137

DB2 update

March 2004

In this issue

3 A REXX/ISPF program to automate the copying of rows in DB2 tables across DB2 subsystems using XCOM

23 SET operators

25 A program to fix tablespaces/ indexes with RESTRICTed access – part 2

43 Using UDB DB2 utilities to generate explain output for dynamic SQL

50 DB2 news

DB2 Update

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

Phone: 214-340-5690
Fax: 214-341-7081

Editor

Trevor Eddolls
E-mail: trevore@xephon.com

Publisher

Nicole Thomas
E-mail: nicole@xephon.com

Subscriptions and back-issues

A year's subscription to *DB2 Update*, comprising twelve monthly issues, costs $380.00 in the USA and Canada; £255.00 in the UK; £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 2000 issue, are available separately to subscribers for $33.75 (£22.50) each including postage.

DB2 Update on-line

Code from *DB2 Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at http://www.xephon.com/db2; you will need to supply a word from the printed issue.

Contributions

When Xephon is given copyright, articles published in *DB2 Update* are paid for at the rate of $160 (£100 outside North America) per 1000 words and $80 (£50) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of $32 (£20) per 100 lines. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

A REXX/ISPF program to automate the copying of rows in DB2 tables across DB2 subsystems using XCOM

Analysts and programmers are constantly asking for rows from production DB2 tables to be copied to maintenance systems to solve/simulate incidents and also to be used for testing. This work can consume a great deal of the DB2 support personnel's time, besides being repetitive.

It is to help with this requirement that the program DB2COPIX was developed. It allows rows to be copied between DB2 tables by people who have no knowledge of DB2.

At our installation, it is used to copy rows of tables from the production environment (the first DB2 subsystem) to (a second DB2 subsystem) the development, quality, or test environments via XCOM, since neither environment has shared disks to facilitate the transmission of data. These target environments reside on another mainframe/partition.

The program allows us to copy rows between DB2 tables (with the same structure and name but in different DB2 subsystems), taking as input the DB2 table or a full imagecopy (any version) of this. It also allows all the rows in the table to be copied, or just a selection of them (by means of WHERE).

The main characteristics and functions of the program DB2COPIX are:

- The main program is REXX/ISPF, which invokes a routine in Assembler. It is executed in TSO with OS/390 V2.9, DB2 V6.1, XCOM V3.0, BMC Unload Plus (or DSNTIAUL), and BMC Load Plus (or DSNUTILB).
- When being executed, the program displays the main panel where the user enters the name of the DB2 table to copy – the input can be the DB2 table in production or a full image copy version (the default is Version 0, the most recent image

copy), the output defines the environment it is copied to (the default is development), and optionally any parameters for the selection of lines using WHERE. The program builds and submits the jobs that unload the rows (it uses BMC Unload, but could use DSNTIAUL), transmit the files via XCOM, orders the execution of the jobs in maintenance via XCOM, and loads the rows in the target DB2 table (it uses BMC Load, but could use IBM Load).

- After the job is submitted, the user's TSO enters a wait state (the program calls an Assembler routine, ESPERA, which uses STIMER to generate a wait of 20 seconds, which is amendable) looking for the return of the load job. If this doesn't arrive, the following message is sent: 'copying the DB2 table *tablename*, job *jobname* current hour (*hhmmss*)'. This cycle repeats until the load job is detected in the spool or the limit of cycles is completed (30 when the input is the tablespace, 60 when it is a full imagecopy, and 60 when the copy includes all the rows). After that the wait state concludes.

The program examines the return job, from where it obtains the quantity of rows copied, which is shown in the main panel. If the copy doesn't conclude, there must have been some abnormal termination, so an explanatory message is shown in the same panel.

The program checks the validity (syntax and columns existence) of the WHERE in the case of a copy with rows selected, before generating the jobs, and it sends an explanatory message to the main panel in the event of not being valid.

Also, in the case of a copy operation with lines selected, the program generates the jobs with the appropriate deletion of lines in the target table to avoid the possible duplication of rows during the load. In the case of a copy of all the rows in a DB2 table, the program generates in the load job (before the job step) a step with ALTER PRIQTY, ALTER SECQTY for the tablespace, and indexes with the production values.

It creates its own log, where the actions of the users are recorded, such as the option chosen, the user's RACF ID, the name of the

DB2 table, whether the input is a tablespace or a full imagecopy, whether the copy is for a selection of lines, the output environment, the date and time of the execution of DB2COPIX, the jobname and jobnumber of the load job, and the number of rows copied.

The program supplies help panels (pressing F1 key). In addition it has explanatory and error messages in two versions – short and long (accessible by pressing the F1 key when it shows a short message). It uses a library of messages (ISPMLIB).

The program needs to replace some variables (hardcoded in the program):

- SSIP – DB2 subsystem of the input table.
- SSID – DB2 subsystem of the output table.
- OWNERP – owner of the input table.
- OWNERP – owner of the output table.
- LUDEVE – LU of the target system (where the output DB2 subsystem resides). Included in the XCOM job transmission.
- NODEPROD – JES2 node identifier of the input system (where the output DB2 subsystem resides). Included in the statement /*ROUTE PRINT NODEPROD of the LOAD job.
- USERFTP – RACF user and password associated with the file transfer program.
- PASSWORD – included in the SYSIN01 file of the XCOM job transmission.
- USERFTP – USERID=USERFTP and PASSWORD=USERFTP.

The benefits from using the program DB2COPIX are:

- Faster response to incidents because non-specialized personnel can perform this task without the support of a DB2 specialist.
- Improves the quality of service.

- Greater productivity.
- Testing and development are speeded up in the maintenance partition.
- It frees specialized DB2 personnel from performing this task.

DB2COPIX

```
/* REXX        DB2COPIX                 CARLOS-OSORIO@EXCITE.COM */
/*-------------------------------------------------------------*/
/* REPLACE THE FOLLOWING VARIABLES (HARDCODE)                  */
/* .SSIP       DB2 SUBSYSTEM OF THE INPUT TABLE                */
/* .SSID       DB2 SUBSYSTEM OF THE OUTPUT TABLE               */
/* .OWNERP     OWNER OF THE INPUT TABLE                        */
/* .OWNERP     OWNER OF THE OUTPUT TABLE                       */
/* .LUDEVE     LU OF THE TARGET SYSTEM                         */
/*             (WHERE THE OUTPUT DB2 SUBSYSTEM RESIDES)        */
/*             INCLUDED IN THE XCOM JOB TRANSMISSION           */
/* .NODEPROD   JES2 NODE IDENTIFY OF THE INPUT SYSTEM          */
/*             (WHERE THE OUTPUT DB2 SUBSYSTEM RESIDES)        */
/*             INCLUDED IN STATEMENT:  ROUTE    PRINT NODEPROD */
/*             OF THE LOAD JOB                                 */
/* .USERFTP/   RACF USER AND PASSWORD ASSOCIATED WITH FILE TRANSFER*/
/*  PASSWORD   INCLUDED IN THE  XCOM JOB TRANSMISSION          */
/*  USERFTP    USERID=USERFTP                                  */
/*             PASSWORD=USERFTP                                */
/*-------------------------------------------------------------*/
TRACE OFF
"ISPEXEC LIBDEF ISPPLIB DATASET ID('YOUR.PROD.PANEL.LIB')"
"ISPEXEC LIBDEF ISPMLIB DATASET ID('YOUR.PROD.MESSAGE.LIB')"
 TABLA = ''
 VERSION = 'ØØØ'
DO FOREVER
/* DEFAULTS                                                */
DB2COP   =  Ø
ENTORNO  = 'D'          /* TARGET ENVIRONMENT = DEVELOPMENT */
INPUT    = 'I'          /* INPUT  = IMAGECOPY               */
HPQTY    = 'N'          /* PRIQTY > 1ØØØØ TRACKS            */
MODO     = 'R'          /* LOAD REPLACE                     */
SQLWHERE = 'N'          /* NOT WHERE CONDITION              */
WHERE1='';WHERE2='';WHERE3='';WHERE4='';WHERE5='';WHERE6='';
DO WHILE DB2COP = Ø
  "ISPEXEC DISPLAY PANEL(DB2COPIP)"
   IF   RC = 8  THEN
        DO
       "ISPEXEC LIBDEF ISPPLIB"
       "ISPEXEC LIBDEF ISPMLIB"
        EXIT
        END
   ELSE DO
        MSG1 = ''
        MSG2 = ''
        HORARIORC = Ø
        CALL HORARIO
        IF HORARIORC = Ø THEN
          DO
          ALLOCSYSRC = Ø
          CALL ALLOCSYS
          IF ALLOCSYSRC = Ø THEN
             DO
               SYSIN.1  = "SELECT * FROM SYSIBM.SYSTABLES " ,
                          "WHERE NAME = '"TABLA"';"
               "EXECIO  *  DISKW SYSIN     (STEM SYSIN."
               "EXECIO  Ø  DISKW SYSIN     (FINIS"
               "FREE DDNAME(SYSIN)"
               CHKTABLARC = Ø
               CALL CHKTABLA
               IF CHKTABLARC = Ø THEN
                  DO
                   IF (WHERE1 <> '' ) THEN SQLWHERE = 'Y'
                   CHKPQTYRC = Ø
                   CALL CHKPQTY
                   IF CHKPQTYRC = Ø THEN
                    DO
                      CHKWHERERC = Ø
                      CALL CHKWHERE
                      IF CHKWHERERC = Ø THEN
                      DO
                        GENJOBRC = Ø
                        CALL GENJOB
                        IF GENJOBRC = Ø THEN
                          DO
                            MSG1    = 'COPYING THE TABLE 'TABLA
                            MSG2    = ''
                            VERSION = 'ØØØ'
                            MODO    = 'R'
                            DB2COP  = 1
                         END
                      END
                    END
                  END
             END
          END
        END
END
CALL CHKJOB
CALL DISRESUL
END
```

```
EXIT
/*------ R U T I N A S ------   CARLOS-OSORIO@EXCITE.COM   ------*/
HORARIO:
AUTORIZ = ' '
IF INPUT = 'T' THEN
DO
 HORA = TIME('H')
 DIA  = SUBSTR(DATE('W'),1,1)
 IF ((HORA > 8 & HORA < 13) | (HORA > 15 & HORA < 18)) & DIA <> 'S' THEN
      DO
        MSG1 = 'TIME NOT AUTHORIZED TO COPY DB2 TABLES              '
        MSG2 = 'RESTRICTION: MONDAY-FRIDAY Ø9:ØØ-13:ØØ 16:ØØ-18:ØØ  '
        AUTORIZ = 'H'
        SQLWHERE = 'C'
        MODO = 'C'
        JOBOUT = USERID() || 'ØØØ(JOBØØØØØ)'
        FILAS = Ø
        CALL GRABALOG
        HORARIORC = 1
      END
END
RETURN
/*----------------------------------------------------------------*/
ALLOCSYS:
DSNIN    = USERID() || '.TMP' || '.IN'  || TIME('S')
DSNREC   = USERID() || '.TMP' || '.REC' || TIME('S')
DSNPRINT = USERID() || '.TMP' || '.PRT' || TIME('S')
   ADDRESS TSO "ALLOC FILE(SYSIN)   DATASET('"DSNIN"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(8Ø) BLKSIZE(2792Ø) RECFM(F B) SPACE(1,1) CYL"
            IF RC <> Ø THEN
               DO
                 ADDRESS ISPEXEC "SETMSG MSG(DBCØØ1)"
                 ALLOCSYSRC = 1
                 RETURN
               END
   ADDRESS TSO "ALLOC FILE(SYSRECØØ)  DATASET('"DSNREC"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(1ØØ) BLKSIZE(279ØØ) RECFM(F B) SPACE(1,1) CYL"
            IF RC <> Ø THEN
               DO
                 ADDRESS ISPEXEC "SETMSG MSG(DBCØØ2)"
                 ALLOCSYSRC = 1
                 RETURN
               END
   ADDRESS TSO "ALLOC FILE(SYSPRINT)   DATASET('"DSNPRINT"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(133) BLKSIZE(2793Ø) RECFM(F B) SPACE(1,1) CYL"
            IF RC <> Ø THEN
               DO
```

```
                 ADDRESS ISPEXEC "SETMSG MSG(DBC003)"
                 ALLOCSYSRC = 1
                 RETURN
               END
RETURN
/*-----------------------------------------------------------------*/
CHKTABLA:
ADDRESS TSO "ALLOC FILE(SYSIN)"    "DATASET('"DSNIN"')    OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSREC00)" "DATASET('"DSNREC"')   OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSPRINT)" "DATASET('"DSNPRINT"') OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSPUNCH) DUMMY"
PUSH "END"
PUSH "RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) PARMS('SQL')"
ADDRESS TSO "DSN SYSTEM(SSIP)"
ADDRESS TSO "EXECIO * DISKR SYSREC00 (STEM REC00. FINIS"
ADDRESS TSO "FREE FILE(SYSIN)"
ADDRESS TSO "FREE FILE(SYSREC00)"
ADDRESS TSO "FREE FILE(SYSPRINT)"
ADDRESS TSO "FREE FILE(SYSPUNCH)"
CALL DELTMP
IF REC00.0 = 0 THEN
   DO
     ADDRESS ISPEXEC "SETMSG MSG(DBC010)"
     CHKTABLARC = 1;MSG1 = '';MSG2 = ''
     RETURN
   END
IF REC00.0 > 1 THEN
   DO
     ADDRESS ISPEXEC "SETMSG MSG(DBC011)"
     CHKTABLARC = 1;MSG1 = '';MSG2 = ''
     RETURN
   END
IF SUBSTR(REC00.1,29,1) = 'V' THEN
   DO
     ADDRESS ISPEXEC "SETMSG MSG(DBC012)"
     CHKTABLARC = 1;MSG1 = '';MSG2 = ''
     RETURN
   END
IF SUBSTR(REC00.1,29,1) = 'A' THEN
   DO
     ADDRESS ISPEXEC "SETMSG MSG(DBC013)"
     CHKTABLARC = 1;MSG1 = '';MSG2 = ''
     RETURN
   END
IF SUBSTR(REC00.1,29,1) = 'T' THEN
   DO
     DBNAME = SUBSTR(REC00.1,30,8)
     IF SUBSTR(DBNAME,1,3) = 'BVB' THEN
        DBNAMEO= SUBSTR(DBNAME,1,5)||ENTORNO||SUBSTR(DBNAME,7,2)
     ELSE DO
```

```
            ADDRESS ISPEXEC "SETMSG MSG(DBCØ14)"
            CHKTABLARC = 1;MSG1 = '';MSG2 = ''
            RETURN
          END
     TSNAME = SUBSTR(RECØØ.1,38,8)
     CHKTABLARC = Ø
     RETURN
   END
RETURN
/*----------------------------------------------------------------*/
CHKPQTY:
  CALL ALLOCSYS
  SYSIN.1  = "SELECT HEX(PQTY),HEX(PARTITION)"
  SYSIN.2  = "FROM SYSIBM.SYSTABLEPART"
  SYSIN.3  = "WHERE TSNAME = '"TSNAME"'"
  SYSIN.4  = "  AND DBNAME = '"DBNAME"'"
  SYSIN.5  = "ORDER BY PARTITION DESC;"
  "EXECIO  *  DISKW SYSIN     (STEM SYSIN."
  "EXECIO  Ø  DISKW SYSIN     (FINIS"
  ADDRESS TSO "ALLOC FILE(SYSIN)"    "DATASET('"DSNIN"')    OLD REUSE"
  ADDRESS TSO "ALLOC FILE(SYSRECØØ)" "DATASET('"DSNREC"')   OLD REUSE"
  ADDRESS TSO "ALLOC FILE(SYSPRINT)" "DATASET('"DSNPRINT"') OLD REUSE"
  ADDRESS TSO "ALLOC FILE(SYSPUNCH) DUMMY"
  PUSH "END"
  PUSH "RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) PARMS('SQL')"
  ADDRESS TSO "DSN SYSTEM(SSIP)"
  ADDRESS TSO "EXECIO * DISKR SYSRECØØ (STEM RECØØ. FINIS"
  ADDRESS TSO "FREE FILE(SYSIN)"
  ADDRESS TSO "FREE FILE(SYSRECØØ)"
  ADDRESS TSO "FREE FILE(SYSPRINT)"
  ADDRESS TSO "FREE FILE(SYSPUNCH)"
  CALL DELTMP
  PRIQTY = (X2D(SUBSTR(RECØØ.1,1,8))) * 4
  PART   =  X2D(SUBSTR(RECØØ.1,9,4))
  IF PRIQTY  > 5ØØØØØ THEN HPQTY = 'Y'           /* 1Ø ØØØ   TRKS */
  CALL CHKPQTYIX
RETURN
/*---------------------------------------------------------------*/
CHKPQTYIX:
SYSIN.1='';SYSIN.2='';SYSIN.3='';
CALL ALLOCSYS
SYSIN.1  = "SELECT NAME  FROM SYSIBM.SYSINDEXES"
SYSIN.2  = "WHERE TBNAME = '"TABLA"' AND DBNAME = '"DBNAME"';"
"EXECIO  *  DISKW SYSIN     (STEM SYSIN."
"EXECIO  Ø  DISKW SYSIN     (FINIS"
ADDRESS TSO "ALLOC FILE(SYSIN)"    "DATASET('"DSNIN"')    OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSRECØØ)" "DATASET('"DSNREC"')   OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSPRINT)" "DATASET('"DSNPRINT"') OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSPUNCH) DUMMY"
PUSH "END"
PUSH "RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) PARMS('SQL')"
ADDRESS TSO "DSN SYSTEM(SSIP)"
ADDRESS TSO "EXECIO * DISKR SYSRECØØ (STEM IX. FINIS"
ADDRESS TSO "FREE FILE(SYSIN)"
ADDRESS TSO "FREE FILE(SYSRECØØ)"
ADDRESS TSO "FREE FILE(SYSPRINT)"
ADDRESS TSO "FREE FILE(SYSPUNCH)"
CALL DELTMP
DO I = 1 TO IX.Ø
   IX.I = SUBSTR(IX.I,3,8)
   SYSIN.1='';SYSIN.2='';
   CALL ALLOCSYS
   SYSIN.1  = "SELECT HEX(MAX(PQTY)) FROM SYSIBM.SYSINDEXPART"
   SYSIN.2  = "WHERE IXNAME = '"IX.I"';"
   "EXECIO  *  DISKW SYSIN     (STEM SYSIN."
   "EXECIO  Ø  DISKW SYSIN     (FINIS"
   ADDRESS TSO "ALLOC FILE(SYSIN)"    "DATASET('"DSNIN"')    OLD REUSE"
   ADDRESS TSO "ALLOC FILE(SYSRECØØ)" "DATASET('"DSNREC"')   OLD REUSE"
   ADDRESS TSO "ALLOC FILE(SYSPRINT)" "DATASET('"DSNPRINT"') OLD REUSE"
   ADDRESS TSO "ALLOC FILE(SYSPUNCH) DUMMY"
   PUSH "END"
   PUSH "RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) PARMS('SQL')"
   ADDRESS TSO "DSN SYSTEM(SSIP)"
   ADDRESS TSO "EXECIO * DISKR SYSRECØØ (STEM RECØØ. FINIS"
   ADDRESS TSO "FREE FILE(SYSIN)"
   ADDRESS TSO "FREE FILE(SYSRECØØ)"
   ADDRESS TSO "FREE FILE(SYSPRINT)"
   ADDRESS TSO "FREE FILE(SYSPUNCH)"
   CALL DELTMP
   PRIQTYIX.I = (X2D(SUBSTR(RECØØ.1,1,8))) * 4
END
RETURN
/*-----------------------------------------------------------------*/
CHKWHERE:
IF SQLWHERE = 'Y' THEN
DO
 DO I=1 TO 8
    SYSIN.I = ''
 END
 MODO = 'S'
 CALL ALLOCSYS
 SYSIN.1  = "    SELECT COUNT(*) FROM OWNERP."TABLA" WHERE "
 SYSIN.2  = WHERE1
 IF WHERE2 <> '' THEN DO; SYSIN.3 = WHERE2;
    IF WHERE3 <> '' THEN DO; SYSIN.4 = WHERE3;
       IF WHERE4 <> '' THEN DO; SYSIN.5 = WHERE4;
          IF WHERE5 <> '' THEN DO; SYSIN.6 = WHERE5;
             IF WHERE6 <> '' THEN DO; SYSIN.7 = WHERE6;
                                      SYSIN.8 = ";"
                                  END;
             ELSE SYSIN.7 = ";"
                                  END;
          ELSE SYSIN.6 = ";"
                               END;
       ELSE SYSIN.5 = ";"
                            END;
    ELSE SYSIN.4 = ";"
                         END;
 ELSE  SYSIN.3 = ";"
 "EXECIO  *  DISKW SYSIN     (STEM SYSIN."
 "EXECIO  Ø  DISKW SYSIN     (FINIS"
ADDRESS TSO "ALLOC FILE(SYSIN)"    "DATASET('"DSNIN"')    OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSRECØØ)" "DATASET('"DSNREC"')   OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSPRINT)" "DATASET('"DSNPRINT"') OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSPUNCH) DUMMY"
PUSH "END"
PUSH "RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) PARMS('SQL')"
ADDRESS TSO "DSN SYSTEM(SSIP)"
IF RC <= 4 THEN
   DO
     CHKWHERERC = Ø
     RETURN
   END
ELSE
   DO
     SQLWHERE = 'N'
     MODO = 'R'
     ADDRESS ISPEXEC "SETMSG MSG(DBCØ51)"
     CHKWHERERC = 1;MSG1 = '';MSG2 = ''
     RETURN
   END
END
RETURN
/*-----------------------------   CARLOS-OSORIO@EXCITE.COM   ----*/
GENJOB:
DSNJOB   = USERID() || '.TMP' || '.JOB'  || TIME('S')
ADDRESS TSO "ALLOC FILE(JOB) DATASET('"DSNJOB"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(8Ø) BLKSIZE(2792Ø) RECFM(F B) SPACE(1,1) CYL"
             IF RC <> Ø  THEN
                DO
                ADDRESS ISPEXEC "SETMSG MSG(DBCØ21)"
                GENJOBRC = 1
                RETURN
                END
BMCCNTL  = "OWNERP."TABLA".SYSCNTL."ENTORNO || TIME('S')
BMCCNTLO = OVERLAY('D',BMCCNTL,4,1)
BMCREC   = "OWNERP."TABLA".SYSREC."ENTORNO  || TIME('S')
BMCRECO  = OVERLAY('D',BMCREC,4,1)
JOBSUFX  = SUBSTR(TIME('S'),3,3)
```

```
JOBSUFC  = SUBSTR((JOBSUFX + 1),1,3)
JOBXCOM  = USERID() || JOBSUFX
JOBCOPY  = USERID() || JOBSUFC
CALL GENJOBLOAD
"NEWSTACK"
QUEUE "//"USERID()"CC JOB OSORIO,DB2,CLASS=F,MSGCLASS=X,"
QUEUE "//      MSGLEVEL=(1,1)"
QUEUE "//STEPU1   EXEC PGM=ADUUMAIN,REGION=ØM,"
QUEUE "//         PARM='SSIP,,NEW,,MSGLEVEL(1)'"
QUEUE "//STEPLIB  DD DISP=SHR,DSN='BMC.PROD.LOAD'"
QUEUE "//SYSPRINT DD SYSOUT=*"
QUEUE "//UTPRINT  DD SYSOUT=*"
QUEUE "//SYSOUT   DD SYSOUT=*"
QUEUE "//SYSIN    DD *"
QUEUE "*"
QUEUE " UNLOAD"
IF INPUT = 'I' THEN
   DO
     IF VERSION = Ø THEN QUEUE "   INFILE IMAGECOPY FULL Ø"
     ELSE                QUEUE "   INFILE IMAGECOPY FULL -"VERSION
     QUEUE "   SHRLEVEL CHANGE"
     QUEUE "   SELECT * FROM OWNERP."TABLA
   END
ELSE
   DO
     QUEUE "   SHRLEVEL CHANGE CONSISTENT YES XBMID XBMP"
     QUEUE "   DIRECT AUTO"
     QUEUE "   SELECT * FROM OWNERP."TABLA
   END
IF SQLWHERE = 'Y' THEN
   DO
     QUEUE "   WHERE"
     DO I = 2 TO 8 WHILE SYSIN.I <> ''
        QUEUE SYSIN.I
     END
   END
QUEUE "*"
IF HPQTY = 'N' THEN
   DO
     QUEUE "//SYSREC  DD DSN="BMCREC","
     QUEUE "//           DISP=(NEW,CATLG,DELETE),"
     QUEUE "//           SPACE=(CYL,(15ØØ,1ØØ),RLSE),"
     QUEUE "//           UNIT=SYSDA"
   END
ELSE
   DO
     QUEUE "//SYSREC  DD DSN="BMCREC","
     QUEUE "//           UNIT=TAPE,VOL=(,,,3Ø),DISP=(NEW,CATLG)"
   END
QUEUE "//SYSCNTL DD DSN="BMCCNTL","
QUEUE "//            DISP=(NEW,CATLG,DELETE),"
QUEUE "//            SPACE=(TRK,(1,1),RLSE),"
QUEUE "//            UNIT=SYSDA"
QUEUE "/*"
QUEUE "//STEPU2   EXEC PGM=IKJEFTØ1,DYNAMNBR=3Ø,REGION=4M"
QUEUE "//SYSEXEC  DD DSN=YOUR.PROD.REXX.LIB,DISP=SHR"
QUEUE "//PLSBYPAS DD DUMMY"
QUEUE "//SYSTSPRT DD SYSOUT=*"
QUEUE "//SYSPRINT DD SYSOUT=*"
QUEUE "//SYSTSIN  DD *"
QUEUE " EXECUTIL  SEARCHDD(YES)"
QUEUE " %ESYSCNTL "BMCCNTL ENTORNO MODO
QUEUE "/*"
QUEUE "//STEPU3   EXEC PGM=IEBGENER"
QUEUE "//SYSIN    DD DUMMY"
QUEUE "//SYSPRINT DD SYSOUT=*"
QUEUE "//SYSUT2   DD SYSOUT=(,INTRDR)"
QUEUE "//SYSUT1   DD DATA,DLM='$$'"
QUEUE "//"JOBXCOM" JOB OSORIO,DB2,CLASS=F,MSGCLASS=Ø,TIME=144Ø,"
QUEUE "//      MSGLEVEL=(1,1)"
QUEUE "//STEPX1   EXEC PGM=XCOMJOB,PARM='ACBNAME=XCOMP,TYPE=EXECUTE'"
QUEUE "//STEPLIB  DD DSN=XCOM.LINKLIB,DISP=SHR"
QUEUE "//SYSTCPD  DD DSN=SYS1.TCPPARMS(TCIPDATA),DISP=SHR"
QUEUE "//SYSUDUMP DD SYSOUT=*"
QUEUE "//SYSINØ1  DD *"
QUEUE "LU=LUDEVE"
QUEUE "USERID=USERFTP"
QUEUE "PASSWORD=USERFTP"
QUEUE "TYPE=SEND"
QUEUE "FILETYPE=FILE"
QUEUE "FILEOPT=CREATE"
QUEUE "LFILE="BMCREC
QUEUE "FILE="BMCRECO
QUEUE "NEWXFER"
QUEUE "LFILE="BMCCNTL
QUEUE "FILE="BMCCNTLO
QUEUE "/*"
QUEUE "//STEPX2   EXEC PGM=XCOMJOB,PARM='ACBNAME=XCOMP,TYPE=EXECUTE'"
QUEUE "//STEPLIB  DD DSN=XCOM.LINKLIB,DISP=SHR"
QUEUE "//SYSTCPD  DD DSN=SYS1.TCPPARMS(TCIPDATA),DISP=SHR"
QUEUE "//SYSUDUMP DD SYSOUT=*"
QUEUE "//SYSINØ1  DD *"
QUEUE "LU=LUDEVE"
QUEUE "USERID=USERFTP"
QUEUE "PASSWORD=USERFTP"
QUEUE "TYPE=SEND"
QUEUE "FILETYPE=JOB"
QUEUE "LFILE="DSNJOBLD
QUEUE "/*"
QUEUE "//STEPX3   EXEC PGM=IDCAMS"
QUEUE "//SYSPRINT DD SYSOUT=*"
QUEUE "//SYSIN    DD *"
QUEUE "  DELETE  "BMCCNTL"  PURGE"
QUEUE "  DELETE  "BMCREC"   PURGE"
QUEUE "  DELETE  "DSNJOBLD" PURGE"
QUEUE "//"
QUEUE "$$"
QUEUE ""
ADDRESS TSO "EXECIO  *  DISKW JOB (FINIS"
ADDRESS TSO "FREE DDNAME(JOB)"
"DELSTACK"
   Y = OUTTRAP(DELSUB.)
   "SUBMIT '"DSNJOB"'"
   Y = OUTTRAP('OFF')
IF RC <> Ø THEN
   DO
   ADDRESS ISPEXEC "SETMSG MSG(DBCØ22)"
   GENJOBRC = 1
   RETURN
   END
MODO = 'R'
DB2COP  = 1
CALL DELTMP
CALL DELJOB
RETURN
/*---------------------------------------------------------------*/
GENJOBLOAD:
DSNJOBLD = USERID() || '.TMP' || '.JLD'  || TIME('S')
ADDRESS TSO "ALLOC FILE(JOBLD) DATASET('"DSNJOBLD"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(8Ø) BLKSIZE(2792Ø) RECFM(F B) SPACE(1,1) CYL"
             IF RC <> Ø   THEN
                DO
                ADDRESS ISPEXEC "SETMSG MSG(DBCØ21)"
                GENJOBRC = 1
                RETURN
                END
"NEWSTACK"
QUEUE "//"JOBCOPY" JOB OSORIO,DB2,CLASS=F,MSGCLASS=Ø,TIME=144Ø,"
QUEUE "//      MSGLEVEL=(1,1)"
QUEUE "/*ROUTE    PRINT NODEPROD"
QUEUE "//STEPL1   EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø"
QUEUE "//SYSTSPRT DD SYSOUT=*"
QUEUE "//SYSPRINT DD SYSOUT=*"
QUEUE "//SYSUDUMP DD SYSOUT=*"
QUEUE "//SYSPUNCH DD DUMMY"
QUEUE "//SYSRECØØ DD DUMMY"
QUEUE "//SYSTSIN  DD *"
QUEUE "  DSN SYSTEM(SSID)"
QUEUE "  RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) PARM('SQL')"
QUEUE "  -START DB("DBNAMEO") SPACE("TSNAME") ACCESS(RW)"
DO J = 1 TO IX.Ø
   QUEUE "  -START DB("DBNAMEO") SPACE("IX.J") ACCESS(RW)"
END
QUEUE "  END"
QUEUE "//SYSIN    DD  *"
IF SQLWHERE = 'Y' THEN
   DO
     QUEUE "  DELETE  FROM  OWNERD."TABLA
     QUEUE "  WHERE"
     DO I = 2 TO 8 WHILE SYSIN.I <> ''
        QUEUE "  "||SYSIN.I
     END
     ADDRESS TSO "FREE FILE(SYSIN)"
     ADDRESS TSO "FREE FILE(SYSRECØØ)"
     ADDRESS TSO "FREE FILE(SYSPRINT)"
     ADDRESS TSO "FREE FILE(SYSPUNCH)"
     CALL DELTMP
   END
IF MODO = 'R' THEN
   DO
     IF PART > Ø THEN
        DO I = 1 TO PART
        QUEUE "  ALTER TABLESPACE "DBNAMEO"."TSNAME" PART "I
        QUEUE "  PRIQTY "PRIQTY" SECQTY "PRIQTY%1Ø";"
        END
     ELSE
        DO
        QUEUE "  ALTER TABLESPACE "DBNAMEO"."TSNAME
        QUEUE "  PRIQTY "PRIQTY" SECQTY "PRIQTY%1Ø";"
        END
     DO J = 1 TO IX.Ø
        QUEUE "  ALTER INDEX OWNERD"."IX.J
        QUEUE "  PRIQTY "PRIQTYIX.J" SECQTY "PRIQTYIX.J%1Ø";"
     END
   END
QUEUE "/*"
QUEUE "//STEPL2   EXEC PGM=DSNUTILB,PARM='SSID,,'"
QUEUE "//SYSTSPRT DD SYSOUT=*"
QUEUE "//UTPRINT  DD SYSOUT=*"
QUEUE "//SYSUDUMP DD SYSOUT=*"
QUEUE "//SYSPRINT DD SYSOUT=*"
QUEUE "//SYSIN    DD *"
QUEUE "  REPAIR   LOG NO SET TABLESPACE "DBNAMEO"."TSNAME" NOCOPYPEND"
QUEUE "/*"
QUEUE "//STEPL3   EXEC PGM=AMUUMAIN,"
QUEUE "//         PARM='SSID,,NEW,,MSGLEVEL(1)',"
QUEUE "//         REGION=ØM"
QUEUE "//STEPLIB  DD DISP=SHR,DSN='BMC.DEVELOP.LOAD'"
QUEUE "//PLSLIGNR DD DUMMY"
QUEUE "//DSSPRINT DD SYSOUT=*"
QUEUE "//SYSPRINT DD SYSOUT=*"
QUEUE "//SYSUDUMP DD SYSOUT=*"
QUEUE "//UTPRINT  DD SYSOUT=*"
QUEUE "//SYSOUT   DD SYSOUT=*"
QUEUE "//SYSIN    DD DSN="BMCCNTLO",DISP=SHR"
QUEUE "//SYSREC   DD DSN="BMCRECO",DISP=SHR"
QUEUE "/*"
QUEUE "//STEPL4   EXEC PGM=AMUUMAIN,"
QUEUE "//         PARM='SSID,"USERID()"."JOBCOPY",TERM,,MSGLEVEL(1)',"
QUEUE "//         REGION=ØM,COND=(4,GT)"
QUEUE "//STEPLIB  DD DISP=SHR,DSN='BMC.DEVELOP.LOAD'"
QUEUE "//SYSPRINT DD SYSOUT=*"
QUEUE "//*"
QUEUE "//STEPL5   EXEC PGM=IDCAMS"
QUEUE "//SYSPRINT DD SYSOUT=*"
QUEUE "//SYSIN    DD *"
QUEUE "  DELETE  "BMCCNTLO" PURGE"
QUEUE "  DELETE  "BMCRECO " PURGE"
QUEUE "//"
QUEUE ""
ADDRESS TSO "EXECIO  *  DISKW JOBLD (FINIS"
ADDRESS TSO "FREE DDNAME(JOBLD)"
"DELSTACK"
RETURN
/*-----------------------------------------------------------------*/
CHKJOB:
"ISPEXEC LIBDEF ISPLLIB DATASET ID('YOUR.PROD.LOAD.LIB')"
IF INPUT = 'T' THEN VUELTAS = 3Ø
IF INPUT = 'I' THEN VUELTAS = 6Ø
IF HPQTY = 'Y' THEN VUELTAS = 6Ø
JOBOUT = ''
DO I=1 TO VUELTAS WHILE JOBOUT = ''
  SAY 'COPIANDO LA TABLA 'TABLA', JOB 'JOBCOPY'........'TIME()
  ADDRESS TSO "ESPERA";
  IF RC <> Ø THEN EXIT
  X = OUTTRAP(JST.)
  INTERPRET   "STATUS  "JOBCOPY;
  X = OUTTRAP('OFF')
  JSTMAX = JST.Ø
  IF SUBSTR(JST.JSTMAX,1,9) = 'IKJ56192I' THEN
       JOBOUT = WORD(JST.JSTMAX,3)
  ELSE NOP
END
RETURN
/*----------------------------------------------------------------*/
DISRESUL:
VUELTAS = 15
IF JOBOUT = '' THEN
   DO
```

```
     MSG1 = 'THE COPY OF THE TABLE 'TABLA' CONTINUE'
     MSG2 = 'WAIT THE RETURN OF THE JOB 'JOBCOPY
   END
ELSE
   DO
     JDSN = USERID()'.'JOBCOPY'.OUTLIST';
     ADDRESS TSO "OUTPUT "JOBCOPY" KEEP PRINT('"JDSN"')"
     IF RC <> Ø THEN
        DO
          MSG1 = 'NOT FOUND OUTLIST RC=' RC
          MSG2 = 'REVIEW JOB OUTPUT: 'JOBOUT
          RETURN
        END
     ADDRESS TSO "ALLOC FILE(JO) DATASET('"JDSN"') OLD REUSE "
     ADDRESS TSO "EXECIO * DISKR JO (STEM JDSN. FINIS"
     IF JDSN.Ø = Ø THEN
        DO
          MSG1 = 'NOT FOUND REGISTERS IN OUTLIST'
          MSG2 = 'REVIEW JOB OUTPUT: 'JOBOUT
          RETURN
        END
     DO I=1 TO JDSN.Ø WHILE SUBSTR(JDSN.I,31,6) <> 'STEPL3'
     END
     IF SUBSTR(JDSN.I,52,2) = 'ØØ' THEN
         DO
           MSG1 = 'THE COPY OF THE TABLE 'TABLA' FINISHED OK'
           DO I=I+1 TO JDSN.Ø WHILE WORD(JDSN.I,1) ¬= 'BMC51475I'
           END
           MSG2 = 'THE NUMBER OF THE ROWS COPIED IS: 'WORD(JDSN.I,4)
           FILAS = WORD(JDSN.I,4)
           CALL GRABALOG
         END
     ELSE
         DO
           MSG1 = 'COPY JOB 'JOBCOPY 'FINISHED NOT OK'
           MSG2 = 'REVIEW JOB OUTPUT: 'JOBOUT
         END
   W = OUTTRAP(DELOUT.)
   ADDRESS TSO "DELETE ('"JDSN"')"
   W = OUTTRAP('OFF')
   END
RETURN
/*-------------------------------------------------------------*/
GRABALOG:
ADDRESS TSO "ALLOC FILE(LOG) DATASET('YOUR.DB2COPIX.LOG') OLD REUSE"
IF RC = Ø THEN
   DO
     ADDRESS TSO "EXECIO * DISKR  LOG (STEM LOG. FINIS"
     ULTIMO = LOG.Ø + 1
     IF RC = Ø THEN
```

```
        ADDRESS TSO "EXECIO Ø DISKW LOG (OPEN FINIS"
     END
ELSE
     ADDRESS TSO "ALLOC FILE(LOG) DATASET('YOUR.DB2COPIX.LOG')" ,
              "NEW CAT REUSE UNIT(SYSDA)" ,
              "LRECL(8Ø) BLKSIZE(2792Ø) RECFM(F B) SPACE(1,1) CYL"
              IF RC <> Ø THEN
                 DO
                   ADDRESS ISPEXEC "SETMSG MSG(DBCØ41)"
                   ALLOCSYSRC = 1
                   RETURN
                 END
USERIDL = LEFT(USERID(),6)
TABLAL  = LEFT(TABLA,8)
LOG.ULTIMO = 'C' || USERIDL || TABLAL || AUTORIZ || ENTORNO || INPUT,
     || SQLWHERE || MODO || DATE('S') || TIME() || JOBOUT  || FILAS
ADDRESS TSO "EXECIO  *  DISKW LOG  (STEM LOG."
ADDRESS TSO "EXECIO  Ø  DISKW LOG  (FINIS"
"FREE DDNAME(LOG)"
RETURN
/*--------------------------------------------------------------*/
DELTMP:
W = OUTTRAP(DELTMP.)
ADDRESS TSO "DELETE ('"DSNIN"')"
ADDRESS TSO "DELETE ('"DSNREC"')"
ADDRESS TSO "DELETE ('"DSNPRINT"')"
W = OUTTRAP('OFF')
RETURN
/*--------------------------------------------------------------*/
DELJOB:
W = OUTTRAP(DELJOB.)
ADDRESS TSO "DELETE ('"DSNJOB"')"
W = OUTTRAP('OFF')
RETURN
```

YOUR.PROD.PANEL.LIB(DB2COPIP)

```
)attr
 % TYPE(TEXT) INTENS(HIGH) SKIP(ON)
 + TYPE(TEXT) INTENS(LOW) SKIP(ON)
 _ TYPE(INPUT) HILITE(&VAHILITE) CAPS(ON)
)BODY EXPAND(\\)
+\-\%COPY OF THE DB2 TABLES FROM PRODUCTION TO MAINTENANCE+\-\
%                                                      User   +&ZUSER
%  Enter the name of the table to copy                 Date   +&Fecha
%  Press Enter Key.                                    Time   +&Hora
+
%       1.+Table            _TABLA   +
+
%       2.+Input           _z+       (I|T)     Version   -_z  +(ØØØ)
+          (Imagecopy or Tablespace)  -        (Imagecopy only) ---
+
%       3.+ENVIRONMENT OF  _Z+       (D|T|C)
+          output                     -
+
+          &MSG1
+          &MSG2
+          Condition (WHERE):
+  ->_where1
+<-
+  ->_where2
+<-
+  ->_where3
+<-
+  ->_where4
+<-
+  ->_where5
+<-
+  ->_where6
+<-
)INIT
 .cursor = tabla
 .zvars ='(input,version,entorno)'
 &Fecha   = '&ZDAY..&ZMONTH..&ZYEAR'
 &Hora    = &ZTIME
)REINIT
 REFRESH (*)
 &Hora    = &ZTIME
)PROC
 REFRESH (*)
VER(&TABLA,NONBLANK,NAME)
VER(&input,NONBLANK,LIST,I,T)
IF (&input = 'I')
    VER(&version,NONBLANK,NUM)
VER(&ENTORNO,NONBLANK,LIST,D,T,C)
)END
```

MEMBER YOUR.PROD.MESSAGE.LIB(DBC00)

```
DBCØØ1   'FAILURE OF SYSIN ALLOC'    .ALARM=YES
'CHECK FOR OUT OF SPACE IN SYSDA/RACF PERMISSIONS'
DBCØØ2   'FAILURE OF SYSREC ALLOC'   .ALARM=YES
'CHECK FOR OUT OF SPACE IN SYSDA/RACF PERMISSIONS'
DBCØØ3   'FAILURE OF SYSPRINT ALLOC' .ALARM=YES
'CHECK FOR OUT OF SPACE IN SYSDA/RACF PERMISSIONS'
MEMBER  'YOUR.PROD.MESSAGE.LIB(DBCØ1)' :
DBCØ1Ø   'TABLE NOT FOUND'              .ALARM=YES
'THE TABLE ENTERED NOT EXIST. PLEASE ENTER A VALID TABLE NAME'
```

```
DBCØ11   'MORE THAN ONE TABLE FOUND'    .ALARM=YES
'MORE THAN ONE TABLE FOUND IN CATALOG DB2. CALL TO SUPPORT'
DBCØ12   'YOU HAVE ENTERED A VIEW '     .ALARM=YES
'YOU ARE TYPED A VIEW , NOT A TABLE. PLEASE ENTER A TABLE NAME'
DBCØ13   'YOU HAVE ENTERED AN ALIAS'    .ALARM=YES
'YOU ARE TYPED A ALIAS, NOT A TABLE. PLEASE ENTER A TABLE NAME'
DBCØ14   'INVALID DATABASE NAME'        .ALARM=YES
'DATABASE NAME MUST BE BVB% (ONLY APLICATIONAL DBNAME). CUSTOMIZE IT'
MEMBER  'YOUR.PROD.MESSAGE.LIB(DBCØ2)' :
DBCØ21   'FAILURE OF JOB ALLOC'    .ALARM=YES
'CHECK FOR OUT OF SPACE IN SYSDA/RACF PERMISSIONS'
DBCØ22   'FAILURE WITH JOB SUBMIT' .ALARM=YES
'FAILURE WITH JOB SUBMIT FROM A REXX . CALL TO SUPPORT'
MEMBER  'YOUR.PROD.MESSAGE.LIB(DBCØ4)' :
DBCØ41   'REXX LOG NOT ACCESSIBLE' .ALARM=YES
'FAILURE IN ACCESS/DEFINE THE REXX LOG.
MEMBER  'YOUR.PROD.MESSAGE.LIB(DBCØ5)' :
DBCØ51   'WHERE NOT VALID'         .ALARM=YES
'THE SQL CONDITION(WHERE) IS NOT VALID. COLUMN NOT FOUND'
```

JCL AND SOURCE TO ASSEMBLE ROUTINE ESPERA

```
//ESPERASS JOB  (OSORIO),'ASS',CLASS=F,MSGCLASS=X,MSGLEVEL=(1,1),
//          NOTIFY=&SYSUID
//ASM      EXEC PGM=ASMA9Ø,
//             PARM=('NOOBJECT,DECK,XREF(SHORT)')
//SYSLIB    DD DISP=SHR,DSN=SYS1.MACLIB
//          DD DISP=SHR,DSN=SYS1.MODGEN
//SYSPUNCH DD DISP=(NEW,PASS),DSN=&&OBJ,
//             SPACE=(3Ø4Ø,(4Ø,4Ø),,,ROUND),
//             UNIT=SYSALLDA,
//             DCB=(BLKSIZE=3Ø4Ø,LRECL=8Ø,RECFM=FBS,BUFNO=1)
//SYSUT1    DD DSN=&&SYSUT1,
//             SPACE=(1Ø24,(12Ø,12Ø),,,ROUND),
//             UNIT=SYSALLDA,
//             DCB=BUFNO=1
//SYSIN     DD *
         TITLE 'ROUTINE ESPERA'
ESPERA   CSECT
SAVE  (14,12)
         LR    R12,R15
USING ESPERA,R12
         ST    R13,SAVEAREA+4
LA    R2,SAVEAREA
         ST    R2,8(,R13)
LR    R13,R2
         EXTRACT TIOTADR,'S',FIELDS=TIOT
L     R8,TIOTADR
         MVC   JOBNAME,Ø(R8)              TO SELECT TSO USERS WHO
```

```
*                                          CAN EXECUTE THIS ROUTINE
CLC   JOBNAME,=C'USER1   '       PERMIT TSO USER *USER1*
         BE    SIGUE
         CLC   JOBNAME,=C'USER2   '       PERMIT TSO USER *USER2*
         BE    SIGUE
         CLC   JOBNAME,=C'USER3   '       PERMIT TSO USER *USER3*
         BE    SIGUE
         L     R13,4(R13)
         RETURN (14,12),RC=12
SIGUE    EQU   *
         STIMER WAIT,DINTVL=WAITTIME
RETURN   EQU   *
         L     R13,4(R13)
         RETURN (14,12),RC=Ø
         USING ESPERA,R12
SAVEAREA DS    18F
EXITSAVE DS    18F
TIOTADR  DS    F
WTOLST1  DC    AL2(WTOLST1X-WTOLST1)
         DC    H'Ø'
JOBNAME  DS    CL8
WTOLST1X EQU   *
WAITTIME DC    C'ØØØØ2ØØØ'               2Ø SECONDS
DEC      DS    D
         LTORG
RØ       EQU   Ø
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         PRINT GEN
         IKJTCB
         END   ESPERA
/*
//SYSLIN   DD DUMMY
//SYSPRINT DD SYSOUT=*
//*
//LKED    EXEC PGM=IEWL,
//            PARM=('LIST,MAP,AC=1,AMODE=ANY,RMODE=24'),
//             COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=&&SYSUT1,
//             SPACE=(1Ø24,(12Ø,12Ø),,,ROUND),
//             UNIT=SYSALLDA,
//             DCB=BUFNO=1
//SYSLIN   DD DISP=(OLD,DELETE),DSN=&&OBJ
//         DD DDNAME=SYSIN
//SYSLMOD  DD DISP=SHR,
//             DSN=YOUR.PROD.LOAD.LIB
//SYSIN    DD *
  ENTRY ESPERA
  NAME ESPERA(R)
/*
//
```

Carlos Osorio Montoya
Database and MQSeries Administrator (Peru)

SET operators

This article looks at the SET operators that are available in DB2 UDB for LUW. I ran the following SQL on a Windows 2000 machine running DB2 V8 FP2.

There are three SET operators available – UNION, EXCEPT, and INTERSECT. I will look at each of these in turn.

For all our examples we will use two tables, hm01 and hm02, created and populated as follows:

```
create table hmØ1 (id int, name char(1Ø))
create table hmØ2 (id int, name char(1Ø))

insert into hmØ1 values(1,'AAA'),(2,'BBB'),(3,'CCC')
insert into hmØ2 values(1,'AAA'),(4,'DDD'),(5,'EEE')
```

The UNION operator lets you join the results of two SELECT statements, with duplicate rows being eliminated (hence requiring a sort). The UNION ALL operator does not eliminate duplicate rows (and hence does not require a sort).

So if we use the UNION operator and select everything from our two tables, you can see that the operator returns only unique rows (one of the duplicate ‘1 AAA’ rows has been eliminated) and the number of rows returned is 5.

```
>db2 select * from hm01 union select * from hm02

ID          NAME
---------- ---------
         1 AAA
         2 BBB
         3 CCC
         4 DDD
         5 EEE

  5 record(s) selected.
```

If we now use the UNION ALL operator, we can see we have two occurrences of the ‘1 AAA’ row and the number of rows returned is 6.

```
>db2 select * from hmØ1 union all select * from hmØ2

ID          NAME
---------- ---------
         1 AAA
         4 DDD
         5 EEE
         1 AAA
         2 BBB
         3 CCC

  6 record(s) selected.
```

If you did a visual explain on both of the above statements using the Control Center, you can see that the UNION version does a SORT whereas the UNION ALL version doesn’t.

Now let’s look at the EXCEPT operator. The EXCEPT operator returns rows in the table to the left of the operator, which do not exist in the table to the right of the operator. The operator is used as shown in the example below.

```
>db2 select * from hmØ1 except select * from hmØ2

ID          NAME
---------- ---------
```

```
          2 BBB
          3 CCC
```

You can see that the result returned is the two rows in table hm01, which are not present in table hm02.

Finally, let's look at the INTERSECT operator. As the name suggests, and in line with normal mathematical set theory, the INTERSECT operator will return rows that exist in only both tables specified either side of the operator. This is shown in the example below.

```
>db2 select * from hmØ1 intersect select * from hmØ2

ID          NAME
---------- ----------
          1 AAA

  1 record(s) selected.
```

The only row that is present in both table hm01 and hm02 is the '1 AAA' row, and this is the row that is returned.

You can use the additional ALL parameter for both the EXCEPT and INTERSECT operators. The manual says that this works in the same way as with UNION and UNION ALL. However, there is no difference as far as I can see between EXCEPT and EXCEPT ALL and INTERSECT and INTERSECT ALL. If you specified EXCEPT ALL in the third example, you would still get only the two rows back, and if you specified INTERSECT ALL in the fourth example, you would still get only the one row returned.

C Leonard
Freelance Consultant (UK)

A program to fix tablespaces/indexes with RESTRICTed access – part 2

This month we conclude the code to change the availability of tablespaces and indexes to RW from any restricted status.

```
        IF ALLOCSYSRC = 1 THEN RETURN;
```

```
 SYSIN.1  = "SELECT ICDATE,ICTIME,DSNAME "
 SYSIN.2  = "FROM SYSIBM.SYSCOPY"
 SYSIN.3  = "WHERE DBNAME   = '"DATABASE"'"
 SYSIN.4  = "  AND TSNAME   = '"NAME"'"
 SYSIN.5  = "  AND ICTYPE   = 'F'"
 SYSIN.6  = "  AND ICBACKUP = '  '"
 SYSIN.7  = "  AND DSNUM    = Ø"
 SYSIN.8  = "  AND ICDATE LIKE 'Ø%'"
 SYSIN.9  = "ORDER BY ICDATE DESC, ICTIME DESC;"
"EXECIO  *   DISKW SYSIN (STEM SYSIN."
"EXECIO  Ø   DISKW SYSIN (FINIS"
 ADDRESS TSO
"ALLOC FILE(SYSIN)    DATASET('"DSNIN"') OLD REUSE"
"ALLOC FILE(SYSRECØØ) DATASET('"DSNREC"') OLD REUSE"
"ALLOC FILE(SYSPRINT) DATASET('"DSNPRINT"') OLD REUSE"
"ALLOC FILE(SYSPUNCH) DUMMY"
 PUSH "END"
 PUSH "RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) PARMS('SQL')"
 ADDRESS TSO "DSN SYSTEM("SSID")"
 ADDRESS TSO "EXECIO * DISKR SYSRECØØ (STEM RECØØ. FINIS"
 ADDRESS TSO "FREE FILE(SYSIN)"
 ADDRESS TSO "FREE FILE(SYSRECØØ)"
 ADDRESS TSO "FREE FILE(SYSPRINT)"
 ADDRESS TSO "FREE FILE(SYSPUNCH)"
 CALL DELTMP;
    DO
   "NEWSTACK"
    ICDSNAME = ''
    V = ''
   "ISPEXEC TBCREATE TICS NAMES(V FECHA HORA VERSION DSNAME)",
   "NOWRITE REPLACE"
       DO I = 1 TO RECØØ.Ø
       FECHA  = SUBSTR(RECØØ.I,1,6)
       HORA   = SUBSTR(RECØØ.I,7,6)
       VERSION= (I-1)*(-1)
       DSNAME = SUBSTR(RECØØ.I,13,44)
      "ISPEXEC TBADD TICS"
       END
   "ISPEXEC ADDPOP  ROW(5) COLUMN(5)"
    RESREC2_EXIT = 'N'
   "ISPEXEC TBTOP   TICS"
   "ISPEXEC TBDISPL TICS PANEL(DB2TREIC)"
    IF RC = 8 THEN
       DO
       RESREC2_EXIT = 'S'
       R = NRES + 1
       END
    ELSE
       DO
       ICDSNAME = STRIP(DSNAME)
```

```
              CALL RESRECPJ;
              END
           END
       "DELSTACK"
       "ISPEXEC REMPOP"
       "ISPEXEC TBEND    TICS"
        END
     IF E = '2' THEN
        DO
        CALL RESRECPJ;
        END
     IF E = '3' THEN
        DO
        CALL RESREPAJ;
        END
     END
"DELSTACK"
"ISPEXEC REMPOP"
CALL LIBDEFPANEL;
RETURN
/*-----------------------------------------------------------------*/
RESRECPJ:
"NEWSTACK"
DSNJOB   = USERID() || '.RES' || '.JOB'  || TIME('S')
ADDRESS TSO "ALLOC FILE(JOB) DATASET('"DSNJOB"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(8Ø) BLKSIZE(2792Ø) RECFM(F B) SPACE(1,1) CYL"
             IF RC <> Ø THEN
                DO
                ADDRESS ISPEXEC "SETMSG MSG(DBCØ21)"
                RETURN
             END
JOBNAME = SUBSTR((USERID() || 'RC'),1,8)
QUEUE "//"JOBNAME" JOB (DB2),'OSORIO',MSGCLASS=X,"
QUEUE "// CLASS=S,MSGLEVEL=(1,1),NOTIFY=&SYSUID"
QUEUE "//STEP1 EXEC PGM=AFRMAIN,REGION=ØM,"
QUEUE "//      PARM='"SSID",,NEW,MSGLEVEL(1),,RDB2STAT(RW)'"
QUEUE "//SYSPRINT DD SYSOUT=*"
QUEUE "//UTPRINT  DD SYSOUT=*"
QUEUE "//SYSIN DD *"
QUEUE " RECOVER  TABLESPACE "DATABASE"."NAME" DSNUM ALL";
IF E = 1 THEN
QUEUE " TOCOPY   "ICDSNAME;
QUEUE " ANALYZE  YES"
QUEUE " REDEFINE YES"
QUEUE " RECOVER  INDEX (ALL)"
QUEUE "          TABLESPACE "DATABASE"."NAME
QUEUE " ANALYZE  YES"
QUEUE " REDEFINE YES"
QUEUE ""
```

```
ADDRESS TSO "EXECIO * DISKW JOB (FINIS"
ADDRESS TSO "EXECIO * DISKR JOB (STEM JOBRECQ. FINIS"
"ISPEXEC TBCREATE TJOBREC",
"NAMES(JOBREC)",
"NOWRITE REPLACE"
DO I = 1 TO JOBRECQ.Ø
   JOBREC = JOBRECQ.I
  "ISPEXEC TBADD  TJOBREC"
END
"ISPEXEC ADDPOP ROW(7) COLUMN(7)"
RESREC3_EXIT = 'N'
ADDRESS TSO "FREE DDNAME(JOB)"
"ISPEXEC TBTOP    TJOBREC"
"ISPEXEC TBDISPL  TJOBREC PANEL(DB2TRECJ)"
IF RC = 8 THEN
   DO
   RESREC3_EXIT = 'S'
   R = NRES + 1
   END
ELSE
   DO
   Y = OUTTRAP(DELSUB.)
  "SUBMIT '"DSNJOB"'"
   Y = OUTTRAP('OFF')
       IF RC <> Ø THEN
          DO
          ADDRESS ISPEXEC "SETMSG MSG(DBCØ22)"
          RETURN
          END
   CALL GRABALOG
   END
"DELSTACK"
"ISPEXEC REMPOP"
"ISPEXEC TBEND    TJOBREC"
CALL DELJOB;
RETURN
/*-------------------------------------------------------------*/
RESREPAJ:
"NEWSTACK"
DSNJOB   = USERID() || '.RES' || '.JOB'  || TIME('S')
ADDRESS TSO "ALLOC FILE(JOB) DATASET('"DSNJOB"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(8Ø) BLKSIZE(2792Ø) RECFM(F B) SPACE(1,1) CYL"
             IF RC <> Ø THEN
                DO
                ADDRESS ISPEXEC "SETMSG MSG(DBCØ21)"
                RETURN
             END
JOBNAME = SUBSTR((USERID() || 'RP'),1,8)
QUEUE "//"JOBNAME" JOB (DB2),'OSORIO',MSGCLASS=X,"
QUEUE "// CLASS=S,MSGLEVEL=(1,1),REGION=ØM,NOTIFY=&SYSUID"
QUEUE "//REPAIR  EXEC PGM=DSNUTILB,PARM='"SSID"',COND=EVEN"
QUEUE "//SYSPRINT DD SYSOUT=*"
QUEUE "//UTPRINT  DD SYSOUT=*"
QUEUE "//SYSERR   DD SYSOUT=*"
QUEUE "//SYSIN    DD *"
IF TYPE = 'TABLESPACE' THEN
QUEUE " REPAIR LOG NO SET TABLESPACE "DATABASE"."NAME" NORCVRPEND";
IF TYPE = 'INDEX'      THEN
QUEUE " REPAIR LOG NO SET INDEX      "IXCREATOR"."NAME" NORCVRPEND";
QUEUE ""
ADDRESS TSO "EXECIO * DISKW JOB (FINIS"
ADDRESS TSO "EXECIO * DISKR JOB (STEM JOBREPQ. FINIS"
"ISPEXEC TBCREATE TJOBREP",
"NAMES(JOBREP)",
"NOWRITE REPLACE"
DO I = 1 TO JOBREPQ.Ø
   JOBREP = JOBREPQ.I
  "ISPEXEC TBADD  TJOBREP"
END
"ISPEXEC ADDPOP ROW(7) COLUMN(7)"
RESREC4_EXIT = 'N'
ADDRESS TSO "FREE DDNAME(JOB)"
"ISPEXEC TBTOP    TJOBREP"
"ISPEXEC TBDISPL  TJOBREP PANEL(DB2TREPJ)"
IF RC = 8 THEN
   DO
   RESREC4_EXIT = 'S'
   R = NRES + 1
   END
ELSE
   DO
   Y = OUTTRAP(DELSUB.)
  "SUBMIT '"DSNJOB"'"
   Y = OUTTRAP('OFF')
       IF RC <> Ø THEN
          DO
          ADDRESS ISPEXEC "SETMSG MSG(DBCØ22)"
          RETURN
          END
   CALL GRABALOG
   END
"DELSTACK"
"ISPEXEC REMPOP"
"ISPEXEC TBEND    TJOBREP"
CALL DELJOB;
RETURN
/*-------------------------------------------------------------------*/
TBDELROWS:
"ISPEXEC TBSTATS TRES ROWCURR(TRESROWS)"
```

```
"ISPEXEC TBTOP TRES"
DO I = 1 TO TRESROWS
"ISPEXEC TBSKIP   TRES "
"ISPEXEC TBDELETE TRES"
END
RETURN
/*------------------------------------------------------------------*/
GRABALOG:
ADDRESS TSO "ALLOC FILE(LOG) DSNAME('"LOGPREFIX".DB2RES.LOG') OLD REUSE"
IF RC = Ø THEN
   DO
   ADDRESS TSO "EXECIO * DISKR  LOG (STEM LOG. FINIS"
   IF RC = Ø THEN ULTIMO = LOG.Ø + 1
   END
ELSE
   DO
   ADDRESS TSO "ALLOC FILE(LOG) DSNAME('"LOGPREFIX".DB2RES.LOG')" ,
               "NEW CAT REUSE UNIT(SYSDA)" ,
               "LRECL(8Ø) BLKSIZE(2792Ø) RECFM(F B) SPACE(1,1) CYL"
                IF RC <> Ø THEN
                   DO
                   ADDRESS ISPEXEC "SETMSG MSG(DBCØ41)"
                   RETURN
                END
   END
USERIDL = LEFT(USERID(),6)
SSIDL   = LEFT(SSID,4)
SELECT
  WHEN DISRES_ZERO = 1 THEN
    LOG.ULTIMO = 'D' || USERIDL   || SSIDL   || LEFT(' ',18),
                     || LEFT(' ',8)          || LEFT(' ',8),
                     || LEFT(' ',1Ø)         || LEFT(' ',4),
                     || DATE('S') || TIME()
  WHEN DISRES_ADD = 1 THEN
    LOG.ULTIMO = 'D' || USERIDL   || SSIDL   || LEFT(STATUS,18),
                     || LEFT(NAME,8)         || LEFT(DATABASE,8),
                     || LEFT(TYPE,1Ø)        || LEFT(PART,4),
                     || DATE('S') || TIME()
  OTHERWISE
    LOG.ULTIMO = 'R' || USERIDL   || SSIDL   || LEFT(STATUS,18),
                     || LEFT(NAME,8)         || LEFT(DATABASE,8),
                     || LEFT(TYPE,1Ø)        || LEFT(PART,4),
                     || DATE('S') || TIME()
END
ADDRESS TSO "EXECIO  *  DISKW LOG  (STEM LOG."
ADDRESS TSO "EXECIO  Ø  DISKW LOG  (FINIS"
"FREE DDNAME(LOG)"
DISRES_ZERO = Ø
DISRES_ADD  = Ø
RETURN
```

```
/*----------------------------------------------------------------*/
ALLOCSYS:
ALLOCSYSRC = Ø
DSNIN    = USERID() || '.RES' || '.IN'  || TIME('S')
DSNREC   = USERID() || '.RES' || '.REC' || TIME('S')
DSNPRINT = USERID() || '.RES' || '.PRT' || TIME('S')
ADDRESS TSO "ALLOC FILE(SYSIN)   DATASET('"DSNIN"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(8Ø) BLKSIZE(2792Ø) RECFM(F B) SPACE(1,1) CYL"
             IF RC <> Ø THEN
                DO
                ADDRESS ISPEXEC "SETMSG MSG(DBCØØ1)"
                ALLOCSYSRC = 1
                RETURN
             END
/*----------------------------------------------------------------*/
ADDRESS TSO "ALLOC FILE(SYSRECØØ)  DATASET('"DSNREC"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(1ØØ) BLKSIZE(279ØØ) RECFM(F B) SPACE(1,1) CYL"
             IF RC <> Ø THEN
                DO
                ADDRESS ISPEXEC "SETMSG MSG(DBCØØ2)"
                ALLOCSYSRC = 1
                RETURN
             END
/*----------------------------------------------------------------*/
ADDRESS TSO "ALLOC FILE(SYSPRINT)   DATASET('"DSNPRINT"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(133) BLKSIZE(2793Ø) RECFM(F B) SPACE(1,1) CYL"
             IF RC <> Ø THEN
                DO
                ADDRESS ISPEXEC "SETMSG MSG(DBCØØ3)"
                ALLOCSYSRC = 1
                RETURN
             END
RETURN
/*----------------------------------------------------------------*/
LIBDEFWINDJ:
SELECT
  WHEN SSID = 'DB2P' THEN
 "ISPEXEC LIBDEF ISPPLIB DATASET ID('YOUR.PROD.WINDOWSJ.USER')"
  WHEN SSID = 'DB2D' THEN
 "ISPEXEC LIBDEF ISPPLIB DATASET ID('YOUR.DESA.WINDOWSJ.USER')"
  WHEN SSID = 'DB2T' THEN
 "ISPEXEC LIBDEF ISPPLIB DATASET ID('YOUR.TEST.WINDOWSJ.USER')"
  OTHERWISE
 "ISPEXEC LIBDEF ISPPLIB DATASET ID('YOUR.PROD.WINDOWSJ.USER')"
END
RETURN
/*----------------------------------------------------------------*/
```

```
LIBDEFPANEL:
SELECT
  WHEN SSID = 'DB2P' THEN
 "ISPEXEC LIBDEF ISPPLIB DATASET ID('YOUR.PROD.PANELLIB.USER')"
  WHEN SSID = 'DB2D' THEN
 "ISPEXEC LIBDEF ISPPLIB DATASET ID('YOUR.DESA.PANELLIB.USER')"
  WHEN SSID = 'DB2T' THEN
 "ISPEXEC LIBDEF ISPPLIB DATASET ID('YOUR.TEST.PANELLIB.USER')"
  OTHERWISE
 "ISPEXEC LIBDEF ISPPLIB DATASET ID('YOUR.PROD.PANELLIB.USER')"
END
RETURN
/*----------------------------------------------------------------*/
DELTMP:
W = OUTTRAP(DELTMP.)
ADDRESS TSO "DELETE ('"DSNIN"')"
ADDRESS TSO "DELETE ('"DSNREC"')"
ADDRESS TSO "DELETE ('"DSNPRINT"')"
W = OUTTRAP('OFF')
RETURN
/*---------------------------------------------------------------*/
DELJOB:
W = OUTTRAP(DELJOB.)
ADDRESS TSO "DELETE ('"DSNJOB"')"
W = OUTTRAP('OFF')
RETURN
/*----------------------------------------------------------------*/
DISUTIBMC:
STATUSRBMC = '';
IF TYPE  = 'TABLESPACE' THEN KIND = 'TP';
IF TYPE  = 'INDEX'      THEN KIND = 'IP';
DO I = 1 TO 9
   SYSIN.I = ''
END
CALL ALLOCSYS;
IF ALLOCSYSRC = 1 THEN RETURN;
 SYSIN.1  = "SELECT UTILID,STATUS "
 SYSIN.2  = "FROM   BMCUTIL.CMN_BMCUTIL"
 SYSIN.3  = "WHERE  UTILID = ( SELECT DISTINCT UTILID "
 SYSIN.4  = "                    FROM BMCUTIL.CMN_BMCSYNC"
 SYSIN.5  = "                   WHERE NAME1  = '"DATABASE"'"
 SYSIN.6  = "                     AND NAME2  = '"NAME"'"
 SYSIN.7  = "                     AND KIND   = '"KIND"');"
"EXECIO  *  DISKW SYSIN     (STEM SYSIN."
"EXECIO  Ø  DISKW SYSIN     (FINIS"
 ADDRESS TSO "ALLOC FILE(SYSIN)"    "DATASET('"DSNIN"')    OLD REUSE"
 ADDRESS TSO "ALLOC FILE(SYSRECØØ)" "DATASET('"DSNREC"')   OLD REUSE"
 ADDRESS TSO "ALLOC FILE(SYSPRINT)" "DATASET('"DSNPRINT"') OLD REUSE"
 ADDRESS TSO "ALLOC FILE(SYSPUNCH) DUMMY"
 PUSH "END"
 PUSH "RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) PARMS('SQL')"
 ADDRESS TSO "DSN SYSTEM("SSID")"
 ADDRESS TSO "EXECIO * DISKR SYSRECØØ (STEM BMCUTI. FINIS"
 ADDRESS TSO "FREE FILE(SYSIN)"
 ADDRESS TSO "FREE FILE(SYSRECØØ)"
 ADDRESS TSO "FREE FILE(SYSPRINT)"
 ADDRESS TSO "FREE FILE(SYSPUNCH)"
 CALL DELTMP
 IF BMCUTI.Ø = Ø THEN NOP                  /* NO HAY UTILITARIOS BMC */
 ELSE
    DO
    UTILIDR    = SUBSTR(BMCUTI.1,1,16)
    STATUSR    = SUBSTR(BMCUTI.1,17,1)
    IF STATUSR = 'X' | ,                 /* BMCUTIL UTILITARIO EN X */
       STATUSR = 'A' | ,                 /* BMCUTIL UTILITARIO EN A */
       STATUSR = 'P' | ,                 /* BMCUTIL UTILITARIO EN P */
       STATUSR = 'I'   THEN              /* BMCUTIL UTILITARIO EN I */
       DO
       STATUSRBMC = 'X'
       ADDRESS ISPEXEC "SETMSG MSG(DBC3Ø8)";
       END
    ELSE CALL TERUTIBMC
    END
RETURN
/*----------------------------------------------------------------*/
TERUTIBMC:
TERUTIBMCOK = Ø
DSNZT    = USERID() || '.RES' || '.IN'  || TIME('S')
   ADDRESS TSO "ALLOC FILE(ZTEMPN)  DATASET('"DSNZT"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(8Ø) BLKSIZE(2792Ø) RECFM(F B) SPACE(1,1) CYL"
PLANNM  = 'ABUD23ØØ'
OPTNAME = ''
QUEUE '+TERM UTIL('UTILIDR')'
QUEUE 'END'
BMCDSN = 'ABUDSN /'SSID'/'PLANNM'/'OPTNAME'/'

X = OUTTRAP("OUTPUT.",'*',"NOCONCAT")
BMCDSN
IF (RC = Ø) THEN TERUTIBMCOK = 1
ELSE
   ADDRESS ISPEXEC "SETMSG MSG(DBC2Ø6)";      /*FALLO TERM UTIL BMC*/
"EXECIO" OUTPUT.Ø "DISKW" ZTEMPN "(OPEN STEM OUTPUT.)"
IF RC > Ø THEN SAY 'WRITE RC = ' RC
"EXECIO"        Ø "DISKW" ZTEMPN "(STEM OUTPUT FINIS)"
IF RC > Ø THEN SAY 'CLOSE RC = ' RC
Y = OUTTRAP(OFF)
IF TERUTIBMCOK = 1 THEN
   DO
  "FREE DDNAME(ZTEMPN)"
```

```
   W = OUTTRAP(DELZT.)
   ADDRESS TSO "DELETE ('"DSNZT"')"
   W = OUTTRAP('OFF')
   END
RETURN
/*-------------------------------------------------------------*/
TERUTIIBMSTOP:
DISUTI:
W = OUTTRAP('UTI.')
    QUEUE "-DIS UTILITY(*)"
    QUEUE "END"
   "DSN SYSTEM("SSID")"
W = OUTTRAP('OFF')
    IF RC > Ø THEN
       DO
       ADDRESS ISPEXEC "SETMSG MSG(DBC2ØØ)"           /* NO HAY DB2 */
       RETURN
       END
IF SUBSTR(UTI.1,1,8) = 'DSNU112I' THEN RETURN
I = 1;
DO I = I WHILE I < UTI.Ø
   IF (SUBSTR(UTI.I,1,8) = 'DSNU1Ø5I' |,         /* UTILITY ACTIVE  */
       SUBSTR(UTI.I,1,8) = 'DSNU1ØØI' ) THEN     /* UTILITY STOPPED */
      DO J = 1 TO 7
         SELECT
         WHEN J = 1  THEN UTIUSERI = WORD(UTI.I,7)
         WHEN J = 2  THEN NOP
         WHEN J = 3  THEN UTILIDI  = WORD(UTI.I,3)
         WHEN J = 4  THEN NOP
         WHEN J = 5  THEN UTILITYI = WORD(UTI.I,3)
         WHEN J = 6  THEN PHASEI   = WORD(UTI.I,3)
         WHEN J = 7  THEN STATUSRIBM = WORD(UTI.I,3)
         END
         IF STATUSRIBM = 'STOPPED' THEN CALL TERUTI;
         IF J < 7 THEN I = I + 1;
      END
END
RETURN
/*-------------------------------------------------------------*/
TERUTI:
   W = OUTTRAP('TER.')
       QUEUE "-TERM UTILITY("UTILIDI")"
       QUEUE "END"
      "DSN SYSTEM("SSID")"
   W = OUTTRAP('OFF')
RETURN
/*-------------------------------------------------------------*/
TRAEIXCREATOR:
DO I = 1 TO 9
   SYSIN.I = ''
```

```
END
CALL ALLOCSYS
IF ALLOCSYSRC = 1 THEN RETURN;
SYSIN.1  = "SELECT CREATOR FROM SYSIBM.SYSINDEXES"
SYSIN.2  = "WHERE INDEXSPACE = '"NAME"' AND DBNAME = '"DATABASE"';"
"EXECIO  *  DISKW SYSIN     (STEM SYSIN."
"EXECIO  Ø  DISKW SYSIN     (FINIS"
ADDRESS TSO "ALLOC FILE(SYSIN)"    "DATASET('"DSNIN"')    OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSRECØØ)" "DATASET('"DSNREC"')   OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSPRINT)" "DATASET('"DSNPRINT"') OLD REUSE"
ADDRESS TSO "ALLOC FILE(SYSPUNCH) DUMMY"
PUSH "END"
PUSH "RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) PARMS('SQL')"
ADDRESS TSO "DSN SYSTEM("SSID")"
ADDRESS TSO "EXECIO * DISKR SYSRECØØ (STEM IX. FINIS"
ADDRESS TSO "FREE FILE(SYSIN)"
ADDRESS TSO "FREE FILE(SYSRECØØ)"
ADDRESS TSO "FREE FILE(SYSPRINT)"
ADDRESS TSO "FREE FILE(SYSPUNCH)"
CALL DELTMP
IXCREATOR = STRIP(SUBSTR(IX.1,1,8));
RETURN
/* ------------------  FIN  END  -- carlos-osorio@excite.com --*/
```

YOUR.PANELI(DB2RESSP)

```
)ATTR
 % TYPE(TEXT)   INTENS(HIGH) SKIP(ON)
   color(turquoise)
 + TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
 _ TYPE(INPUT)  INTENS(HIGH) CAPS(ON)            HILITE(USCORE)
 ! TYPE(OUTPUT) INTENS(LOW)  CAPS(ON) JUST(LEFT) HILITE(REVERSE)
   color(turquoise)
 @ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)            HILITE(REVERSE)
   color(turquoise)
)BODY EXPAND(\\)
@                            SUBSYSTEM(S) DB2
@
%COMMAND ===>_ZCMD
%
%                         Opc  Subsystem  DB2
+                        %--  -------------
)MODEL
+                         _O+ !SDB2+
)INIT
 .help   = db2ressh
 .cursor = O
 &SCRO   = PAGE
)REINIT
```

```
 .cursor = O
)PROC
if (.PFKEY = 'PFØ1')
    &pfkeyin = 's'
IF (&ZTDSELS > ØØØØ)
    VER(&O,LIST,S)
)END
```

YOUR.PANELI(DB2RESSH)

```
)attr
 % TYPE(TEXT)   INTENS(HIGH) SKIP(ON)
   color(turquoise)
 + TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
 _ TYPE(INPUT)  INTENS(HIGH) CAPS(ON)            HILITE(USCORE)
 ! TYPE(OUTPUT) INTENS(LOW)  CAPS(ON) JUST(LEFT) HILITE(REVERSE)
   color(turquoise)
 @ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)            HILITE(REVERSE)
   color(turquoise)
)BODY EXPAND(\\)
@                            SUBSYSTEM(S) DB2
@
%   Opc (OPTIONS) valids :+
+
+  %S+       - To select the subsystem DB2.
+
+
%  %Description of the columns:+
+
+  %Subsystem + Name/Identification of subsystem DB2
+               defined in MVS - OS/39Ø (SYS1.PARMLIB).
+
)END
```

YOUR.XXXX.PANELLIB.USER(DB2RESP)

```
)ATTR
 % TYPE(TEXT)   INTENS(HIGH) SKIP(ON)
   color(turquoise)
 + TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
 _ TYPE(INPUT)  INTENS(HIGH) CAPS(ON)            HILITE(USCORE)
 ! TYPE(OUTPUT) INTENS(LOW)  CAPS(ON) JUST(LEFT) HILITE(REVERSE)
   color(turquoise)
 @ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)            HILITE(REVERSE)
   color(turquoise)
)BODY EXPAND(\\)
@                     TABLESPACE(S)/INDEX(ES) IN RESTRICT
@
```

```
%COMMAND ===>_ZCMD                                                    +%SCROLL
===>_SCRO+
%
%   Opc  Status              Name      Database  Part  Type
+  %--  -----------------  --------  --------  ----  ----------
)MODEL
+   _O+ !STATUS           +!NAME    +!DATABASE+!PART+!TYPE      +
)INIT
 .help   = db2resh
 .cursor = o
 &SCRO   = CSR
)REINIT
 .cursor = o
)PROC
if (.PFKEY = 'PFØ1')
    &pfkeyin = 's'
IF (&ZTDSELS > ØØØØ)
    VER(&O,LIST,R)
)END
```

YOUR.XXXX.PANELLIB.USER(DB2RESH)

```
)attr
 % TYPE(TEXT)   INTENS(HIGH) SKIP(ON) color(turquoise)
 ! TYPE(TEXT)   INTENS(HIGH) SKIP(ON) color(red)
 + TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
 _ TYPE(INPUT)  INTENS(HIGH) CAPS(ON)            HILITE(USCORE)
 @ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)            HILITE(REVERSE)
   color(turquoise)
)BODY EXPAND(\\)
@                     TABLESPACE(S)/INDEX(ES) IN RESTRICT
@
%   Opc (OPTIONS) valids :+
+
+  %R+       - Fix (repair). Unique option.
+              Generate commands and/or jobs for fix TS/IX in RESTRICT.
+
!              Always begin for Fix (repair) the TS and then the IXs.
+
%  %Description of the columns :+
+
+  %Status+  - Status of+%TS/IX (Tablespace/Index).+
+              May be a combination of:
%
RW,RO,STOP,STOPE,STOPP,UT,UTRO,UTRW,UTUT,ACHKP,AUXW,CHKP,COPY,
%
GRECP,ICOPY,LPL,LSTOP,PSRBD,RBDP,RBDP*,RECP,REORP,REST,RESTP.
+
+  %Name+    - Name of TS/IX in RESTRICT.
```

```
+
+  %Database+- Database of TS/IX en RESTRICT.
+
+  %Part+    - Partition of TS/IX en RESTRICT.
+
+  %Type+    - Type of object DB2. May be %TS/IX (Tablespace/Index).+
)END
```

YOUR.XXXX.WINDOWSJ.USER(DB2TCOPJ)

LRECL 50 BLKSIZE 5000.

```
)ATTR
 % TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
                   COLOR(TURQUOISE)
 @ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(TURQUOISE)
 $ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(blue)
 # TYPE(OUTPUT) INTENS(LOW)  CAPS(ON)
   HILITE(REVERSE) COLOR(white)
 ! TYPE(OUTPUT) INTENS(LOW)  CAPS(ON)
   HILITE(REVERSE) COLOR(TURQUOISE)
)BODY WINDOW(5Ø,16)
%COMMAND ===>_ZCMD
$   JOB¿JOBNAME $SUBMITED FOR RESET COPYPENDING
$-------------------------------------------------
)MODEL
!JOBCOP
)INIT
)PROC
)END
```

YOUR.XXXX.WINDOWSJ.USER(DB2TCHKJ)

LRECL 50 BLKSIZE 5000.

```
)ATTR
 % TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
                   COLOR(TURQUOISE)
 @ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(BLUE)
 $ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(BLUE)
 # TYPE(OUTPUT) INTENS(LOW)  CAPS(ON)
   HILITE(REVERSE) COLOR(WHITE)
 ! TYPE(OUTPUT) INTENS(LOW)  CAPS(ON)
   HILITE(REVERSE) COLOR(TURQUOISE)
```

```
)BODY WINDOW(5Ø,16)
%COMMAND ===>_ZCMD
$  JOB#JOBNAME $SUBMITED FOR RESET CHECKPENDING
$-------------------------------------------------
)MODEL
!JOBCHK
)INIT
)PROC
)END
```

YOUR.XXXX.WINDOWSJ.USER(DB2TRBDJ)

LRECL 50 BLKSIZE 5000.

```
)ATTR
 % TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
                   COLOR(TURQUOISE)
 @ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(BLUE)
 $ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(BLUE)
 # TYPE(OUTPUT) INTENS(LOW)  CAPS(ON)
   HILITE(REVERSE) COLOR(WHITE)
 ! TYPE(OUTPUT) INTENS(LOW)  CAPS(ON)
   HILITE(REVERSE) COLOR(TURQUOISE)
)BODY WINDOW(5Ø,16)
%COMMAND ===>_ZCMD
$  JOB#JOBNAME $SUBMITED FOR RESET REBUILDPENDING
$-------------------------------------------------
)MODEL
!JOBRBD
)INIT
)PROC
)END
```

YOUR.XXXX.WINDOWSJ.USER(DB2TRECO)

LRECL 50 BLKSIZE 5000.

```
)ATTR
 % TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
                   COLOR(TURQUOISE)
 @ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(BLUE)
 # TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(TURQUOISE)
 $ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(BLUE)
 ! TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(WHITE)
 _ TYPE(INPUT)  INTENS(HIGH) CAPS(ON)
)BODY WINDOW(37,Ø8) ASIS
$ TYPE OF RECOVER :!DATABASE!NAME
@------------------------------------
%
_E%1. TO FULL IMAGECOPY
%
%  2. TO ACTUAL POINT
%
%  3. REPAIR
)INIT
.CURSOR = E
.HELP   = DB2RECOH
)PROC
VER(&E,NONBLANK,LIST,1,2,3)
)END
```

YOUR.XXXX.WINDOWSJ.USER(DB2TREIC)

LRECL 50 BLKSIZE 5000.

```
)ATTR
 $ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(BLUE)
 ! TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(WHITE)
 % TYPE(TEXT)   INTENS(HIGH) SKIP(ON)
 + TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
 _ TYPE(INPUT)  INTENS(HIGH) CAPS(ON)
 @ TYPE(OUTPUT) INTENS(LOW)  CAPS(ON) JUST(RIGHT)
   HILITE(REVERSE) COLOR(TURQUOISE)
 ? TYPE(OUTPUT) INTENS(LOW)  CAPS(ON) JUST(LEFT)
   HILITE(REVERSE) COLOR(TURQUOISE)
)BODY WINDOW(44,18) ASIS
+
$ IMAGECOPYS OF TABLESPACE!DATABASE!NAME
$--------------------------------------------
%COMMAND ===>_ZCMD       +%SCROLL ===>_SCRO+
%
%    OPC    DATE    HOUR   VERSION
+   %--   ------  ------  -------
)MODEL
+    _V+  ?FECHA +?HORA  +@VERSION+
)INIT
 &FECHA  = '&ZDAY..&ZMONTH..&ZYEAR'
 &HORA   = &ZTIME
 &SCRO   = PAGE
```

```
)REINIT
 &HORA    = &ZTIME
)PROC
IF (&ZTDSELS > ØØØØ)
    VER(&V,LIST,R)
)END
```

YOUR.XXXX.WINDOWSJ.USER(DB2TRECJ)

LRECL 50 BLKSIZE 5000.

```
)ATTR
 % TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
                   COLOR(TURQUOISE)
 @ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(BLUE)
 $ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(BLUE)
 # TYPE(OUTPUT) INTENS(LOW)  CAPS(ON)
   HILITE(REVERSE) COLOR(WHITE)
 ! TYPE(OUTPUT) INTENS(LOW)  CAPS(ON)
   HILITE(REVERSE) COLOR(TURQUOISE)
)BODY WINDOW(5Ø,16) ASIS
%COMMAND ===>_ZCMD
$ JOB#JOBNAME $SUBMITED FOR RESET RECOVERYPENDING
$-------------------------------------------------
)MODEL
!JOBREC
)INIT
)PROC
)END
```

YOUR.XXXX.WINDOWSJ.USER(DB2TREPJ)

LRECL 50 BLKSIZE 5000.

```
)ATTR
 % TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
                   COLOR(TURQUOISE)
 @ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(BLUE)
 $ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
   HILITE(REVERSE) COLOR(BLUE)
 # TYPE(OUTPUT) INTENS(LOW)  CAPS(ON)
   HILITE(REVERSE) COLOR(WHITE)
 ! TYPE(OUTPUT) INTENS(LOW)  CAPS(ON)
   HILITE(REVERSE) COLOR(TURQUOISE)
)BODY WINDOW(5Ø,16) ASIS
```

```
%COMMAND ===>_ZCMD
$ JOB#JOBNAME $SUBMITED FOR REPAIR RECOVERYPENDING
$--------------------------------------------------
)MODEL
!JOBREP
)INIT
)PROC
)END
```

YOUR.XXXX.MSGLIB(DBC00)

```
DBCØØ1   'FAILURE OF SYSIN ALLOC'    .ALARM=YES
'CHECK FOR OUT OF SPACE IN SYSDA/RACF PERMISSIONS'
DBCØØ2   'FAILURE OF SYSREC ALLOC'   .ALARM=YES
'CHECK FOR OUT OF SPACE IN SYSDA/RACF PERMISSIONS'
DBCØØ3   'FAILURE OF SYSPRINT ALLOC' .ALARM=YES
'CHECK FOR OUT OF SPACE IN SYSDA/RACF PERMISSIONS'
```

YOUR.XXXX.MSGLIB(DBC02)

```
DBCØ21   'FAILURE OF JOB ALLOC'    .ALARM=YES
'CHECK FOR OUT OF SPACE IN SYSDA/RACF PERMISSIONS'
DBCØ22   'FAILURE WITH JOB SUBMIT' .ALARM=YES
'FAILURE WITH JOB SUBMIT FROM A REXX . CALL TO SUPPORT'
```

YOUR.XXXX.MSGLIB(DBC04)

```
DBCØ41   'REXX LOG NOT ACCESSIBLE' .ALARM=YES
'FAILURE IN ACCESS/DEFINE THE REXX LOG.'
```

YOUR.XXXX.MSGLIB(DBC20)

```
DBC2ØØ   'CONNECTION DB2 FAULT'     .ALARM=YES  .WINDOW=LNORESP
'THE CONNECTION DB2 IS NOT AVAILABLE. STARTUP THE DB2 SUBSYSTEM'
DBC2Ø6   'TERM UTIL BMC FAULT'      .ALARM=YES  .WINDOW=LNORESP
'FAILURE IN COMMAND TERM UTIL BMC (NOT IBM). CALL TO SUPPORT'
```

YOUR.XXXX.MSGLIB(DBC30)

```
DBC3ØØ   'CONNECTION DB2 FAULT'      .ALARM=YES .WINDOW=LNORESP
'THE CONNECTION DB2 IS NOT AVAILABLE. STARTUP THE DB2 SUBSYSTEM'
DBC3Ø1   'NOT FOUND TS/IX RESTRICT'  .ALARM=YES .WINDOW=LNORESP
'NOT FOUND TABLESPACES AND/OR INDEXS IN STATUS RESTRICT'
DBC3Ø2   'DISPLAY RESTRICT FAULT'    .ALARM=YES .WINDOW=LNORESP
'PLAN BCT DB2 COMMANDS WAS VICTIM IN TIMEOUT/DEADLOCK. TRY AGAIN LATER'
DBC3Ø3   'START OF TS/IX FAULT'      .ALARM=YES .WINDOW=LNORESP
```

```
'REVIEW GRANTS STARTDB ON DATABASE FOR USER. CALL TO SUPPORT'
DBC304    'TS/IX STARTED'              .ALARM=YES .WINDOW=LNORESP
'THE TABLESPACE/INDEX WAS STARTED SUCCESSFULLY'
DBC308    'FIX NOT PROCEED'            .ALARM=YES .WINDOW=LNORESP
'FOUND A UTILITY ACTIVE RELATED TO DB2 OBJECT IN RECOVERY. TRY LATER'
```

YOUR.XXXX.MSGLIB(DBC40)

```
DBC400    'FIX NOT PROCEED'            .ALARM=YES .WINDOW=LNORESP
'DB2 OBJECT IS BELONG TO DATABASE OF DB2 BASE SOFTWARE'
DBC401    'FIX NOT PROCEED'            .ALARM=YES .WINDOW=LNORESP
'DB2 OBJECT IS BELONG TO DATABASE OF QMF'
DBC402    'FIX NOT PROCEED'            .ALARM=YES .WINDOW=LNORESP
'DB2 OBJECT IS BELONG TO DATABASE OF BMC PRODUCTS'
DBC403    'STATUS NOT SUPPORTED'       .ALARM=YES .WINDOW=LNORESP
'STATUS NOT SUPPORTED IN THIS VERSION OF REXX DB2RES'
```

Carlos German Osorio Montoya
Database Administrator
BBVA Banco Continental (Peru)

Using UDB DB2 utilities to generate explain output for dynamic SQL

With more and more SQL being dynamic in nature, there is an ever-increasing need to be able to 'explain' such statements. This article discusses how to use the utilities available in the UDB DB2 environment to generate explain output for dynamic SQL. I ran all the examples on a Windows 2000 machine running UDB DB2 V8.1.

There are a couple of basic methods of obtaining explain information for a query:

- Using the EXPLAIN SQL function of the Control Center (or the Command Center). This gives you a visual representation of the access path and will also optionally populate the EXPLAIN tables. You can do this by firing up the Control

Center, right-clicking on the database name, selecting EXPLAIN SQL, and then typing in (or importing) the SQL you want to explain. If you are using the Command Center, then you need to check the *Automatically generate access plan* box under the *Command Center/Options/Access Plan* tab. You can then type in/run your SQL in the *Interactive* pane, and switch to the *Access path* pane to see the generated access path.

Both the Control Center and Command Center methods run the SQL before generating the access path. If you want just the access path information without running the SQL, then you need to use a green screen (see method 1 below), which uses the CURRENT EXPLAIN SNAPSHOT and CURRENT EXPLAIN MODE special register variables.

- If you don't have access to a Windows session, there are three ways of using commands via a green screen session to get the access path/explain information. The first method also gives you the option of not running the SQL to obtain the access path information.

GREEN SCREEN METHOD 1

There are two special register variables in UDB that control whether EXPLAIN data is obtained for dynamic SQL – namely CURRENT EXPLAIN SNAPSHOT and CURRENT EXPLAIN MODE. *Appendix K, Explain register values*, in the *SQL Reference* manual describes both of these register variables so I won't do it here. What I will do is show how they could both be set in order to obtain the EXPLAIN information. Let's first look at the CURRENT EXPLAIN SNAPSHOT special register value. Its default is NO, which means do not take a snapshot or gather EXPLAIN data. You can set it to YES or EXPLAIN, depending on whether you want to execute the query and gather the EXPLAIN data (YES), or not execute the query, but gather the EXPLAIN data (EXPLAIN). The advantage of setting it to EXPLAIN is that then you don't actually run the SQL – which can be very beneficial if it's a long-running query! The downside is that you have to switch off the special register before you can run any other SQL from your CLP session (strictly speaking, any other query will run; it’s just that you can see the result!).

You can find the value of the special register by using:

```
>db2 values current explain snapshot
```

and you can set the value of the special register by using:

```
>db2 set current explain snapshot <value>
```

where <*value*> is YES/EXPLAIN/NO.

The CURRENT EXPLAIN MODE special register determines whether you populate the EXPLAIN tables, and whether you want indexes evaluated/recommended or not. The possible values are NO/YES/EXPLAIN/RECOMMEND INDEXES/ EVALUATE INDEXES. You query and set the value of CURRENT EXPLAIN MODE in the same way as you did for CURRENT EXPLAIN SNAPSOT, namely by using the **db2 values** and the **db2 set** commands.

So if you just want to obtain the cost of a query (without populating the EXPLAIN tables), the sequence of commands would be:

a Check the value of the CURRENT EXPLAIN MODE special register.

b Check the value of the CURRENT EXPLAIN SNAPSHOT special register.

c Set the CURRENT EXPLAIN SNAPSHOT special register value to EXPLAIN.

d Run the SQL.

e Run the db2exfmt utility to get the EXPLAIN output.

f Set the special register value to NO.

This is shown below:

```
(a) >db2 values current explain mode

1
```

```
-----
NO

(b) >db2 values current explain snapshot

1
--------
NO

(c) >db2 set current explain snapshot explain

(d) >db2 select * from  sysibm.sysdummy1
SQLØ217W  The statement was not executed as only Explain information
requests are being processed.  SQLSTATE=Ø16Ø4

(e) >db2exfmt
```

Type in the database name (in our case ‘sample’), and accept the defaults for all the other prompts:

```
IBM DATABASE 2 Explain Table Format Tool
Enter Database Name ==> sample
Connecting to the Database.
Connect to Database Successful.
Enter up to 26 character Explain timestamp (Default -1) ==>
Enter up to 8 character source name (SOURCE_NAME, Default %) ==>
Enter source schema (SOURCE_SCHEMA, Default %) ==>
Enter section number (Ø for all, Default Ø) ==>
Enter outfile name. Default is to terminal ==>
IBM DATABASE 2 Explain Table Format Tool
```

I won’t show the rest of the output, but it does give you the original SQL statement, the optimized one, and the total cost of the statement.

```
(f) >db2 set current explain snapshot no
```

The above sequence of commands gives me the cost of the query, and this is all I have to do if I want just the cost of the query (because the CURRENT EXPLAIN MODE was set to NO, the explain information was not written to the EXPLAIN tables).

You only get the message ‘SQL0217W The statement was not executed as only Explain information requests are being processed. SQLSTATE=01604’ from your CLP session; if you sign on to another CLP session with the same userid, you will not get this message.

If you want to populate the EXPLAIN tables, you would follow the same steps as above, but in addition you would set the CURRENT EXPLAIN MODE special register value to EXPLAIN. This is shown below:

```
>db2 set current explain snapshot explain

>db2 set current explain mode explain

>db2 select * from sysibm.sysdummy1
SQLØ217W  The statement was not executed as only Explain information
requests are being processed.  SQLSTATE=Ø16Ø4

>db2 set current explain snapshot no

>db2 set current explain mode no
```

You can check on the contents of the EXPLAIN tables by using a query such as:

```
>db2 select int(a.total_cost), int(a.io_cost),
int(a.cpu_cost),a.operator_type, a.stmtno,a.sectno, a.explain_time,
substr(b.statement_text,1,7Ø) from explain_operator a ,
explain_statement b where a.explain_time = b.explain_time and
b.statement_text like 'SELECT%'

1           2           3           OPERATOR_TYPE STMTNO      SECTNO
EXPLAIN_TIME

          Ø           Ø         12Ø RETURN                  1
2Ø1 2ØØ2-11-28-22.Ø5.1Ø.184ØØ1 SELECT 'Y' AS "IBMREQD" FROM (SELECT 'Y'
FROM (VALUES 1) AS
 Q1) AS Q2
          Ø           Ø         12Ø TBSCAN                  1
2Ø1 2ØØ2-11-28-22.Ø5.1Ø.184ØØ1 SELECT 'Y' AS "IBMREQD" FROM (SELECT 'Y'
FROM (VALUES 1) AS
 Q1) AS Q2
```

Column 1 gives you the total cost, and the last column gives you the SQL that was explained. I have added columns for io_cost and cpu_cost because they may be useful in comparing the costs of various SQL statements.

Once the EXPLAIN tables have been populated, you can use the visual explain feature of the Control Center to get a graphical representation of the access plan. Fire up the Control Center, right-click on the database name (‘sample’ in our case), and select *Show Explained Statements history*. You will then get a list of statements that have been explained and are in the EXPLAIN tables. Simply click once on the statement you want to see, then select the *Statements* tab and click on *Show Access Plan*.

GREEN SCREEN METHOD 2

If you want to see a green screen access graph, you have to use the db2expln utility, as shown below. You need to specify the database name (**-d**), the fact that you want the output to go to the terminal (**-t**), and the SQL statement that you want to see the access path for (**-q**). You do not need to set either of the two special register values for this utility to work, and it does not store the explain information in the EXPLAIN tables. In fact, you don't even need to be connected to the database (that is what the **-d** parameter does for you!). I have used a different query as an example, to show the format of the access path:

```
>db2expln -d sample -t -q "select empno from employee where workdept =
'AØØ'"

        Isolation Level          = Cursor Stability
        Blocking                 = Block Unambiguous Cursors
        Query Optimization Class = 5

        Partition Parallel       = No
        Intra-Partition Parallel = No

SQL Path                 = "SYSIBM", "SYSFUN", "SYSPROC", "DB2ADMIN"

SQL Statement:

select empno   from employee   where workdept = 'AØØ'

Estimated Cost        = 5Ø.118404
Estimated Cardinality = 3.Ø8ØØØØ

Access Table Name = DB2ADMIN.EMPLOYEE  ID = 2,5
|  #Columns = 1
|  Relation Scan
|  |  Prefetch: Eligible
|  Lock Intents
|  |  Table: Intent Share
|  |  Row   : Next Key Share
|  Sargable Predicate(s)
|  |  #Predicates = 1
|  Return Data to Application
|  |  #Columns = 1
Return Data Completion

End of section
```

GREEN SCREEN METHOD 3

Perhaps a quicker method of populating the EXPLAIN tables is to use the EXPLAIN ALL WITH SNAPSHOT command – I guess this sets the special register values to YES for both under the covers! I don't think the command is as flexible as setting the special registers yourself, but if you don't need the index advisor part, then it is a quicker option than setting the special register values manually. It's invoked as follows:

```
>db2 explain all with snapshot for "SELECT * from sysibm.sysdummy1"
```

You can then select from the EXPLAIN tables as discussed previously.

If you are using summary tables, then these can be used only with dynamic SQL. I would therefore use the **db2 explain all with snapshot for ...** command to do the explain, because other utilities produce a temporary package containing the SQL, which would be static SQL and therefore not compatible with summary tables.

C Leonard
Freelance Consultant (UK)

DB2 news

Software Engineering has announced RT-ExtentManager, which eliminates the problem of exploding DB2 spaces caused by unexpected amounts of data or unknown storage sizes needed by table spaces, for example those belonging to ERP systems like SAP.

By monitoring extents, RT-ExtentManager can automatically eliminate space problems by intelligent sizing of the secondary quantity of DB2 objects in real-time, even with a LOAD job running in parallel.

For further information contact:
Software Engineering, Robert-Stolz-Str 5, D-40470 Duesseldorf, Germany.
Tel: +49 (211) 96149 620
SEGUS Inc, 12007 Sunrise Valley Dr, Reston, VA 20191, USA.
Tel: (703) 391-9650.
URL: http://www.segus.com/RealTimeMaintain.htm.

* * *

LSI Logic Storage Systems has announced a new back-up and recovery solution in its LogicStor Solutions Program that combines snapshot replication with IBM Tivoli Storage Manager to support IBM DB2 UDB.

Powered by LSI Logic Storage Systems SANtricity Snapshot, the LogicStor Backup and Recovery Solution provides instant back-up and recovery. SANtricity Snapshot enables a point-in-time copy of storage volumes that are then used by back-up software, such as IBM Tivoli Storage Manager, to nearly eliminate the back-up window and provide rapid recovery for DB2 users.

LSI Logic Storage Systems, 1621 Barber Lane, Milpitas, CA 95035, USA.
Tel: (408) 433-8000.
URL: http://www.lsilogic.com/products/storage_systems/index.html.

* * *

Lakeview Technology has announced the release of OmniReplicator Version 4.0.1.

OmniReplicator is a real-time, changed data replication solution that enables companies to move information across the most popular relational databases, including DB2, Informix, Oracle, MS SQL Server, and Sybase. It supports DB2 UDB on Unix, Linux, and Windows, and its streamlined model management process, incremental commit, were developed to meet increasing market needs.

Version 4.0.1 also includes incremental commit, which minimizes downtime by reducing the number of affected tables to only those that have had a change since the last commit.

Lakeview Technology, 1901 South Meyers Rd, Suite 600, Oakbrook Terrace, IL 60181 USA.
Tel: (630) 282 8100.
URL: http://www.omnireplicator.com/latestnews/pr_2003_10_20.asp.

* * *