables.            */
/*                                                                  */
/*  T£IDX - index statistics                                        */
/*  T£CLM - columns statistics                                      */
/*                                                                  */
/*------------------------------------------------------------------*/
/*                                                                  */
/*  CSTAB - Table statistics panel                                  */
/*                                                                  */
/*  The statistic for the number of rows on the table can be        */
/*  modified. The effect of the modification is propagated          */
/*  to related table and tablespace statistics.                     */
/*                                                                  */
/*  The variable length in the row length is appoximated            */
/*  from current catalog statistics.                                */
/*                                                                  */
/*  The user can choose to update index stats, column stats,        */
/*  or both.                                                        */
/*                                                                  */
/*  The table/tablespace stats cannot be updated without            */
/*  updating associated index stats.                                */
/*                                                                  */
/*  CSIDX - Index statistics panel                                  */
/*  T£IDX - Index statistics table                                  */
/*                                                                  */
/*  Fullkeycard, Nleaf, and Nlevels are calculated from the         */
/*  number of rows entered for the table and no further             */
/*  modification of these fields is catered for.                    */
/*  Firstkeycard and Clusterratio are amendable.                    */
/*                                                                  */
/*  An update request for this panel will update table,             */
/*  tablespace, and index statistics and then display the           */
/*  column statistics.                                              */
/*                                                                  */
/*  The column statistics can be displayed directly bypassing       */
/*  the table, tablespace, and index statistics update.             */
/*                                                                  */
/*  CSCLM - Column statistics panel                                 */
/*  T£CLM - Column statistics table                                 */
```

```
/*                                                                   */
/*  The panel displays index-related columns.                        */
/*  The fields shown are column name, column type, Colcard,          */
/*  Low2key, and High2key. Colcard,Low2key, and High2key are         */
/*  updateable. Low2key and High2key are decoded to allow            */
/*  easier use.                                                      */
/*                                                                   */
/*  If the updates from the CSIDX panel have been actioned           */
/*  this panel will display the updated number of rows or            */
/*  else the original number of rows.                                */
/*                                                                   */
/* ------------------------------------------------------------------*/
/*                                                                   */
/* The following are CSTAB panel variables.                          */
/*                                                                   */
/*  Input variables                                                  */
/*                                                                   */
/*   ZCMD     - command line input                                   */
/*   DB2S     - DB2 Subsystem                                        */
/*   CRTR     - Creator (table qualifier)                            */
/*   TBNAM    - Table name                                           */
/*   ROWS     - Number of rows                                       */
/*   VARUSE   - Variable length used on average in the row           */
/*                                                                   */
/*  Output variables                                                 */
/*                                                                   */
/*   MESS     - message                                              */
/*   PDB2S    - DB2 Subsystem previously entered                     */
/*   PCRTR    - Creator previously entered                           */
/*   PTBNAM   - Table name previously entered                        */
/*   PVARUSE  - Variable length previously entered                   */
/*   PROWS    - Number of rows previously entered                    */
/*                                                                   */
/*   OCARD    - Original cardinality                                 */
/*   ONPGES   - Original NPAGES                                      */
/*   OPCPAGE  - Original PCTPAGES                                    */
/*   RECLEN   - Record length of row including variables             */
/*   VARLEN   - Total length of variable columns                     */
/*   LENUSE   - Actual row dependent on variable length used         */
/*   NPAGES   - Calculated NPAGES                                    */
/*   PCTPAGES - Calculated PCTPAGES                                  */
/*                                                                   */
/*   DBNAM    - Database name                                        */
/*   TSPACE   - Tablespace name                                      */
/*   TBNUM    - Number of tables within tablespace                   */
/*   OACTIVE  - Original NACTIVE                                     */
/*   NACTIVE  - Calculated NACTIVE                                   */
/*                                                                   */
/*  Commands                                                         */
/*                                                                   */
/*   'COL'    - display column statistics panel CSCLM                */
/*   'IND'    - display index statistics panel CSIDX                 */
```

```
/*   'END'    - exit                                                 */
/*   'EXIT'   - exit                                                 */
/*                                                                   */
/* ----------------------------------------------------------------*/
/*                                                                   */
/* The following are CSIDX panel variables.                          */
/*                                                                   */
/*  Input variables                                                  */
/*                                                                   */
/*   ZCMD     - command line input                                   */
/*   NCRIO    - New CLUSTERRATIO held on table T£IDX                 */
/*   NFSKCRD  - New FIRSTKEYCARD held on table T£IDX                 */
/*                                                                   */
/*  Output variables                                                 */
/*                                                                   */
/*   INAME    - Index name, held on table T£IDX                      */
/*   OCRIO    - Old CLUSTERRATIO, held on table T£IDX                */
/*   FSKCRD   - Old FIRSTKEYCARD, held on table T£IDX                */
/*   FLKCRD   - FULLKEYCARD, held on table T£IDX,                    */
/*              from entered number of rows                          */
/*   NLEAF    - Calculated NLEAF, held on table T£IDX                */
/*   NLEVELS  - Calculated NLEVELS, held on table T£IDX              */
/*   IXLEN    - Length of index, held on table T£IDX                 */
/*   IUPD     - Update pending flag, held on table T£IDX             */
/*              set when FLKCARD or NLEAF or NLEVELS differ          */
/*              from current catalog statistics                      */
/*                                                                   */
/*   CRTR     - from CSTAB panel                                     */
/*   TBNAM    - from CSTAB panel                                     */
/*   ROWS     - from CSTAB panel                                     */
/*   OCARD    - from CSTAB panel                                     */
/*   DB2S     - from CSTAB panel                                     */
/*   DBNAM    - from CSTAB panel                                     */
/*   TSPACE   - from CSTAB panel                                     */
/*   MESS     - message                                              */
/*                                                                   */
/*  Commands                                                         */
/*                                                                   */
/*   'COL'    - do not action any updates                            */
/*              display column statistics panel CSCLM                */
/*   'UPD'    - update table,tablespace & index statistics           */
/*              display column statistics panel CSCLM                */
/*   'END'    - return to panel CSTAB                                */
/*   'EXIT'   - exit                                                 */
/*                                                                   */
/* ----------------------------------------------------------------*/
/*                                                                   */
/* The following are CSCLM panel variables.                          */
/*                                                                   */
/*  Input variables                                                  */
/*                                                                   */
/*   ZCMD     - command line input                                   */
```

```
/*   NCDATAD  - New decoded column data, held on table T£CLM */
/*    this field, depending on CFLD, contains               */
/*     CCARD  - New COLCARD                                 */
/*     NL2KEY - New Low2key                                 */
/*     NH2KEY - New High2key                                */
/*                                                          */
/*  Output variables                                        */
/*                                                          */
/*   CRTR     - from CSTAB panel                            */
/*   TBNAM    - from CSTAB panel                            */
/*   DB2S     - from CSTAB panel                            */
/*   ROWS     - current CARD for table                      */
/*                                                          */
/*   CNAME    - held on table, T£CLM, this field can hold   */
/*              Column name or column type and length       */
/*   CFLD     - held on table, T£CLM, this field can hold   */
/*              'Colcard', 'Lo2key', or 'Hi2key'            */
/*                                                          */
/*   NCDATAE  - New column data encoded in hexadecimal      */
/*    held on table T£CLM                                   */
/*    this field, depending on CFLD, contains               */
/*       New COLCARD                                        */
/*       New Low2key encoded                                */
/*       New High2key encoded                               */
/*                                                          */
/*   CCDATAE - Current column data encoded,                 */
/*             held on table T£CLM                          */
/*    this field, depending on CFLD, contains               */
/*       New COLCARD                                        */
/*       New Low2key encoded                                */
/*       New High2key encoded                               */
/*                                                          */
/*   MESS     - message                                     */
/*                                                          */
/*  Commands                                                */
/*                                                          */
/*   'UPD'    - update column statistics                    */
/*              return to panel CSTAB                       */
/*   'END'    - return to panel CSTAB                       */
/*   'EXIT'   - exit                                        */
/*                                                          */
/* ---------------------------------------------------------*/
/*                                                          */
/*----------------------------------------------------------*/
/*  Define ISPF panel library required                      */
/*----------------------------------------------------------*/
  address ispexec
  "libdef ispplib dataset id('DEVGRP.ISPPLIB')"
  PCRTR  = ' '
  PTBNAM = ' '
  PDB2S  = ' '
  DB2S   = ' '
```

```
   CRTR   = ' '
   TBNAM  = ' '
   DB2CON = Ø
   ROWS   = Ø
   PROWS  = Ø
   OCARD  = Ø
   RECLEN = Ø
   VARLEN = Ø
   VARUSE = Ø
   PVARUSE= Ø
   LENUSE = Ø
   call A_LOOP
 /*———————————————————————————*/
 /*  Free ISPF panel library */
 /*———————————————————————————*/
   address ispexec 'libdef ispplib'
   exit

 /*———————————————————*/
 /*  Process panel */
 /*———————————————————*/
A_LOOP:
   do forever
      ZCMD = ''
      address ispexec "display panel(CSTAB)"
      if rc ¬= Ø then leave
      if ZCMD = exit then exit
      if ZCMD = end  then leave
      call AA_VALIDATE
      if MESS ¬= ' ' then iterate
      address ispexec "CONTROL ERRORS RETURN"
      if RC > Ø then say "ISPF ERROR, CANNOT CONTINUE"
      £RC = Ø
 /*————————————————————————*/
 /* Define error handler */
 /*————————————————————————*/
      signal on failure
      if UENV(DB2) ¬= "OK" then do
         MESS = "Error - Unable to enable RXDB2"
         address ispexec "DISPLAY PANEL(CSTAB)"
         return 16
         end

      if TBNAM <> PTBNAM | CRTR  <> PCRTR | DB2S  <> PDB2S
      then do
         call B_RETRIEVE
         if MESS = ' ' then call C_CALC
         iterate
      end
      if ROWS <> PROWS | VARUSE <> PVARUSE
      then do
         call C_CALC
```

```
         iterate
      end
      if ZCMD = 'IND' then do
         ZCMD = ''
         DUPD = Ø
         call D_INDEX
         if DUPD = 1 then do
            DUPD = Ø
            call E_COLUMNS
            SVAR = VARUSE
            call B_RETRIEVE
            VARUSE = SVAR
            call C_CALC
         end
      end
      if ZCMD = 'COL' then do
         ZCMD = ''
         ROWS = OCARD
         call E_COLUMNS
         ROWS = PROWS
      end
   address ispexec "TBDEL T£IDX"
   address ispexec "TBDEL T£CLM"
   end
   return

 /*———————————————————————— */
 /*  Validate CSTAB panel input */
 /*———————————————————————— */
AA_VALIDATE:
   MESS = ' '
 /*————————————————————————*/
 /*  Validate DB2 subsystem  */
 /*————————————————————————*/
   if DB2S = '' then DB2S = 'DB2T'
   if DB2S <> 'DB2T'
   then if DB2S <> 'DB2U'
   then if MESS = ' ' then
      MESS = 'DB2 Subsystem ' DB2S ' not set up for use'
 /*————————————————————————*/
 /*  Validate table qualifier */
 /*————————————————————————*/
   if CRTR    = '' then if MESS = ' ' then
      MESS = 'Enter Creator'
 /*————————————————————————*/
 /*  Validate table name    */
 /*————————————————————————*/
   if TBNAM   = '' then if MESS = ' ' then
      MESS = 'Enter Table Name'
 /*————————————————————————*/
 /*  Validate entered rows   */
 /*————————————————————————*/
```

```
   if ROWS    = '' then ROWS = OCARD
 /*————————————————————————————————*/
 /*  Validate variable length used */
 /*————————————————————————————————*/
   if VARUSE  = '' then VARUSE = VARLEN
   if VARUSE ¬> VARLEN then LENUSE = RECLEN - VARLEN + VARUSE
   else if MESS = ' ' then
      MESS = 'Variable length used cannot be > Variable Length'
   if MESS   ¬= ' ' then ZCMD = ''
   return

 /*————————————————————————————————————————————————————————*/
 /* Retrieve table & tablespace statistics from catalog */
 /*————————————————————————————————————————————————————————*/
B_RETRIEVE:
 /*——————————————— */
 /* Initialize data */
 /*——————————————— */
  ROWS    = Ø
  PROWS   = Ø
  OCARD   = Ø
  DBNAM   = ''
  TSPACE  = ''
  NPAGES  = Ø
  ONPGES  = Ø
  PCTPAGES= Ø
  OPCPAGES= Ø
  NACTIVE = Ø
  OACTIVE = Ø
  TBNUM   = Ø
  RECLEN  = Ø
  VARLEN  = Ø
  VARUSE  = Ø
  PVARUSE = Ø
  LENUSE  = Ø
 /*————————————————————————————————————————*/
 /* Connect to DB2 - set DB2 connect flag */
 /*————————————————————————————————————————*/
  address db2 "SIGNON" DB2S
  if RC > Ø then do
     MESS = "Error - Unable to connect to " DB2S
     RETURN16
     end
  DB2CON = 1
 /*————————————————————————————————————————————————————————*/
 /* Set up SQL statement to retrieve table/tablespace data.*/
 /* Declare and open cursor then fetch rows                */
 /*————————————————————————————————————————————————————————*/
  SLCT =      "SELECT A.DBNAME,A.TSNAME,A.CARD,A.NPAGES",
              ",A.PCTPAGES,A.RECLENGTH,B.NACTIVE,B.NTABLES",
              "FROM SYSIBM.SYSTABLES A,SYSIBM.SYSTABLESPACE B "
  WHRCLS =    "WHERE A.CREATOR='"CRTR"' AND A.NAME='"TBNAM"'",
```

```
             "AND A.DBNAME=B.DBNAME AND A.TSNAME=B.NAME"
 address db2 "DECLARE RXCSR1 CURSOR FOR",
             SLCT,
             WHRCLS
 if RC ¬= Ø then
    MESS = 'Error declaring cursor - RC ' RC
 else do
    address db2 "OPEN RXCSR1"
    address db2 "FETCH RXCSR1"
    if SQLCODE ¬= Ø then do
       MESS = CRTR||'.'||TBNAM||' DOES NOT EXIST IN '||DB2S
       address db2 "SIGNOFF"
       DB2CON = Ø
       return
    end
    else do
       DBNAM  = RXCSR1.1
       TSPACE = RXCSR1.2
       OCARD  = RXCSR1.3
       NPAGES = RXCSR1.4
       PCTPAGES= RXCSR1.5
       RECLEN  = RXCSR1.6
       NACTIVE = RXCSR1.7
       TBNUM   = RXCSR1.8
    end
    address db2 "CLOSE RXCSR1"
 end
/*———————————————————————————————————————————————————*/
/* Set up SQL statement to retrieve variable lengths from */
/* syscolumns. Declare and open cursor and fetch row.     */
/*———————————————————————————————————————————————————*/
 SLCT =      "SELECT SUM(LENGTH) FROM SYSIBM.SYSCOLUMNS"
 WHRE =    "WHERE TBCREATOR='"CRTR"' AND TBNAME='"TBNAM"'",
             "AND COLTYPE LIKE '%VAR%'"
 address db2 "DECLARE RXCSR3 CURSOR FOR",
             SLCT,
             WHRE
 if RC ¬= Ø then
    MESS = 'Error declaring cursor3 - RC ' RC
 else do
    address db2 "OPEN RXCSR3"

    address db2 "FETCH RXCSR3"
    if SQLCODE ¬= Ø then
       VARLEN = Ø
    else
       VARLEN = RXCSR3.1
       if VARLEN = '' then VARLEN = Ø
    address db2 "CLOSE RXCSR3"
 end
/*———————————————————————————————————————————————*/
```

```
 /* Disconnect from DB2 - reset DB2 connect flag */
 /*-------------------------------------------------------*/
  address db2 "SIGNOFF"
  DB2CON = Ø
 /*-------------------------------------------------------------------------*/
 /* Calculate variable leng usage from table cardinality and npages  */
 /* Calculate rows per page(rpp),row len used then variable len used */
 /*-------------------------------------------------------------------------*/
  VARUSE = VARLEN
  if VARUSE > Ø then
     if NPAGES > Ø then do
        RPP  = ( OCARD / NPAGES ) +.9
        RPP  = trunc(RPP)
        if RPP > 127 then RPP = 127
        LENUSE = 4Ø74 / RPP
        LENUSE = trunc(LENUSE)
        VARUSE = LENUSE - RECLEN + VARLEN
     end
  if VARUSE > VARLEN then VARUSE = VARLEN
 /*-----------------------------------*/
 /* Set control and panel fields */
 /*-----------------------------------*/
  PVARUSE= VARUSE
  LENUSE = RECLEN - VARLEN + VARUSE
  PROWS  = OCARD
  ROWS   = OCARD
  OACTIVE= NACTIVE
  ONPGES = NPAGES
  OPCPAGES=PCTPAGES
  PTBNAM = TBNAM
  PDB2S  = DB2S
  PCRTR  = CRTR
BYE:
  return £RC

/*------------------ */
/* ERROR HANDLING  */
/*------------------ */
FAILURE:
   £RC = RC
   say "** FAILURE - LINE" SIGL "**" SOURCELINE(SIGL)
   say "** RC=" £RC "£RSN=" £RSN
   say "** £DB2RC=" £DB2RC "£DB2RSN=" £DB2RSN
   say "** SQLCODE" SQLCODE "SQLERRD3" SQLERRD3 "SQLWARN" SQLWARN
   say "** SQLERRM" SQLERRM
   if SQLEMSG.Ø > Ø then do I = 1 TO SQLEMSG.Ø
      say SQLEMSG.I
      end
   signal BYE
return
```

```
 /*-------------------------*/
 /* Calculate statistics */
 /*-------------------------*/
C_CALC:
 /*-------------------------------- */
 /* Calculate rows per page RPP */
 /*-------------------------------- */
   LENUSE = RECLEN - VARLEN + VARUSE
   RPP  = 4Ø74 / LENUSE
   RPP  = trunc(RPP)
   if RPP > 127 then RPP = 127
   if ROWS < 1 then do
      NPAGES  = Ø
      NACTIVE = Ø
      PCTPAGES= Ø
   end
   else do
 /*---------------------------------*/
 /* Calculate NPAGES for table */
 /*---------------------------------*/
      NPAGES  = ( ROWS / RPP ) + Ø.99
      NPAGES  = trunc(NPAGES)
 /*-------------------------------------*/
 /* Calculate NACTIVE for tablespace */
 /*-------------------------------------*/
      if TBNUM > 1 then
         NACTIVE = NPAGES + OACTIVE - ONPGES
      else
         NACTIVE = NPAGES
      if NPAGES > NACTIVE then NACTIVE = NPAGES
 /*-------------------------------- */
 /* Calculate PCTPAGES for table */
 /*-------------------------------- */
      PCTPAGES= NPAGES * 1ØØ / NACTIVE
      PCTPAGES= trunc(PCTPAGES)
   end
 /*-------------------------------*/
 /* set panel control fields  */
 /*-------------------------------*/
   PROWS   = ROWS
   PVARUSE = VARUSE
return

 /*-------------------------------------------------------------------*/
 /* Retrieve index statistics from catalog for entered table */
 /*-------------------------------------------------------------------*/
D_INDEX:
 /*-------------------------------------------------------------  */
 /* Define ISPF Table to hold data for possible mutiple indexes */
 /*-------------------------------------------------------------------*/
  address ispexec "TBCREATE T£IDX",
```

```
       "NAMES(ICRTR INAME NFSKCRD FSKCRD FLKCRD NLEAF NLVLS IXLEN ",
       "OFLKCRD ONLEAF ONLVLS NCRIO OCRIO IUPD)",
       "NOWRITE REPLACE"
 ICRTR  = ""
 INAME  = ""
 NFSKCRD= ""
 FSKCRD = ""
 FLKCRD = ""
 NLEAF  = ""
 NLVLS  = ""
 OFLKCRD= ""
 ONLEAF = ""
 ONLVLS = ""
 IXLEN  = ""
 CLRNG  = ""
 NCRIO  = ""
 OCRIO  = ""
 IUPD   = ""
/*---------------------------------------- */
/* Connect to DB2 set DB2 connect flag */
/*---------------------------------------- */
 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
/*------------------------------------------------------------------------ */
/* Set up SQL statement to retrieve index data from sysindexes,    */
/* syskeys, and syscolumns. Declare and open cursor then fetch rows*/
/*------------------------------------------------------------------------ */
 SLCT =  "SELECT A.TBCREATOR,A.TBNAME,A.CREATOR,A.NAME",
         ",A.FIRSTKEYCARD,SUM(C.LENGTH),A.FULLKEYCARD",
         ",A.NLEAF,A.NLEVELS,A.CLUSTERING,A.CLUSTERRATIO",
         "FROM SYSIBM.SYSINDEXES A ,SYSIBM.SYSKEYS B",
         ",SYSIBM.SYSCOLUMNS C"
 WHRE =  "WHERE A.TBCREATOR='"CRTR"' AND A.TBNAME='"TBNAM"'",
         "AND B.IXCREATOR=A.CREATOR AND B.IXNAME=A.NAME ",
         "AND C.TBCREATOR=A.TBCREATOR AND C.TBNAME=A.TBNAME",
         "AND C.NAME=B.COLNAME"
 GRBY =  "GROUP BY A.TBCREATOR,A.TBNAME,A.CREATOR,A.NAME",
         ",A.FIRSTKEYCARD,A.FULLKEYCARD",
         ",A.NLEAF,A.NLEVELS,A.CLUSTERING,A.CLUSTERRATIO"
 address db2 "DECLARE RXCSR2 CURSOR FOR",
             SLCT,
             WHRE,
             GRBY
 if RC ¬= 0 then
    MESS = 'Error declaring cursor - RC ' RC
```

```
 else do
    address db2 "OPEN RXCSR2"
    do A = 1 by 1 until SQLCODE ¬= Ø
       address db2 "FETCH RXCSR2"
       if SQLCODE ¬= Ø then iterate A
/*———————————————————————— */
/* Populate ISPF table T£IDX */
/*———————————————————————— */
       ICRTR  = RXCSR2.3
       INAME  = RXCSR2.4
       NFSKCRD= RXCSR2.5
       FSKCRD = RXCSR2.5
       IXLEN  = RXCSR2.6
       OFLKCRD= RXCSR2.7
       ONLEAF = RXCSR2.8
       ONLVLS = RXCSR2.9
       CLING  = RXCSR2.1Ø
       OCRIO  = RXCSR2.11
       NCRIO  = RXCSR2.11
       if CLING  = 'Y' then NCRIO = 1ØØ
       call DA_IXSTATS
       IUPD = '*'
       if OFLKCRD = FLKCRD then
          if ONLEAF = NLEAF then
             if ONLVLS = NLVLS then IUPD = ' '
       address ispexec "TBADD T£IDX"
    end
    address db2 "CLOSE RXCSR2"
 end
/*———————————————————————————————————————————————*/
/* Disconnect from DB2 - reset DB2 connect flag */
/*———————————————————————————————————————————————*/
 address db2 "SIGNOFF"
 DB2CON = Ø
/*———————————————————————————————————————————————*/
/* Process CSIDX displaying ISPF table T£IDX  */
/*———————————————————————————————————————————————*/
 ZCMD = ' '
 address ispexec "TBTOP T£IDX"
 address ispexec "TBDISPL T£IDX PANEL(CSIDX) AUTOSEL(NO)"
/*———————————————————————————————————*/
/* Process index statistics panel */
/*———————————————————————————————————*/
 do forever
    MESS = 'No updates performed for Table/Index statistics'
    if rc > 4 then leave
    K = ZTDSELS + Ø
    do until K < 1
/*———————————————————————————————————————————————*/
/* Update ISPF table T£IDX with any alterations  */
/*———————————————————————————————————————————————*/
       if K > Ø then do I = 1 by 1 to K
```

```
            if NCRIO = '' | NCRIO = ' ' then NCRIO = OCRIO
            if NFSKCRD = '' | NFSKCRD = ' ' then NFSKCRD = FSKCRD
            address ispexec "TBPUT T£IDX"
            address ispexec "TBDISPL T£IDX"
         END /* I LOOP */
         K = ZTDSELS + Ø
      end  /* k loop */
      if rc > 4 then leave
      if ZCMD = exit then exit
      if ZCMD = end  then leave
      if ZCMD = 'COL' then leave
      if ZCMD = 'UPD' then do
         ZCMD = ''
         call DB_UPDATE
         leave
      end
      MESS = ' '
      address ispexec "TBDISPL T£IDX"
   end
return

 /*———————————————————————————*/
 /* Calculate Index statistics */
 /*———————————————————————————*/
DA_IXSTATS:
   FLKCRD = ROWS
 /*———————————————————————————————————*/
 /* Calculate number of leaf pages */
 /*———————————————————————————————————*/
   SIZE = IXLEN + 4
   IXP = 4Ø5Ø - SIZE
   IXP = trunc(IXP)
   IXPS = ( IXP / SIZE )
   IXPS = trunc(IXPS)
   NLEAF = ( ROWS / IXPS ) + Ø.9999
   NLEAF = trunc(NLEAF)
 /*———————————————————————————————————*/
 /* Calculate number of index levels */
 /*———————————————————————————————————*/
   NLVLS = 1
   N = NLEAF
   if N > 1 then do until N = 1
      PCU = 1ØØ
      if NLVLS > 1 then PCU = 9Ø
      IXPS = 4Ø5Ø * PCU / 1ØØ
      IXPS = trunc(IXPS)
      N = ( N * SIZE / IXPS ) + Ø.999
      N = trunc(N)
      NLVLS = NLVLS + 1
   end
return
```

```
DB_UPDATE:
  signal on failure
 /*-------------------------------------------- */
 /* Update Table data - if data has changed */
 /*-------------------------------------------- */
  if ROWS <> OCARD | NPAGES <> ONPGES | PCTPAGES <> OPCPAGES then do
     if DB2CON = 0 then do
        address db2 "SIGNON" DB2S
        if RC > 0 then do
           MESS = "Error - Unable to connect to " DB2S
           RETURN16
        end
        DB2CON = 1
     end
     UPDT = "UPDATE SYSIBM.SYSTABLES",
            "SET CARD="ROWS",NPAGES="NPAGES",",
            "PCTPAGES="PCTPAGES
     WHRCLS = "WHERE CREATOR='"CRTR"' AND NAME='"TBNAM"'"
     address db2 UPDT,
                 WHRCLS
     if SQLCODE <> 0 then DO
        MESS = 'Error on table update rc ' RC ' sqlcode ' SQLCODE
        return
     end
  end
 /*---------------------------*/
 /* Update Tablespace data */
 /*---------------------------*/
  if NACTIVE <> OACTIVE then do
     if DB2CON = 0 then do
        address db2 "SIGNON" DB2S
        if RC > 0 then do
           MESS = "Error - Unable to connect to " DB2S
           RETURN16
        end
        DB2CON = 1
     end
     UPDT = "UPDATE SYSIBM.SYSTABLESPACE ",
            "SET NACTIVE="NACTIVE
     WHRCLS = "WHERE DBNAME='"DBNAM"' AND NAME='"TSPACE"'"
     address db2 UPDT,
                 WHRCLS
     if SQLCODE <> 0 then DO
        MESS = 'Error on tablespace update rc ' RC ' sqlcode ' SQLCODE
        return
     end
  end
```

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

Liz Page
Independent Consultant (UK)

Extracting from LISTCAT output

In my current shop, the DBA's regular work includes monitoring the size of the database and the space used by the database objects for the purpose of performance management and capacity planning.

The volume information will be changed every month when the reorg is done. This is important and is kept handy so that, if an emergency such as a DASD head crash occurs, the DBA immediately knows which database objects are on the bad DASD and can prepare for immediate recovery. The volume information can also be used during the monthly maintenance work after the reorg job to see whether any indexes were allocated on the same volume as the tablespace that they refer to, and, if so, can be spread to different volumes to avoid DASD contention.

Another step in the monthly maintenance is to reset multiple extents of a dataset. In our case, it is reset when it goes over six extents in the same volume or it extends to multiple volumes.

IBM's LISTCAT utility provides rich information on datasets; however, there is too much information for easy readibility.

The following REXX program extracts the essential data from the large LISTCAT output and shows the following information in a simpler format:

- The tablespace or index name and its VCAT name.
- The number of partitions (one if non-partitioned).
- The volume serial number of the DASD where the dataset resides.
- The number of extents of the dataset.
- The space allocated in tracks.
- The space allocation parameter of the dataset (primary, secondary).
- The space in use (by percentage).
- The total space allocated for the database and its percentage utilization.

```
                         Volume listing
                         as of
                         13 Jan 1999

VCAT       TS/IX      PART  VOLSER   EXT   TRKS    ALLOC PARM        %USE
———        ———        ——    ———      —     ——      —————             ——
AASPUZ     UZSTØØ1      1   DBL139    1     918Ø   CYL(612,123)        8Ø
AASPUZ     UZSTØØ2      1   DBXØ24    1     834Ø   CYL(556,56)         91
AASPUZ     UZSTØØ3      1   AVSØ79    1      945   CYL(63,7)           83
AASPUZ     UZSTØØ4      1   BVSØ54    1        4   TRK(4,1)            54
AASPUZ     UZSTØØ5      1   BVSØ54    1        9   TRK(9,3)            62
  .           .         .     .       .        .       .                .
  .           .         .     .       .        .       .                .
AASPUZ     UZX1ACT      1   AVSØ61    1     1875   CYL(125,13)         85
AASPUZ     UZX1ACTF     1   AVSØ62    1        4   TRK(4,1)            56
AASPUZ     UZX1ACTM     1   BVSØ4Ø    2      39Ø   CYL(17,9)           96
AASPUZ     UZX1ACTT     1   BVSØ46    1        1   TRK(1,1)           1ØØ
AASPUZ     UZX1ADDR     1   BVSØ4Ø    1        1   TRK(1,1)           1ØØ
  .           .         .     .       .        .       .                .
  .           .         .     .       .        .       .                .
AASPUZ     UZX3CNTR     1   AVSØ76    1      255   CYL(17,2)           88

            TOTAL 339Ø SPACE =  914Ø CYLS  5 TRK   7.8 GB    82.2 % USED.
```

Figure 1: Example of Report 1

This gives Report 1 – an example is shown in Figure 1.

Report 2 (see Figure 2) strips more information and sorts the output by the VOLSER where the objects reside. It shows the VCAT name, the number of extents, the space allocated in tracks, and the percentage of the space in use. As mentioned, the DBA checks this report after he runs reorg on his database to see whether there are any indexes located on the same volume as the tablespace they refer to.

The REXX program follows. Before running the program, two datasets need to be created and their names must be hardcoded in the ALLOC statement of the program:

- Your.hilevel.listcat: recfm=VBA, lrecl=124, blksize=28,329
- Your.hilevel.lcattemp: recfm=FB, lrecl=80, blksize=26,400


```
AASPUZ   UZSTØØ7  DBL137  1   6675  77
AASPUZ   UZSTØØ1  DBL139  1   9180  8Ø
AASPUZ   UZX2MRUL CBSØØ4  1      1 1ØØ
AASPUZ   UZSTØ69  CBSØØ7  1     1Ø 1ØØ
AASPUZ   UZX1TEXT CBS1Ø5  1    135  22
AASPUZ   UZSTØ55  CBS119  2      8  75
AASPUZ   UZX1STHS DBXØØ7  1   3345  96
AASPUZ   UZST ØØ2 DBXØ24  1   834Ø  91
AASPUZ   UZX1CMPL BVSØØ2  1      1 1ØØ
AASPUZ   UZX1CNTL BVSØØ2  1      1 1ØØ
  .         .       .     .      .   .
  .         .       .     .      .   .
AASPUZ   UZX1NOTE TVSØØ3  1     18  28
AASPUZ   UZX2NAME TVSØ27  1    345  78
```

Figure 2: Example of Report 2

SOURCE CODE

```
/* REXX                                                          */
/*****************************************************************/
/*  This program extracts DB2 tablespace and index dataset       */
/*  information from the IDCAMS LISTCAT output such as:          */
/*     - volume serial number where the dataset resides          */
/*     - number of partitions of the dataset                     */
/*     - number of extents of the dataset                        */
/*     - space allocation parameter of the dataset               */
/*     - total space used for the database                       */
/*****************************************************************/
/*  Update                                                       */
/*    - Correct the total space allocation. In a fragmented      */
/*      volume an extent of a multi-extents dataset can be       */
/*      smaller than the primary or secondary space amount.      */
/*    - Add the total utilization                                */
/*****************************************************************/
ARG listclvl
Address TSO
"Alloc ddname(listcdd) dsname('"your.hilevel.listcat"') shr reuse"
"Alloc ddname(outdd) dsname('"your.hilevel.lcattemp"') shr reuse"
"Listc lvl('"listclvl"') data alloc outfile(listcdd)"
"Execio * diskr listcdd (finis"
Numeric digits 7
number_of_lines = queued()

If number_of_lines <> Ø then do       /*     Print title        */
   Say ' '
```

```
   Say ' '
   Say '                                Volume listing'
   Say '                                    as of  '
   Say '                                ' DATE()
   Say ' '
   Say 'VCAT      ' 'TS/IX   ' 'PART  ' 'VOLSER ' 'EXT    ',
       'TRKS ' 'ALLOC PARM        ' '%USE'
   Say '____    ' '____  ' '__  ' '____ ' '_    ',
       '__ ' '______       ' '__'
   End
Else Exit                              /*    Nothing to process  */

/*****************************************************************/
/*          Initialize variables                                 */
/*****************************************************************/
vcat_name.          = ''
tspace_name.        = ''   /* Name of the tablespace or index   */
num_of_partitions.  = ''   /* For a partitioned tablespace      */
space_type.         = ''   /* CYLINDER or TRACK                 */
primary_space.      = ''   /* Primary allocation quantity       */
secondry_space.     = ''   /* Secondary allocation quantity     */
space_alloc_parm.   = ''   /* (CYL,(10,5)) for 10 cyl primary   */
                           /* and 5 cyl for secondary           */
volser.             = ''   /* Volume serial of the DASD         */
differnt_volumes.   = ''   /* # of volumes for multi-volume ds  */
num_of_extents.     = ''   /* Number of extents for the ds      */
hi_alloc_rba.       = ''   /* Hi-Alloc-RBA in LISTCAT           */
hi_used_rba.        = ''   /* Hi-Used-RBA in LISTCAT            */
outrec.             = ''   /* Output record image in the buffer */
used_tracks.        = 0    /* Tracks in use in a TS or index ds */
alloc_tracks.       = 0    /* Allocated trks of a tablespace or */
                           /* index in one volume               */
percent_used.       = 0    /* Space utilization of tablespace   */
                           /* or index dataset                  */
total_alloc_db_cyls = 0    /* Total alloc in cyl for the ds     */
total_alloc_remains = 0    /* and remaining in tracks           */
total_used_db_trks  = 0    /* Total tracks used in a database   */
total_alloc_db_trks = 0    /* Total allocated tracks for the db */
percent_total_usage = 0    /* Total space utilization of the db */
total_gbyte         = 0    /* Size of the database in GBytes    */
extcount            = 0    /* Counter                           */
i                   = 0    /* Counter                           */
j                   = 0    /* Counter                           */
k                   = 0    /* Counter                           */
m                   = 0    /* Counter                           */

/*****************************************************************/
/*            Process the LISTCAT output                         */
/*****************************************************************/
Do number_of_lines
```

```
Pull 2 word1 word2 word3 word4

If word1 = 'DATA' then do
     differnt_volumes.i = volnum
     i = i + 1
     Parse var word3 part1 '.' rpart2
     vcat_name.i = part1

     Parse var rpart2 part2 '.' part3 '.' part4 '.' part5 '.' part6
     tspace_name.i = part4
     partno = substr(part6,2,3)
     num_of_partitions.i = strip(partno,l,'Ø')
     volnum = Ø
     End
If substr(word1,1,1Ø) = 'SPACE-TYPE' then do
     Parse var word1 part1 '-' part2 '-' part3
     space_type.i = strip(part3,l,'-')
     If space_type.i = 'CYLINDER' then
        space_type.i = 'CYL'
     Else
        space_type.i = 'TRK'
     Parse var word2 part1 '-' part2 '-' part3 '-' part4
     hi_alloc_rba.i = strip(part4,l,'-')
     End
If substr(word1,1,9) = 'SPACE-PRI' then do
     Parse var word1 part1 '-' part2 '-' part3
     primary_space.i = strip(part3,l,'-')
     If space_type.i = 'CYL' then
        total_alloc_db_cyls = total_alloc_db_cyls + primary_space.i
     else
        total_alloc_remains = total_alloc_remains + primary_space.i
     Parse var word2 part1 '-' part2 '-' part3 '-' part4
     hi_used_rba.i = strip(part4,l,'-')
     /*********************************************************/
     /*      percent use is calculated by the rate :          */
     /*            hi-u-rba / hi-a-rba                        */
     /*      hi-u-rba and hi-a-rba always start from Ø.       */
     /*********************************************************/
     percent_used.i = (hi_used_rba.i * 1ØØ) % hi_alloc_rba.i
     end
If substr(word1,1,9) = 'SPACE-SEC' then do
     parse VAR word1 part1 '-' part2 '-' part3
     secondry_space.i = strip(part3,l,'-')
     end
If substr(word1,1,6) = 'VOLSER' then do
     volnum = volnum + 1
     if volnum > 5 then goto next_rec
     parse var word1 part1 '-' part2
     volser.volnum.i = strip(part2,l,'-')
```

```
      word4 = strip(word4,l)
      word4 = strip(word4,t)
      Parse var word4 part1 '-' part2 '-' part3
      num_of_extents.volnum.i = strip(part3,l,'-')
      alloc_tracks.volnum.i = Ø
      extcount = num_of_extents.volnum.i
      Do while extcount > Ø
         Pull 2 word1 word2 word3 word4
         If substr(word1,1,8) = 'LOW-CCHH' then do
            Parse var word3 part1 '-' part2
            alloc_tracks.volnum.i = alloc_tracks.volnum.i ,
                                  + strip(part2,l,'-')
            extcount = extcount - 1
         End
      End
End
  next_rec :
End

j = 1
m = 1
Do i - 1
  vcat_name.j = left(vcat_name.j,8)
  tspace_name.j = left(tspace_name.j,8)
  num_of_partitions.j = right(num_of_partitions.j,2)
  num_of_extents.1.j = right(num_of_extents.1.j,2)
  alloc_tracks.1.j = right(alloc_tracks.1.j,6)
  space_alloc_parm.j = space_type.j ||'('|| primary_space.j|| ',' ||,
                       secondry_space.j || ')'
  space_alloc_parm.j = left(space_alloc_parm.j,13)
  percent_used.j = right(percent_used.j,3)
  Say vcat_name.j ' ' tspace_name.j ' ' num_of_partitions.j ' ',
      volser.1.j ' ' num_of_extents.1.j,
      ' ' alloc_tracks.1.j ' ' space_alloc_parm.j ' ' percent_used.j
  total_alloc_db_trks = total_alloc_db_trks + alloc_tracks.1.j
  used_tracks.1.j = alloc_tracks.1.j * ( percent_used.j / 1ØØ )
  total_used_db_trks = total_used_db_trks + used_tracks.1.j
  outrec.m = vcat_name.j tspace_name.j volser.1.j num_of_extents.1.j,
             alloc_tracks.1.j percent_used.j
  If differnt_volumes.j > 1 then do
    k = 1
    Do differnt_volumes.j - 1
       k = k + 1
       alloc_tracks.k.j = RIGHT(alloc_tracks.k.j,6)
       num_of_extents.k.j = RIGHT(num_of_extents.k.j,2)
       Say '                          ' volser.k.j,
       ' ' num_of_extents.k.j ' ' alloc_tracks.k.j
       total_alloc_db_trks = total_alloc_db_trks + alloc_tracks.k.j
       used_tracks.k.j = alloc_tracks.1.j * ( percent_used.j / 1ØØ )
       total_used_db_trks = total_used_db_trks + used_tracks.k.j
       m = m + 1
```

```
         outrec.m = vcat_name.j tspace_name.j volser.k.j,
                    num_of_extents.k.j  alloc_tracks.k.j
      End
   End
   j = j + 1
   m = m + 1
End

/*********************************************************************/
/*     Write the total allocation and utilization information      */
/*********************************************************************/
/*     Hardcoded the values for 3390 type DASD device:             */
/*         track capacity = 56,664 bytes / track                   */
/*         tracks per cylinder = 15 / cyl                          */
/*     Update the values for use in different device type.         */
/*********************************************************************/
total_alloc_db_cyls = trunc(total_alloc_db_trks / 15)
total_alloc_remains = total_alloc_db_trks - total_alloc_db_cyls * 15
total_gbyte = total_alloc_db_trks * 56.664 / 1000000
total_gbyte = format(total_gbyte,6,1)
percent_total_usage = 100 * total_used_db_trks / total_alloc_db_trks
percent_total_usage = format(percent_total_usage,4,1)
Say ' '
Say '            Total 3390 space = ' total_alloc_db_cyls 'CYLS',
    total_alloc_remains 'TRK  ' total_gbyte 'GB ',
    percent_total_usage '% used.'

/*********************************************************************/
/*      Wrap-up                                                      */
/*********************************************************************/
Address TSO
"Execio * diskw outdd (stem outrec."
"Execio 0 diskw outdd (finis"
"Free fi(listcdd)"
"Free fi(outdd)"

Exit
```

JCL

The following JCL will execute the REXX program from a PDS member LCATEXEC. Note the EXEC statement in SYSTSIN that has a parameter – it gives the high-level qualifiers of the database datasets.

The first step, SUMMARY, will create REPORT 1 in a SYSOUT. The DSN in OUTDD of the step will be passed to the next step to be sorted and it will be written to a dataset in SORTOUT. This will give REPORT 2.

```
//UZLISTC  JOB (A,A,SPUZSP,R),'CAM2,DBA',MSGCLASS=V,
//  NOTIFY=....,GROUP=.....,USER=....    <==== your job card
/*JOBPARM ROOM=....
//SUMMARY  EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD DUMMY
//OUTDD    DD DISP=OLD,DSN=your.listcat  <==== same in next step
//SYSTSIN DD *
  EXEC 'my.rexx.library(LCATEXEC)' 'AASPUZ.DSNDBD.UZDP001'
/*
//SORT  EXEC PGM=SORT,REGION=0K
//SORTIN   DD   DISP=SHR,DSN=your.listcat <=== passed from 1st step
//SORTOUT  DD   DISP=SHR,DSN=your.lcattemp  <=== REPORT 2
//*
//SORTWK01 DD   DISP=(NEW,DELETE),UNIT=DASD,SPACE=(TRK,(200,200))
//SORTWK02 DD   DISP=(NEW,DELETE),UNIT=DASD,SPACE=(TRK,(200,200))
//SORTWK03 DD   DISP=(NEW,DELETE),UNIT=DASD,SPACE=(TRK,(200,200))
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSIN    DD   *
  SORT FIELDS=(19,6,A,10,8,A),FORMAT=CH,EQUALS
/*
```

Samuel Park
DBA (USA)

Verifying start-up parameters

The following program reads the values of ZPARM in the DB2 control blocks, converts the parameter values to report format, and writes to a report file.

You can find the map in the installation's DB2 macro library (DSNMACS) in members DSN6ARVP, DSN6LOGP, DSN6SYSP, DSN6FAC, DSN6GRP, and DSN6SPRM, and DB2 sample library (DSNSAMP) in the member DSNWCBDS.

The program finds the value of ZPARM that was used at DB2 subsystem start time and accesses the ZPARM block via DB2 control block linkages.

This program was developed and tested in a DB2 Version 4.1 environment.

PROGRAM SOURCE

```
      TITLE 'PROGRAM READS VALUES OF ZPARM IN DB2 CONTROL BLOCK'
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
* FUNCTION : READS VALUES OF ZPARM FROM DB2 CONTROL BLOCK              *
*            DYNAMIC CHANGE INITIAL PARAMETER                         *
*                                                                     *
* REFERENCE: DB2 DIAGNOSIS GUIDE AND REFERENCE  CHAPTER 8-2           *
*            'DB2SYS.DSN41Ø.SDSNSAMP(DSNWCBDS)'                       *
*            'DB2SYS.DSN41Ø.SDSNMACS(DSN6*)'                          *
*                                                                     *
* @@V1.Ø : DB2 V4.1                                                   *
*                                                                     *
*                                                                     *
* NOTE  : STEPLIB MUST BE APF LIBRARY BECAUSE OF 'MODESET' INSTRUCTION*
*         RUN ONLY IN BATCH.                                          *
*                                                                     *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
         SPACE
DB2PARMS CSECT
DB2PARMS AMODE 31
DB2PARMS RMODE 24
*
         SPACE
         STM   R14,R12,12(R13)         SAVE REGISTERS
         BALR  R12,Ø                   NEW BASE ADDRESSABILITY
         USING *,R12,R11               GET ADDRESSABILITY
         LA    R11,1
         LA    R11,4Ø95(R11,R12)
         LA    R1Ø,1
         LA    R1Ø,4Ø95(R1Ø,R11)
         LA    R3,SAVEAREA             GET PGM SAVE AREA ADDRESS
         ST    R3,8(R13)               FORWARD CHAIN
         ST    R13,4(R3)               BACKWARD CHAIN
         LR    R13,R3                  SET PGM SAVE AREA POINTER
         EJECT
*
PARMRTN  DS    ØH
         LR    R7,R1                   LOAD PARM ADDRESS
         L     R7,Ø(R7)                POINT PARM LIST
         LH    R8,Ø(R7)                LOAD PARM LIST LENGTH
         BCTR  R8,Ø                    LEN - 1 FOR EX
         EX    R8,MOVEPARM             MOVE PARAMETER (DB2 SSN)
*
PARMEXT  DS    ØH
START    DS    ØH
         BAL   R14,OPENRTN             PERFORM OPENRTN
         BAL   R14,MAINRTN             PERFORM MAINRTN
         BAL   R14,CLOSERTN            PERFORM CLOSERTN
*
MOVEPARM MVC   PARAM(Ø),2(R7)          LOAD DB2 NAME PARM
```

```
*
OPENRTN  DS    ØH
         ST    R14,OPENSAVE               SAVE RETURN ADDR
         OPEN  (ZPARMDD,OUTPUT),MODE=31   OPEN DATASET OUTPUT
*
OPENEXT  DS    ØH
         L     R14,OPENSAVE               SET RETURN ADDR
         BR    R14                        RETURN NEXT INST ADDR
*
OPENSAVE DS    F'Ø'
         EJECT
*
CLOSERTN DS    ØH
*
         L     R13,SAVEAREA+4          RESTORE SAVE AREA
         XR    R15,R15                 CLEAR R15
         L     R15,RETCODE             LOAD RETURN CODE VALUE
         RETURN (14,12),,RC=(15)       EXIT PGM
*
CLOSEXT  DS    ØH
         EJECT
*
MAINRTN  DS    ØH
         ST    R14,MAINSAVE            SAVE RETURN ADDR
         MODESET KEY=ZERO,MODE=SUP     IN SUPERVISOR STATE
         BAL   R14,LINKRTN             PERFORM LINKAGE RTN
         BAL   R14,RPTRTN              PERFORM REPORT RTN
*
MAINEXT  DS    ØH
         L     R14,MAINSAVE            SET RETURN ADDR
         BR    R14                     RETURN NEXT INST ADDR
*
MAINSAVE DS    F'Ø'
         EJECT
*
LINKRTN  DS    ØH
         ST    R14,LINKSAVE            SAVE RETURN ADDR
         L     R5,16                   CVT POINTER
         L     R5,296(R5)              JESCT POINTER
         L     R5,24(R5)               SSCVT POINTER
*
DB2SSCT  DS    ØH
         CLC   8(4,R5),PARAM           CHECK SS WITH DB2 NAME
         BE    ERLYØØØØ                ... IF EQUAL, OK
         L     R5,4(R5)                NEXT SSCVT POINTER
         LTR   R5,R5                   CHECK COHERENCY POINTER
         BNZ   DB2SSCT                 IF GOOD, LOAD NEW SSCVT
         MVC   RETCODE,=F'8'           MOVE 8 IN RETURN CODE
         B     CLOSERTN                EXIT WITH ERROR RCODE
*
```

```
ERLY0000 DS    0H
         L     R5,20(R5)              ERLY POINTER
         L     R5,56(R5)              SCOM POINTER
         L     R5,144(R5)             ACOM POINTER
         L     R5,048(R5)             ZPARM POINTER
*
LINKEXT  DS    0H
         L     R14,LINKSAVE           SET RETURN ADDR
         BR    R14                    RETURN NEXT INST ADDR
*
LINKSAVE DS    F'0'
         EJECT
*
RPTRTN   DS    0H
         ST    R14,RPTSAVE            SAVE RETURN ADDR
         BAL   R14,ARVPRTN            PERFORM ARVP REPORT ROUTINE
         BAL   R14,LOGPRTN            PERFORM LOGP REPORT ROUTINE
         BAL   R14,SYSPRTN            PERFORM SYSP REPORT ROUTINE
         BAL   R14,GRPRTN             PERFORM GRP  REPORT ROUTINE
         BAL   R14,FACRTN             PERFORM FAC  REPORT ROUTINE
         BAL   R14,SPRMRTN            PERFORM SPRM REPORT ROUTINE
         MVC   RETCODE,=F'0'          MOVE 0 IN RETURN CODE
*
RPTEXT   DS    0H
         L     R14,RPTSAVE            SET RETURN ADDR
         BR    R14                    RETUEN NEXT INST ADDR
*
RPTSAVE  DS    F'0'
         EJECT
*
ARVPRTN  DS    0H
         ST    R14,ARVPSAVE           SAVE RETURN ADDR
         L     R6,032(R5)             ARVP POINTER
         USING DSN6ARVP,R6            MAPPING DSECT
         TM    ARVPFLG1,X'80'         CATALOG IS 'YES'
         BNO   ARVP0000               IF NO, GO TO ARVP0000
         MVC   PCATALOG,=C'YES'       MOVE 'YES' TO CATALOG-RPT AREA
         B     ARVP0005
*
ARVP0000 DS    0H
         MVC   PCATALOG,=C' NO'       MOVE 'NO' TO CATALOG-RPT AREA
*
ARVP0005 DS    0H
         TM    ARVPFLG1,X'40'         ALCUINT IS 'CYL'
         BNO   ARVP0010               IF NO, GO TO ARVP0010
         MVC   PALCUNIT,=C'CYL'       MOVE 'CYL' TO ALCUNIT-RPT AREA
         B     ARVP0030
*
ARVP0010 DS    0H
         TM    ARVPFLG1,X'20'         ALCUINT IS 'TRK'
         BNO   ARVP0020               IF NO, GO TO ARVP0020
```

```
         MVC   PALCUNIT,=C'TRK'       MOVE 'TRK' TO ALCUNIT-RPT AREA
         B     ARVP0030
*
ARVP0020 DS    0H
         MVC   PALCUNIT,=C'BLK'       MOVE 'BLK' TO ALCUNIT-RPT AREA
*                                     (DEFAULT VALUS SETTING)
ARVP0030 DS    0H
         TM    ARVPFLG1,X'10'         PROTECT IS 'YES'
         BNO   ARVP0040               IF NO, GO TO ARVP0040
         MVC   PPROTECT,=C'YES'       MOVE 'YES' TO PROTECT-RPT AREA
         B     ARVP0045
*
ARVP0040 DS    0H
         MVC   PPROTECT,=C' NO'       MOVE 'NO' TO PROTECT-RPT AREA
*
ARVP0045 DS    0H
         TM    ARVPFLG1,X'08'         ARCWTOR IS 'YES'
         BNO   ARVP0050               IF NO, GO TO ARVP0050
         MVC   PARCWTOR,=C'YES'       MOVE 'YES' TO ARCWTOR-RPT AREA
         B     ARVP0055
*
ARVP0050 DS    0H
         MVC   PARCWTOR,=C' NO'       MOVE 'NO' TO ARCWTOR-RPT AREA
*
ARVP0055 DS    0H
         TM    ARVPFLG1,X'04'         COMPACT IS 'YES'
         BNO   ARVP0060               IF NO, GO TO ARVP0060
         MVC   PCOMPACT,=C'YES'       MOVE 'YES' TO COMPACT-RPT AREA
         B     ARVP0065
*
ARVP0060 DS    0H
         MVC   PCOMPACT,=C' NO'       MOVE 'NO' TO COMPACT-RPT AREA
*
ARVP0065 DS    0H
         TM    ARVPFLG1,X'02'         TSTAMP  IS 'YES'
         BNO   ARVP0070               IF NO, GO TO ARVP0070
         MVC   PTSTAMP,=C'YES'        MOVE 'YES' TO TSTAMP-RPT AREA
         B     ARVP0075
*
ARVP0070 DS    0H
         MVC   PTSTAMP,=C' NO'        MOVE 'NO' TO TSTAMP-RPT AREA
*
ARVP0075 DS    0H
         MVC   PARCPFX1,ARVPRE1N      MOVE ARCH PREFIX1
         MVC   PARCPFX2,ARVPRE2N      MOVE ARCH PREFIX2
         MVC   PUNIT,ARVPUNT1         MOVE UNIT
         MVC   PUNIT2,ARVPUNT2        MOVE UNIT2
         LH    R7,ARVPRETN            RETENTION PERIOD
         CVD   R7,PACKWK01            CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
```

```
         ED    EDWORK10,PACKWK01+4
         MVC   PARCRETN,EDWORK10
         L     R7,ARVPBKSZ             BLOCKSIZE
         CVD   R7,PACKWK01             CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PBLKSIZE,EDWORK10
         L     R7,ARVPRISP             PRIMARY QUANTITY
         CVD   R7,PACKWK01             CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PPRIQTY,EDWORK10
         L     R7,ARVPSECS             SECONDARY QUANTITY
         CVD   R7,PACKWK01             CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PSECQTY,EDWORK10
         LH    R7,ARVPMQP              QUIESCE PERIOD
         CVD   R7,PACKWK01             CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PQUIESCE,EDWORK10
         LA    R8,PRTARVP              SET PRTARVP ADDR
         MVC   LOOPCNT,=X'000A'        SET LOOPCNT
         LH    R9,LOOPCNT
         BAL   R14,WRTRTN
*
ARVPEXT  DS    0H
         L     R14,ARVPSAVE            SET RETURN ADDR
         BR    R14                     RETURN NEXT INST ADDR
*
ARVPSAVE DS    F'0'
         EJECT
*
LOGPRTN  DS    0H
         ST    R14,LOGPSAVE            SAVE RETURN ADDR
         L     R6,052(R5)              LOGP POINTER
         USING DSN6LOGP,R6             MAPPING DSECT
         TM    LOGOPT1,X'80'           TWOACTV IS 'YES'
         BNO   LOGP0000                IF NO, GO TO LOGP0000
         MVC   PTWOACTV,=C'YES'        MOVE 'YES' TO TWOACTV-RPT AREA
         B     LOGP0005
*
LOGP0000 DS    0H
         MVC   PTWOACTV,=C' NO'        MOVE 'NO' TO TWOACTV-RPT AREA
*
LOGP0005 DS    0H
         TM    LOGOPT2,X'80'           TWOARCH IS 'YES'
         BNO   LOGP0010                IF NO, GO TO LOGP0010
         MVC   PTWOARCH,=C'YES'        MOVE 'YES' TO TWOARCH-RPT AREA
```

```
         B     LOGPØØ15
*
LOGPØØ1Ø DS    ØH
         MVC   PTWOARCH,=C' NO'       MOVE 'NO' TO TWOARCH-RPT AREA
*
LOGPØØ15 DS    ØH
         LH    R7,LOGPDMIN            DEALLOCATION MINUTE
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PDEALLC1,EDWORK1Ø
         LH    R7,LOGPDSEC            DEALLOCATION SECOND
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PDEALLC2,EDWORK1Ø+7
         L     R7,LOGPIBPS            INPUT BUFFER POOL SIZE
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PINBUFF,EDWORK1Ø
         L     R7,LOGPOBPS            OUT BUFFER POOL SIZE
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   POUTBUFF,EDWORK1Ø
         LH    R7,LOGPWRTH            WRITE THRESHOLD
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PWRTHRSH,EDWORK1Ø
         L     R7,LOGPARCL            MAX ARCHIVE ENTRIES IN BSDS
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PMAXARCH,EDWORK1Ø
         LH    R7,LOGPWRTH            OUTPUT BUFFER THRESHOLD
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PWRTHRSH,EDWORK1Ø
         LH    R7,LOGPMRTU            MAX ARCHIVE READ TAPE UNIT
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PMAXRTU,EDWORK1Ø
         LA    R8,PRTLOGP             SET PRTLOGP ADDR
         MVC   LOOPCNT,=X'ØØØ5'       SET LOOPCNT
         LH    R9,LOOPCNT
         BAL   R14,WRTRTN
```

```
*
LOGPEXT  DS    ØH
         L     R14,LOGPSAVE           SET RETURN ADDR
         BR    R14                    RETURN NEXT INST ADDR
*
LOGPSAVE DS    F'Ø'
         EJECT
*
SYSPRTN  DS    ØH
         ST    R14,SYSPSAVE           SAVE RETURN ADDR
         L     R6,Ø72(R5)             SYSP POINTER
         USING DSN6SYSP,R6            MAPPING DSECT
         LH    R7,SYSPCT              CTHREAD
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PCTHREAD,EDWORK1Ø
         LH    R7,SYSPIDF             IDFORE
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PIDFORE,EDWORK1Ø
         LH    R7,SYSPIDB             IDBACK
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PIDBACK,EDWORK1Ø
         LH    R7,SYSPCDB             CONDBAT
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PCONDBAT,EDWORK1Ø
         LH    R7,SYSPRMT             MAXDBAT
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PMAXDBAT,EDWORK1Ø
         LH    R7,SYSPDFRQ            DLDFREQ
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PDLDFREQ,EDWORK1Ø
         L     R7,SYSPLOGL            LOGLOAD
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PLOGLOAD,EDWORK1Ø
         TM    SYSPAUDT,X'8Ø'         AUDITST IS 'YES'
         BNO   SYSPØØØØ               IF NO, GO TO SYSPØØØØ
         MVC   PAUDITST,=C'YES'       MOVE 'YES' TO AUDITST-RPT AREA
```

```
         B     SYSP0005
*
SYSP0000 DS    0H
         MVC   PAUDITST,=C' NO'       MOVE 'NO' TO AUDITST-RPT AREA
*
SYSP0005 DS    0H
         TM    SYSPSMFA,X'80'         SMFACCT IS 'YES'
         BNO   SYSP0010               IF NO, GO TO SYSP0010
         MVC   PSMFACCT,=C'YES'       MOVE 'YES' TO SMFACCT-RPT AREA
         B     SYSP0015
*
SYSP0010 DS    0H
         MVC   PSMFACCT,=C' NO'       MOVE 'NO' TO SMFACCT-RPT AREA
*
SYSP0015 DS    0H
         TM    SYSPSMFS,X'80'         SMFSTAT IS 'YES'
         BNO   SYSP0020               IF NO, GO TO SYSP0020
         MVC   PSMFSTAT,=C'YES'       MOVE 'YES' TO SMFSTAT-RPT AREA
         B     SYSP0025
*
SYSP0020 DS    0H
         MVC   PSMFSTAT,=C' NO'       MOVE 'NO' TO SMFSTAT-RPT AREA
*
SYSP0025 DS    0H
         LH    R7,SYSPSTIM            STATIME
         CVD   R7,PACKWK01            CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PSTATIME,EDWORK10
         TM    SYSPMON,X'80'          MON IS 'YES'
         BNO   SYSP0030               IF NO, GO TO SYSP0030
         MVC   PMON,=C'YES'           MOVE 'YES' TO MON-RPT AREA
         B     SYSP0035
*
SYSP0030 DS    0H
         MVC   PMON,=C' NO'           MOVE 'NO' TO MON-RPT AREA
*
SYSP0035 DS    0H
         L     R7,SYSPMONS            MONSIZE
         CVD   R7,PACKWK01            CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PMONSIZE,EDWORK10
         TM    SYSPTRST,X'80'         TRACSTR IS 'YES'
         BNO   SYSP0040               IF NO, GO TO SYSP0040
         MVC   PTRACSTR,=C'YES'       MOVE 'YES' TO TRACSTR-RPT AREA
         B     SYSP0045
*
SYSP0040 DS    0H
         MVC   PTRACSTR,=C' NO'       MOVE 'NO' TO TRACSTR-RPT AREA
```

```
*
SYSPØØ45 DS    ØH
         LH    R7,SYSPTRSZ            TRACTBL
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PTRACTBL,EDWORK1Ø
         TM    SYSPRLFR,X'8Ø'          RLF IS 'YES'
         BNO   SYSPØØ5Ø                IF NO, GO TO SYSPØØ5Ø
         MVC   PRLF,=C'YES'            MOVE 'YES' TO RLF-RPT AREA
         B     SYSPØØ55
*
SYSPØØ5Ø DS    ØH
         MVC   PRLF,=C' NO'            MOVE 'NO' TO RLF-RPT AREA
*
SYSPØØ55 DS    ØH
         MVC   PRLFTBL+1(2),SYSPRLFT   RLFTBL
         MVC   PRLFAUTH+1(6),SYSPRLFA RLFAUTH(SYSIBM)
         TM    SYSPRLFR,X'4Ø'          RLFERR IS 'NOLIMIT '
         BNO   SYSPØØ6Ø                IF NO, GO TO SYSPØØ6Ø
         MVC   PRLFERR,=C'NOLIMIT'     MOVE 'NOLIMIT' TO RLFERR-RPT AREA
         B     SYSPØØ65
*
SYSPØØ6Ø DS    ØH
         TM    SYSPRLFR,X'2Ø'          RLFERR IS 'NOLIMIT '
         BNO   SYSPØØ65                IF NO, GO TO SYSPØØ6Ø
         MVC   PRLFERR,=C'  NORUN'     MOVE 'NOLIMIT' TO RLFERR-RPT AREA
*
SYSPØØ65 DS    ØH
         L     R7,SYSPRLFR+1           RLFERR = 'INGEGER'
         CVD   R7,PACKWKØ1             CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PRLFERR,EDWORK1Ø+3
         MVC   PSTORPRO,SYSPSPPN       STORED PROCEDURE NAME
         LH    R7,SYSPSPAB             STORMAXB
         CVD   R7,PACKWKØ1             CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PSTORMAX,EDWORK1Ø
         LH    R7,SYSPSPTO             STORTIME
         CVD   R7,PACKWKØ1             CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PSTORTIM,EDWORK1Ø+4
         LA    R8,PRTSYSP              SET PRTSYSP ADDR
         MVC   LOOPCNT,=X'ØØØD'        SET LOOPCNT
         LH    R9,LOOPCNT
         BAL   R14,WRTRTN
*
SYSPEXT  DS    ØH
```

```
         L     R14,SYSPSAVE           SET RETURN ADDR
         BR    R14                    RETURN NEXT INST ADDR
*
SYSPSAVE DS    F'Ø'
         EJECT
*
GRPRTN   DS    ØH
         ST    R14,GRPSAVE            SAVE RETURN ADDR
         L     R6,Ø92(R5)             GRP POINTER
         USING DSN6GRP,R6             MAPPING DSECT
         TM    GRPDSHR,X'8Ø'          DSHARE  IS 'YES'
         BNO   GRP0000                IF NO, GO TO GRPØØØØ
         MVC   PDSHARE,=C'YES'        MOVE 'YES' TO DSHARE-RPT AREA
         B     GRPØØ1Ø
*
GRPØØØØ  DS    ØH
         CLI   GRPDSHR,X'ØØ'          DSHARE  IS 'NO'
         BNE   GRPØØØ5                IF NO, GO TO GRPØØ1Ø
         MVC   PDSHARE,=C' NO'        MOVE 'NO' TO DSHARE-RPT AREA
         B     GRPØØ1Ø
*
GRPØØØ5  DS    ØH
         MVC   PDSHARE,=C'ERR'        MOVE 'ERR' TO DSHARE-RPT AREA
*
GRPØØ1Ø  DS    ØH
         MVC   PGRPNAME,GRPNAME       MOVE GRPNAME TO GRPNAME-RPT AREA
         MVC   PMEMBNAM,GRPMNAME      MOVE MEMBNAME TO MEMBNAME-RPT ARE
         LA    R8,PRTGRP              SET PRTGRP ADDR
         MVC   LOOPCNT,=X'ØØØ3'       SET LOOPCNT
         LH    R9,LOOPCNT
         BAL   R14,WRTRTN
*
GRPEXT   DS    ØH
         L     R14,GRPSAVE            SET RETURN ADDR
         BR    R14                    RETURN NEXT INST ADDR
*
GRPSAVE  DS     F'Ø'
         EJECT
*
FACRTN   DS    ØH
         ST    R14,FACSAVE            SAVE RETURN ADDR
         L     R6,112(R5)             FAC POINTER
         USING DSN6FAC,R6             MAPPING DSECT
         CLI   FACSTART,C'C'          DDF     IS 'COMMAND'
         BNE   FAC0000
         MVC   PDDF,=C'COMMAND'
         B     FACØØ1Ø
*
FACØØØØ  DS    ØH
         CLI   FACSTART,C'A'          DDF     IS 'AUTO'
         BNE   FACØØØ5
```

```
         MVC   PDDF,=C'   AUTO'
         B     FAC0010
*
FAC0005  DS    0H
         MVC   PDDF,=C'     NO'
*
FAC0010  DS    0H
         MVC   PCMTSTAT,FACCMST       CMTSTAT
         LH    R7,FACTOIN             IDTHTOIN
         CVD   R7,PACKWK01            CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PIDTHTOI,EDWORK10
         LH    R7,FACRESYC            RESYNC
         CVD   R7,PACKWK01            CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PRESYNC,EDWORK10
         TM    FACRLFER,X'80'         RLFERRD IS 'NOLIMIT'
         BNO   FAC0015                IF NO, GO TO FAC0015
         MVC   PRLFERRD,=C'NOLIMIT'   MOVE 'NOLIMIT'TO PRLFERRD-PRT ARE
         B     FAC0025
*
FAC0015  DS    0H
         TM    FACRLFER,X'40'         RLFERRD IS 'NORUN'
         BNO   FAC0020                IF NO, GO TO FAC0020
         MVC   PRLFERRD,=C'  NORUN'   MOVE 'NORUN'TO PRLFERRD-PRT AREA
         B     FAC0025
*
FAC0020  DS    0H
         MVC   PRLFERRD,=C'  LIMIT'   MOVE 'LIMIT'TO PRLFERRD-PRT AREA
*
FAC0025  DS    0H
*
         L     R7,FACRLFN             RLFERR EXEC TIME
         CVD   R7,PACKWK01            CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PRLFERRT,EDWORK10
         LA    R8,PRTFAC              SET PRTFAC ADDR
         MVC   LOOPCNT,=X'0004'       SET LOOPCNT
         LH    R9,LOOPCNT
         BAL   R14,WRTRTN
*
FACEXT   DS    0H
         L     R14,FACSAVE            SET RETURN ADDR
         BR    R14                    RETURN NEXT INST ADDR
*
FACSAVE  DS     F'0'
         EJECT
*
```

```
SPRMRTN  DS    ØH
         ST    R14,SPRMSAVE           SAVE RETURN ADDR
         L     R6,132(R5)             SPRM POINTER
         USING DSN6SPRM,R6            MAPPING DSECT
         TM    SPRMSTRT,X'8Ø'         RESTART AND ALL CHECK
         BNO   SPRM0ØØØ
         MVC   PRESTART+Ø(14),=C'  RESTART, ALL'
         B     SPRMØØØ5
*
SPRM0ØØØ DS    ØH
         TM    SPRMSTRT,X'4Ø'         RESTART AND ALL CHECK
         BNO   SPRMØØØ5
         MVC   PRESTART+Ø(25),=C'  RESTART, LIST    OR    '
         MVC   PRESTART+25(1Ø),=C'DEFER, ALL'
*
SPRMØØØ5 DS    ØH
         TM    SPRMMISZ,X'8Ø'         ABEXP CHECK
         BNO   SPRMØØ1Ø
         MVC   PABEXP,=C'YES'
         B     SPRMØØ15
*
SPRMØØ1Ø DS    ØH
         MVC   PABEXP,=C' NO'
*
SPRMØØ15 DS    ØH
         TM    SPRMMISZ,X'4Ø'         HOPAUTH  CHECK
         BNO   SPRMØØ2Ø
         MVC   PHOPAUTH,=C'YES'
         B     SPRMØØ25
*
SPRMØØ2Ø DS    ØH
         MVC   PHOPAUTH,=C' NO'
*
SPRMØØ25 DS    ØH
         TM    SPRMMISZ,X'1Ø'         SEQCACH CHECK
         BNO   SPRMØØ3Ø
         MVC   PSEQCACH,=C'   SEQ'
         B     SPRMØØ35
*
SPRMØØ3Ø DS    ØH
         MVC   PSEQCACH,=C'BYPASS'
*
SPRMØØ35 DS    ØH
         TM    SPRMMISZ,X'Ø4'         RRULOCK CHECK
         BNO   SPRMØØ4Ø
         MVC   PRRULOCK,=C'YES'
         B     SPRMØØ45
*
SPRMØØ4Ø DS    ØH
         MVC   PRRULOCK,=C' NO'
*
```

```
SPRM0045 DS    0H
         CLI   SPRMABN,C'E'           ABIND CHECK ('E' OR 'D')
         BNE   SPRM0050
         MVC   PABIND,=C'YES'
         B     SPRM0055
*
SPRM0050 DS    0H
         MVC   PABIND,=C' NO'
*
SPRM0055 DS    0H
         CLI   SPRMAUTH,C'E'          AUTH  CHECK ('E' OR 'D')
         BNE   SPRM0060
         MVC   PAUTH,=C'YES'
         B     SPRM0065
*
SPRM0060 DS    0H
         MVC   PAUTH,=C' NO'
*
SPRM0065 DS    0H
         LH    R7,SPRMAUCA            AUTHCACH
         CVD   R7,PACKWK01            CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PAUTHCAC,EDWORK10
         MVC   PBINDNV,SPRMBNVA       BINDNV
         LH    R7,SPRMBMP             BMPTOUT
         CVD   R7,PACKWK01            CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PBMPTOUT,EDWORK10
         MVC   PSPRMCAT+0(8),SPRMVCAT+12         CATALOG
         MVC   PCDSSRDE,SPRMCDEG      CDSSRDEF
         TM    SPRMMISC,X'40'         DECDIV3 CHECK
         BNO   SPRM0070
         MVC   PDECDIV3,=C'YES'
         B     SPRM0075
*
SPRM0070 DS    0H
         MVC   PDECDIV3,=C' NO'
*
SPRM0075 DS    0H
         TM    SPRMMISC,X'08'         CHGDC   CHECK
         BNO   SPRM0080
         MVC   PCHGDC,=C'YES'
         B     SPRM0085
*
SPRM0080 DS    0H
         MVC   PCHGDC,=C' NO'
*
SPRM0085 DS    0H
         TM    SPRMMISC,X'04'         EDPROP  CHECK
```

```
         BNO   SPRM0090
         MVC   PEDPROP,=C'YES'
         B     SPRM0095
*
SPRM0090 DS    0H
         MVC   PEDPROP,=C' NO'
*
SPRM0095 DS    0H
         CLI   SPRMDXTP,X'02'          DEFIXTP CHECK
         BNE   SPRM0100
         MVC   PDEFIXTP,=C'2'
         B     SPRM0105
*
SPRM0100 DS    0H
         MVC   PDEFIXTP,=C'1'
*
SPRM0105 DS    0H
         MVC   PDEFLTID,SPRMDFID       DEFLTID
         LH    R7,SPRMDLI              DLITOUT
         CVD   R7,PACKWK01             CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PDLITOUT,EDWORK10
         LH    R7,SPRMDSMX             DSMAX
         CVD   R7,PACKWK01             CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PDSMAX,EDWORK10
         SR    R8,R8                   CLEAR WORK REGISTER
         L     R9,SPRMEDPL             EDMPOOL
         D     R8,=F'1024'             DIVIDE BY 1024
         CVD   R9,PACKWK01             CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PEDMPOOL,EDWORK10
         TM    SPRMIAUT,X'80'          IRLMAUT CHECK
         BNO   SPRM0110
         MVC   PIRLMAUT,=C'YES'
         B     SPRM0115
*
SPRM0110 DS    0H
         MVC   PIRLMAUT,=C' NO'
*
SPRM0115 DS    0H
         MVC   PIRLMPRC,SPRMIPRC       IRLMPRC
         MVC   PIRLMSID,SPRMISID       IRLMSID
         L     R7,SPRMTOUT             IRLMRWT
         CVD   R7,PACKWK01             CONVERT TO DECIMAL
         MVC   EDWORK10,=X'40206B2020206B202120'
         ED    EDWORK10,PACKWK01+4
         MVC   PIRLMRWT,EDWORK10
```

```
         L     R7,SPRMISWT            IRLMSWT
         CVD   R7,PACKWKØ1            CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PIRLMSWT,EDWORK1Ø
         SR    R8,R8                   CLEAR WORK REGISTER
         L     R9,SPRMRMAX             MAXRBLK
         M     R8,=F'16'               CONVERT TO KBYTE
         CVD   R9,PACKWKØ1             CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PMAXRBLK,EDWORK1Ø
         L     R7,SPRMLKTS             NUMLKTS
         CVD   R7,PACKWKØ1             CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PNUMLKTS,EDWORK1Ø
         L     R7,SPRMLKUS             NUMLKUS
         CVD   R7,PACKWKØ1             CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PNUMLKUS,EDWORK1Ø
         TM    SPRMHRCL,X'8Ø'          RECALL CHECK
         BNO   SPRMØ12Ø
         MVC   PRECALL,=C'YES'
         B     SPRMØ125
*
SPRMØ12Ø DS    ØH
         MVC   PRECALL,=C' NO'
*
SPRMØ125 DS    ØH
         LH    R7,SPRMHRCD             RECALLD
         CVD   R7,PACKWKØ1             CONVERT TO DECIMAL
         MVC   EDWORK1Ø,=X'4Ø2Ø6B2Ø2Ø2Ø6B2Ø212Ø'
         ED    EDWORK1Ø,PACKWKØ1+4
         MVC   PRECALLD,EDWORK1Ø
         MVC   PRGFCOLI,SPRMREGC       RGFCOLID
         MVC   PRGFDBNA,SPRMREGN       RGFDBNAM
         MVC   PRGFNMOR,SPRMREGO       RGFNMORT
         MVC   PRGFNMPR,SPRMREGA       RGFNMPRT
         TM    SPRMREGF,X'8Ø'          RGFINSTL CHECK
         BNO   SPRMØ13Ø
         MVC   PRGFINST,=C'YES'
         B     SPRMØ135
```

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

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

Embarcadero has announced the integration of IBM's DB2 Universal Database with its database solutions packages.

Embacadero's database Solution Pack includes DBArtisan, a database administration tool used to provide a single point of control for the management of a variety of databases; ER/Studio, a multi-level data modelling tool for logical and physical database design; and Rapid SQL, an integrated development environment for creating, editing, debugging, tuning, and deploying server-side objects residing on SQL databases. Integration with DB2 Universal Database will help users to simplify and automate the processes of database management and design and improve the overall availability, stability, and reliability of their data.

For further information contact:
Embarcadero Technologies, 400 Montgomery Street, Suite 300, San Francisco, CA 94104, USA.
Tel: (415) 834 3131.
Embarcadero Europe, Broadway House, Maidenhead, Berkshire, SL6 1NJ, UK.
Tel: (01628) 418122.
URL: http://www.embarcadero.com.

* * *

Software Engineering GmbH has released DBRM/Checker Version 1.20, which checks DBRMs for violations of SQL usage or programming rules. DB2-related quality defects can be recognized at the earliest possible stage during application development.

In case of rule violations, a return-code is set according to the user-defined severity of the rule. DBRM/Checker can be used both as part of a compile procedure and as a stand-alone analysis tool for DBRM libraries providing flexible member selection.

For further information contact:
Software Engineering GmbH, Robert-Stolz-Strasse 5, D-40470 Dusseldorf, Germany.
Tel: (211) 961490.
URL: http://www.seg.de.

* * *

Xephon is holding its *DB2 '99* conference at the Mountbatten Hotel in London on 17-18 May 1999.

DB2 '99 is an update and analysis of key developments in the DB2 environment. Delegates attending *DB2 '99* will gain a clearer understanding of the evolution of DB2 both within the mainframe environment and across multiple platforms, and be better able to evaluate and exploit future DB2 developments.

Sessions at the conference cover both DB2-specific issues and issues relating to the exploitation of database technologies in the enterprise environment.

The preferential attendance fee for subscribers is £555.00 plus £64.75 VAT. For further information about *DB2 '99* contact Xephon on 01635 33823.

* * *