75

DB2 update

January 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 £245.00 in the UK; $365.00 in the USA and Canada; £251.00 in Europe; £257.00 in Australasia and Japan; and £255.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1994 issue, are available separately to subscribers for £21.00 ($31.00) each including postage.

Printed in England.

SUPERCE – an alternative to REXX

INTRODUCTION

Numerous articles in *DB2 Update* have described tools for checking output from a utility based around REXX, so that further processing may be driven conditionally following the generation of certain messages.

While REXX is a very useful tool, there are other ways of achieving this that can turn out to be easier to implement, interpret, or tailor – especially if the interpretation or tailoring has to be done by an operations analyst or operator in the early hours, while on the telephone to a DBA at home.

SUPERCE is well known as an on-line comparison and search-for tool, but it does not seem to be widely used within batch procedures for checking output – unlike the functionally similar GREP and DIFF commands in Unix.

I have used SUPERCE search-for batch job steps to test for text strings within output produced by utilities, compilers, etc. It has the advantage of being a standard utility and not requiring any special code to run. An experienced operations analyst can easily work out what a job step is doing, without needing to be an expert in REXX.

RETURN CODES PRODUCED BY SUPERCE SEARCH-FOR

SUPERCE search-for, or ISRSUPC, returns a value in register 15 which can be tested with conditional logic in JCL procedures (or with the older COND parameter). The values that I have generally used are:

- ‘00’ – indicating that the character string was not found in the dataset defined on the NEWDD DD-statement.
- ‘01’ – indicating that the given character string was found.

Another useful return code is ‘16’, which is returned when the PDS member referred to in the NEWDD DD-statement does not exist, although the PDS itself does (ISRSUPC intercepts the 013-18 abend in these circumstances).

AN EXAMPLE – START-OF-DAY CHECKS

The following is an example of a job I have implemented to run every morning prior to the start of on-line TP services.

The general principle behind the job is to run DB2 DISPLAY commands in separate steps: routing the output to temporary datasets; printing the temporary datasets; then searching the output for a character string indicating that the tablespaces are all available for on-line use – no utilities are still running, and the required database and its tablespaces have no restrictions that would prevent updates or other required functionality.

If the required text is not found, a further step is executed to force the job to abend – this is then detected by the scheduler which can alert an operations analyst, who in turn can seek advice from a DBA.

Alternatively, an IKJEFT01 step could be included to XMIT or SEND a message to relevant recipients, or even submit a further job – perhaps to stop relevant on-line transactions and load a sign-on message on the TP system to alert users to the non-availability of parts of the service.

This is all done within JCL, testing for return codes – no code has to be written, or maintained.

Of course, if the message text returned by DB2 in a future release changes, the input to ISRSUPC will also have to be changed.

ISRSUPC

```
//       JOB (ACCT#),'SEL',NOTIFY=xxxxx,
//         CLASS=A,MSGCLASS=X
//*
//*
//JOBLIB    DD DSN=xxxx.SDSNLOAD,DISP=SHR
//********************************************
//* DB2 DISPLAY / CHECK MESSSAGES JOB (DB2T)
//********************************************
//DISDBRES EXEC PGM=IKJEFTØ1,
//         DYNAMNBR=2Ø,
//         COND=(4,LT)
//SYSTSPRT DD  DISP=(NEW,PASS),DSN=&&DISDBRES,SPACE=(TRK,(1,1)),
//            UNIT=WORK,DCB=(RECFM=VBA,LRECL=133,BLKSIZE=Ø)
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
```

```
//SYSTSIN  DD  *
  DSN SYSTEM(DB2T)
  -DIS DB(JAADBØ1) LIMIT(*) RESTRICT
  END
//DISSPRES EXEC PGM=IKJEFTØ1,
//         DYNAMNBR=2Ø,
//         COND=(4,LT)
//SYSTSPRT DD  DISP=(NEW,PASS),DSN=&&DISSPRES,SPACE=(TRK,(1,1)),
//            UNIT=WORK,DCB=(RECFM=VBA,LRECL=133,BLKSIZE=Ø)
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(DB2T)
  -DIS DB(JAADBØ1) SPACENAM(*) LIMIT(*) RESTRICT
  END
//DISUTIL  EXEC PGM=IKJEFTØ1,
//         DYNAMNBR=2Ø,
//         COND=(4,LT)
//SYSTSPRT DD  DISP=(NEW,PASS),DSN=&&DISUTIL,SPACE=(TRK,(1,1)),
//            UNIT=WORK,DCB=(RECFM=VBA,LRECL=133,BLKSIZE=Ø)
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(DB2T)
  -DIS UTIL(*)
  END
//DISTHD   EXEC PGM=IKJEFTØ1,
//         DYNAMNBR=2Ø,
//         COND=(4,LT)
//SYSTSPRT DD  DISP=(NEW,PASS),DSN=&&DISTHD,SPACE=(TRK,(1,1)),
//            UNIT=WORK,DCB=(RECFM=VBA,LRECL=133,BLKSIZE=Ø)
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(DB2T)
  -DIS THREAD(*)
  END
//*
//PRINT    EXEC PGM=IEBGENER
//SYSIN    DD   DUMMY
//SYSPRINT DD   SYSOUT=*
//SYSUT2   DD   SYSOUT=*
//SYSUT1   DD   DISP=(OLD,PASS),DSN=&&DISDBRES
//         DD   DISP=(OLD,PASS),DSN=&&DISSPRES
//         DD   DISP=(OLD,PASS),DSN=&&DISUTIL
//         DD   DISP=(OLD,PASS),DSN=&&DISTHD
//*
//* CHECK THE DIS DB OUTPUT - RC 1 => DATABASE OK
//*                             Ø => DATABASE NOT OK
//DATABASE EXEC PGM=ISRSUPC,PARM=(SRCHCMP,'')
```

```
//NEWDD  DD DSN=&&DISDBRES,DISP=(OLD,DELETE)
//OUTDD  DD SYSOUT=(*)
//SYSIN  DD *
SRCHFOR  'NOT RESTRICTED'
/*
//IFA    IF DATABASE.RC=Ø
//       THEN                 /* DATABASE NOT READY */
//       EXEC PGM=NOTFOUND    /* FORCE S8Ø6 ABEND   */
//ENDIFA ENDIF
//*
//* CHECK THE DIS DB SPACENAM OUTPUT - RC 1 => TABLESPACES OK
//*                                       Ø => TABLESPACES NOT OK
//TBLSPACE EXEC PGM=ISRSUPC,
//            PARM=(SRCHCMP,
//            '')
//NEWDD  DD DSN=&&DISSPRES,DISP=(OLD,DELETE)
//OUTDD  DD SYSOUT=(*)
//SYSIN  DD *
SRCHFOR  'NO SPACES FOUND'
/*
//IFB    IF TBLSPACE.RC=Ø
//       THEN                 /* TABLESPACES NOT READY */
//       EXEC PGM=NOTFOUND    /* FORCE S8Ø6 ABEND   */
//ENDIFB ENDIF
//*
//* CHECK THE DIS UTIL - RC 1 => NO UTILS RUNNING
//*                         Ø => UTILS STILL RUNNING OR FAILED
//UTILITY  EXEC PGM=ISRSUPC,
//            PARM=(SRCHCMP,
//            '')
//NEWDD  DD DSN=&&DISUTIL,DISP=(OLD,DELETE)
//OUTDD  DD SYSOUT=(*)
//SYSIN  DD *
SRCHFOR  'NO AUTHORIZED UTILITY'
/*
//IFC    IF UTILITY.RC=Ø
//       THEN                 /* UTILITY STILL RUNNING */
//       EXEC PGM=NOTFOUND    /* FORCE S8Ø6 ABEND   */
//ENDIFC ENDIF
```

OTHER POTENTIAL APPLICATIONS

This approach can also be used to simplify compile procedures. Instead of having one job for DB2 programs and another for non-DB2 programs, you can treat all as DB2 programs, and run them through the precompiler. The precompile step is followed by an ISRSUPC to search for the characters ‘DBRM’ in the DBRM produced by the precompiler. If it is not found, the program did not include any EXEC SQL statements and so the DBRM is empty and can be deleted. If a BIND PACKAGE step follows later in the job, that too can be conditional upon whether or not a valid DBRM was generated by the precompiler.

It is probable that a use could also be found in many other routine processes where specific text strings are the only indication of a potential problem.

Phil Button
Consultant (UK)

PLAN and PACKAGE management – part 3

This month we conclude the article giving procedures to build libraries and jobs to REBIND a full project, or BIND a new project starting from an older one.

MDB2027 EDIT MACRO

```
/* REXX */
isredit macro
isredit exclude '''SUPERC -'''                 4  all
isredit exclude '''NEW: '''                    2  all
isredit exclude '''LISTING OUTPUT SECTION'''  23  all
isredit exclude '''ID       SOURCE LINES'''    2  all
isredit exclude '''——+——'''                    6  all
isredit exclude '''        '''                 6  all
isredit delete x all
isredit sort
isredit save
isredit end
```

SAMPLE BATCH SUBMIT PROCEDURE

```
//???????J JOB (account),'SAMPLE JOB',CLASS=?,MSGCLASS=X,
//       REGION=3M,MSGLEVEL=(1,1),NOTIFY=&SYSUID
/*JOBPARM BYTES=999999,LINES=999999
```

```
//* ------------------------------------------------- *
//* -    * Parameter List description  * Default *  - *
//* - +-------------------------------------------+ - *
//* -     DB2 Subsystem name ---:   no default      - *
//* -     Bind extractor type ---:   no default     - *
//* -     Creator/Owner  -----:   no default        - *
//* -     Hi-level work areas ---:   USERID         - *
//* -     Esoteric unit name ---:   WORKA           - *
//* -     Account job name ----:   no default       - *
//* - +-------------------------------------------+ - *
//* ---------------------------------------------- *
//JOBLIB     DD  DISP=SHR,DSN=SYS1.DSN310.SDSNLOAD
//           DD  DISP=SHR,DSN=PLI.V2R3M0.PLILINK
//           DD  DISP=SHR,DSN=PLI.V2R3M0.SIBMLINK
//* ---------------------------------------------- *
//DELVIEW  EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SYSTSIN   DD  *
 DSN SYSTEM(DSNZ)
 RUN PROGRAM(DSNTEP2) PLAN(DSNTEP31) LIB('DSN310.RUNLIB.LOAD')
//SYSIN    DD  *
DROP VIEW VBPPACK   ;
DROP VIEW VBPACK    ;
DROP VIEW VBPMEMB   ;
//* ------------------------ *
//LAB0     IF (DELVIEW.RC LT 9) THEN
//CREVIEW  EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SYSTSIN   DD  *
 DSN SYSTEM(DSNZ)
 RUN PROGRAM(DSNTEP2) PLAN(DSNTEP31) LIB('DSN310.RUNLIB.LOAD')
//SYSIN    DD  *

   CREATE VIEW VBPPACK
   (SEQ,PLAN,PACKAGE,COLLECT,CREAT,QUAL,VL,AQ,IS,RE,EX) AS
   SELECT  B.SEQNO,A.NAME,B.NAME,B.COLLID,A.CREATOR,A.QUALIFIER,
           VALIDATE,A.ACQUIRE,A.ISOLATION,A.RELEASE,A.EXPLAN
           FROM SYSIBM.SYSPLAN      A ,
                SYSIBM.SYSPACKLIST B
           WHERE B.COLLID   LIKE '%'        AND
                 A.NAME     LIKE '%'        AND
                 A.CREATOR  =    'DBA000'  AND
                 A.NAME     =    B.PLANNAME ;

   CREATE VIEW VBPACK
   (PACKAGE,COLLECT,OWNER,QUAL,VL,EX,PDSNA,IS,RE) AS
   SELECT  A.NAME,A.COLLID,A.OWNER,A.QUALIFIER,
           A.VALIDATE,A.EXPLAIN,A.PDSNAME,A.ISOLATION,A.RELEASE
           FROM SYSIBM.SYSPACKAGE   A
           WHERE A.COLLID   LIKE '%'        AND
                 A.NAME     LIKE '%'        AND
                 A.OWNER    =    'DBAØØØ'  ;

   CREATE VIEW VBPMEMB
   (PLAN,DBRM,CREAT,QUA,VL,AQ,IS,RE,EX,PDSNA) AS
   SELECT  A.NAME,B.NAME,A.CREATOR,A.QUALIFIER,A.VALIDATE,ACQUIRE,
           A.ISOLATION,A.RELEASE,A.EXPLAN,B.PDSNAME
           FROM SYSIBM.SYSPLAN     A ,
                SYSIBM.SYSDBRM     B
           WHERE A.NAME     LIKE '%'        AND
                 A.CREATOR  =    'DBAØØØ'  AND
                 A.NAME     =     B.PLNAME  ;

//* ————————————————————————— *
//LAB1      IF (CREVIEW.RC EQ Ø) THEN
//COPYTMP EXEC PGM=IEBCOPY
//SYSPRINT  DD  SYSOUT=*
//SYSUT3    DD  UNIT=VIO,SPACE=(CYL,(5,1))
//SYSUT4    DD  UNIT=VIO,SPACE=(CYL,(5,1))
//INP1      DD  DISP=SHR,DSN=ISP.V3R5MØ.ISPTENU
//OUT1      DD  DSN=&&TENU,DISP=(,PASS),SPACE=(CYL,(1,1,1Ø)),
//          UNIT=WORKA
//SYSIN     DD *
 COPY OUTDD=OUT1,INDD=INP1
//* ————————————————————————— *
//LAB2      IF (COPYTMP.RC EQ Ø) THEN
//REXXØØ   EXEC PGM=IKJEFTØ1,DYNAMNBR=15Ø
//SYSTSPRT  DD  SYSOUT=*
//* ———>  CLIST and Macro library name  <——— *
//SYSPROC   DD  DSN=USER.CLIST.LIBRARY,DISP=SHR
//ISPLOG    DD  DUMMY
//ISPPROF   DD  DISP=(,DELETE,DELETE),SPACE=(CYL,(1,1,1Ø)),
//          DCB=(LRECL=8Ø,BLKSIZE=312Ø,RECFM=FB,DSORG=PO),DSN=&&PROF,
//          UNIT=WORKA
//ISPPLIB   DD  DISP=SHR,DSN=ISP.V3R5MØ.ISPPENU
//ISPMLIB   DD  DISP=SHR,DSN=ISR.V3R5MØ.ISRMENU
//          DD  DISP=SHR,DSN=ISP.V3R5MØ.ISPMENU
//ISPTLIB   DD  DISP=(OLD,PASS),DSN=&&TENU
//ISPSLIB   DD  DISP=SHR,DSN=ISP.V3R5MØ.ISPSLIB
//SYSTSIN   DD  *
PROF WTPMSG MSGID
 ISPSTART CMD(@DB2BIND DSNZ,YYY,DBAØØØ,*,*,account)
//LABØEND  ENDIF
//LAB1END  ENDIF
//LAB2END  ENDIF
//* ————————————————————————— *
```

DB2RNND

```
/* REXX */
trace ?o
   /*    ──────────────────────────────────────────────────────    */
   /*  +─   Rebind statement generator for DB2 v3.1   ─+          -*/
   /*-                /////////// * ///////////                   -*/
   /*-             P A R A M E T E R S               DEFAULT      -*/
   /*-  ────────────────────────────┼─────────────────────────────-*/
   /*-   - Subsys : DB2 subsystem name             | no default     -*/
   /*-   - Creat  : Creator/Owner                  | no default     -*/
   /*-   - Autosub: Submit Rebind jobs             | * = no         -*/
   /*-   - Jobname: Rebind job name                | * = Auto build -*/
   /*-   - Hiwork : HI-level work areas            | * = userid     -*/
   /*-   - Esounit: Esoteric name for allocation   | * = worka      -*/
   /*-   - Account: Name of job ACCOUNT            | no default     -*/
   /*-  ────────────────────────────┼─────────────────────────────-*/
   /*-               /////////// * ///////////                    -*/
   /*   ────────────────────────────────────────────────────────    */

blk    =
wrk    =
sisub  = no
build  = yes
user   = userid()
notif  = userid()
arg parmin
parm     = translate(parmin,' ',',')
nparm    = words(parm)
subsys   = word(parm,1)
creat    = word(parm,2)
autosub  = word(parm,3)
jn       = word(parm,4)
hiwork   = word(parm,5)
esounit  = word(parm,6)
account  = word(parm,7)

   /*───────────────────────────────────────────*/
   /*─      Test input parameters           ───*/
   /*───────────────────────────────────────────*/

 if nparm < 7 then do
    say '         '
    say '         '
    say '>>>>>>>>'
    say '>>>>>>>>  Parameter string is incomplete !!!!'
    say '>>>>>>>>          'parmin
    say '>>>>>>>>'
    say '         '
    say '         '
    exit
    end
 if autosub = '*' then
    autosub = NO
  if jn   = '*' then
     jna = REB||substr(creat,1,2)||substr(creat,6,1)||substr(creat,4,2)
  else
     jna = jn
 if hiwork = '*' then
    hiwork = userid()
 if esounit = '*' then
    esounit = worka

   /*--------------------------------------------*/
   /*-      File job rebind Plan allocation   ---*/
   /*--------------------------------------------*/

 outdsrbn= hiwork'.'subsys'.'creat'.REBIND'
 xx=outtrap(trp01.)
    address tso "delete '"outdsrbn"'"
    "alloc da('"outdsrbn"') dir(0) space(15,15) dsorg(ps)" ,
    "recfm(f,b) lrecl(80) blksize(27920) tracks ",
    "unit("esounit") new catalog f(firbn) "
 xx=outtrap(off)
 if rc > 0 then do
    do a = 1 to trp01.0
       say trp01.a
    end
    say '        '
    say '        '
    say '>>>>>>>>'
    say '>>>>>>>> "'outdsrbn'"  Allocation OK'
    say '>>>>>>>>  RC='rc'. Verify.            '
    say '>>>>>>>>'
    say '        '
    say '        '
    exit
    end
 else
    say '>>>>>>>>  'outdsrbn '          Allocation OK  '

   /*--------------------------------------------*/
   /*-      File systsinp allocation          ---*/
   /*--------------------------------------------*/

 outds1= hiwork'.'subsys'.'creat'.REBIND.SYSTSINP'
 xx=outtrap(trp02.)
    address tso "delete '"outds1"'"
    "alloc da('"outds1"') dir(0) space(5,1) dsorg(ps)" ,
    "recfm(f,b) lrecl(80) blksize(27920) tracks ",
    "unit("esounit") release new catalog f(systsinp) "
 xx=outtrap(off)
 if rc > Ø then do
    do a = 1 to trpØ2.Ø
       say trpØ2.a
    end
    say '        '
    say '        '
    say '>>>>>>>>'
    say '>>>>>>>> "'outds1'"  Allocation OK'
    say '>>>>>>>>  RC='rc'. Verify.          '
    say '>>>>>>>>'
    say '        '
    say '        '
    exit
    end
 else
    say '>>>>>>>>  'outds1 ' Allocation OK  '
sk.1='DSN SYSTEM('subsys')                                           '
sk.2='RUN PROGRAM(DSNTEP2) PLAN(DSNTEP31) LIB(''DSN31Ø.RUNLIB.LOAD'')'
sk.3='END                                                            '
 sk.Ø=3;
 "execio * diskw systsinp (stem sk. finis"
 call Pulisci
   /*--------------------------------------------*/
   /*-      Number of select to do            ---*/
   /*--------------------------------------------*/
 do i = 1 to 2
   /*-  Extract planname                      ---*/
    if i = 1 then do
       tipsel = RBPLAN
       call Allsysp
       call Allsysi
       jobw = sysin
       "alloc da('"outdsØ"') f("jobw") shr reuse"
sk.1=' SELECT * FROM V'creat'_RBPLAN ;                           '
       sk.Ø=1
       call Scrivi
       call Rundb2
       macnr = @mdb2Ø25
       call Exmacro
       call Testout
       if build = yes then do
          sisub = yes
          call Wrbplan
          say '        '
          say '>>>>>>>>'
          say '>>>>>>>>  +-------------------------------+'
          say '>>>>>>>>    Total Rebind Plan  no' e - 1
          say '>>>>>>>>  +-------------------------------+'
          say '>>>>>>>>'
          say '        '
          address tso "delete '"outdsØ"'"
          address tso "delete '"outds2"'"
          say '        '
          say '        '
          say '        '
          end
       else do
          build = yes
          say '        '
          say '>>>>>>>>'
          say '>>>>>>>> The select RBPLAN has Ø rows.    '
          say '>>>>>>>> There is no plan to rebind !!!!  '
          say '>>>>>>>>'
          say '        '
          address tso "delete '"outdsØ"'"
          address tso "delete '"outds2"'"
          end
       end
    if i = 2 then do
       tipsel = RBPACK
       call Allsysp
       call Allsysi
       jobw = sysin
       "alloc da('"outdsØ"') f("jobw") shr reuse"
sk.1=' SELECT * FROM V'creat'_RBPACK ;                          '
       sk.Ø=1
       call Scrivi
       call Rundb2
       macnr = @mdb2Ø26
       call Exmacro
       call Testout
       if build = yes then do
          sisub = yes
          call Wrbpack
          say '        '
          say '>>>>>>>>'
          say '>>>>>>>>  +-------------------------------+'
          say '>>>>>>>>    Total Rebind Package no' f - 1
          say '>>>>>>>>  +-------------------------------+'
          say '>>>>>>>>'
          say '        '
          address tso "delete '"outdsØ"'"
          address tso "delete '"outds2"'"
          say '        '
          say '        '
          say '        '
          end
       else do
```

```
         build = yes
         say '        '
         say '>>>>>>>>'
         say '>>>>>>>> The select RBPACK has 0 rows .      '
         say '>>>>>>>> There is no package to rebind !!!! '
         say '>>>>>>>>'
         say '        '
         address tso "delete '"outds0"'"
         address tso "delete '"outds2"'"
         end
      end
   end
  /*--------------------------------------------*/
  /*-      Automatic submit of Rebind job    ---*/
  /*--------------------------------------------*/
if autosub = yes then do
   if sisub = yes then do
      xx=OUTTRAP(trp08.)
         address tso "submit '"outdsrbn"'"
      xx=OUTTRAP(OFF)
      if rc > 0 then do
         do a = 1 to trp08.0
            say trp08.a
            end
         exit
         end
      mess  = substr(trp08.1,1,22)
      say '      '
      say time()
      say time() '-->> il 'mess' has been submitted..... '
      say time()
      say '      '
      say '      '
      end
   else do
      say '        '
      say '>>>>>>>>'
      say '>>>>>>>> There are no Rebind for the user 'creat '!!!!'
      say '>>>>>>>>'
      say '        '
      end
   address tso "delete '"outdsrbn"'"
   end
call Free
exit
  /*--------------------------------------------*/
  /*-      File sysin allocation             ---*/
  /*--------------------------------------------*/
Allsysi:
   outds0= hiwork'.'subsys'.'creat'.'tipsel'.SYSIN'
   xx=outtrap(trp09.)
      address tso "delete '"outds0"'"
      "alloc da('"outds0"') dir(0) space(5,1) dsorg(ps)" ,
      "recfm(f,b) lrecl(80) blksize(27920) tracks ",
      "unit("esounit") release new catalog f(sysin)"
   xx=outtrap(off)
   if rc > 0 then do
      do a = 1 to trp09.0
         say trp09.a
         end
      say '        '
      say '>>>>>>>>'
      say '>>>>>>>> "'outds0'"  Allocation OK'
      say '>>>>>>>>  RC='rc'. Verify.            '
      say '>>>>>>>>'
      say '        '
      say '        '
      exit
      end
   else do
      say '>>>>>>>>  'outds0 '    Allocation OK  '
      say '          '
      end
   return
  /*--------------------------------------------*/
  /*-      File sysprint allocation          ---*/
  /*--------------------------------------------*/
Allsysp:
   outds2= hiwork'.'subsys'.'creat'.'tipsel'.SYSPRINT'
   xx=outtrap(trp10.)
      address tso "delete '"outds2"'"
      "alloc da('"outds2"') dir(0) space(150,60) dsorg(ps)" ,
      "recfm(f,b,a) lrecl(121) blksize(1210)  tracks ",
      "unit("esounit") release new catalog f(sysprint) "
   xx=outtrap(off)
   if rc > 0 then do
      do a = 1 to trp10.0
         say trp10.a
         end
      say '        '
      say '        '
      say '>>>>>>>>'
      say '>>>>>>>> "'outds2'"  Allocation OK'
      say '>>>>>>>>  RC='rc'. Verify.          '
      say '>>>>>>>>'
      say '        '
      say '        '
      exit
      end
   else
```

```
      say '>>>>>>>>  'outds2 ' Allocation OK  '
   return
  /*--------------------------------------------*/
  /*-      Run DB2                           ---*/
  /*--------------------------------------------*/
Rundb2 :
   wrk = center(tipsel,8)
   say ' *********************************************************** '
   say '* ' i '                                       'time()'     *'
   say '*      The query 'wrk    'is running                       *'
   say '*                                                          *'
   say '*               Please Stand by ......                     *'
   say '*                                                          *'
   say ' *********************************************************** '
   say '                                                             '
   xx=outtrap(trp11.)
     address tso "ex '"outds1"'"
   xx=outtrap(off)
   if rc > Ø then do
      analisi = substr(trp11.1,1,8)
      if analisi = 'DSNE1ØØI' then do
         subs = center(subsys,1Ø)
         say '      '
         say '      '
         say '>>>>>>'
         say '>>>>>>'
         say '>>>>>> The DB2 subsystem ->'subs'<- is not active'
         say '>>>>>>'
         say '>>>>>>'
         say '      '
         say '      '
         end
      else do
         do a = 1 to trp11.Ø
            say trp11.a
            end
         end
      exit
      end
      say '>>>>>>>>  DB2 query terminated.  '
      say '       '
  /*--------------------------------------------*/
  /*-      Print output DB2 query            ---*/
  /*--------------------------------------------*/
   address tso "pr dataset('"outds2"') class(R) writer(nØz2) fcb(6) "
   return
  /*--------------------------------------------*/
  /*-    Macro to process output of sysprint  --*/
  /*--------------------------------------------*/
Exmacro:
   xx=OUTTRAP(trp12.)
      "ispexec edit dataset('"outds2"') macro("macnr")"
   xx=OUTTRAP(OFF)
   if rc > 0 then do
      do a = 1 to trp12.0
         say trp12.a
         end
      exit
      end
   return
  /*--------------------------------------------*/
  /*-      Test output DB2 query             ---*/
  /*--------------------------------------------*/
Testout:
   xx=outtrap(trp13.)
      "execio * diskr sysprint (stem sysprint. finis"
      "free fi(sysprint)"
   xx=outtrap(off)
   if rc > 0 then do
      do a = 1 to trp13.0
         say trp13.a
         end
      say '        '
      say '        '
      say '>>>>>>>>'
      say '>>>>>>>>  Error reading file "'outds2'"'
      say '>>>>>>>>  RC='rc'. Verify.             '
      say '>>>>>>>>'
      say '        '
      say '        '
      exit
      end
   if sysprint.0 = 0 then do
      say '        '
      say '        '
      say '>>>>>>>>'
      say '>>>>>>>>  The file "'outds2'" is empty.    '
      say '>>>>>>>>  Probably error accessing DB2 !!!!'
      say '>>>>>>>>'
      say '        '
      say '        '
      exit
      end
   if substr(sysprint.1,33,8) = ' 0 ROW(S)' then
      build = no
   return
  /*--------------------------------------------*/
  /*-      Build rebind Plan statement       ---*/
  /*--------------------------------------------*/
Wrbplan:
    say '>>>>>>>>  Building job of Rebind Plan '
    jobw = firbn
    "alloc da('"outdsrbn"') f("jobw") mod reuse"
    call Hdrbnd
    DO e = 1 to sysprint.Ø
       pla = word(sysprint.e,1)
       lpla = length('REBIND PLAN('''pla''')')
       wrk = 5Ø - lpla
sk.e='REBIND PLAN('pla')'copies(' ',wrk)
       end
sk.e='//* —————————————————————————————————————————————————— *        '
    sk.Ø = e
    call Scrivi
    return
   /*————————————————————————————————————————————*/
   /*—     Build rebind Package statement     ——*/
   /*————————————————————————————————————————————*/
 Wrbpack:
    say '>>>>>>>>  Building job of Rebind Package '
    jobw = firbn
    "alloc da('"outdsrbn"') f("jobw") mod reuse"
    call Hdrbnd
    DO f = 1 to sysprint.Ø
       pak = word(sysprint.f,1)
       col = word(sysprint.f,2)
       lpak = length('REBIND PACKAGE('''col''''.''''pak''')')
       wrk = 5Ø - lpak
sk.f='REBIND PACKAGE('col'.'pak')'copies(' ',wrk)
       end
sk.f='//* ——————————————————————————————————————————————————  *       '
    sk.Ø = f
    call Scrivi
    return
   /*————————————————————————————————————————————*/
   /*—      Header job BIND                   ——*/
   /*————————————————————————————————————————————*/
 Hdrbnd:
    jobw = firbn
    "alloc da('"outdsrbn"') f("jobw") mod reuse"
sk.1='//'jna' JOB ('account'),'' BIND JOB '',CLASS=S,MSGCLASS=X,      '
sk.2='//      USER='user',REGION=3M,MSGLEVEL=(1,1),NOTIFY='notif'     '
sk.3='/*JOBPARM BYTES=999999,LINES=9999                                   '
sk.4='//* —————————————————————————————————————————————————— *        '
sk.5='//*                  Rebind  JOB                       *        '
sk.6='//* —————————————————————————————————————————————————— *        '
sk.7='//JOBLIB    DD  DSN=SYS1.DSN31Ø.SDSNLOAD,DISP=SHR                   '
sk.8='//BIND  EXEC  PGM=IKJEFTØ1,DYNAMNBR=1ØØ                             '
sk.9='//SYSTSPRT  DD  SYSOUT=*                                            '
sk.1Ø='//REPORT    DD  SYSOUT=*                                           '
sk.11='//SYSTSIN   DD  *                                                  '
sk.12='DSN SYSTEM('subsys')'
    sk.Ø = 12
    call Scrivi
    return
   /*---------------------------------------------*/
   /*-      Write routine output record        ---*/
   /*---------------------------------------------*/
 Scrivi :
    "EXECIO * DISKW "jobw" (STEM sk. FINIS"
 Pulisci:
    DO g = 1 to sk.Ø
       sk.g = blk
       end
    return
   /*---------------------------------------------*/
   /*-      Free datasets                      ---*/
   /*---------------------------------------------*/
 Free    :
    "free fi(systsinp)"
     address tso "delete '"outds1"'"
     "free fi(firbn)"
    return
```

MDB2025 EDIT MACRO

```
isredit macro
isredit exclude all
isredit find '_|' all
isredit find ' Ø ROW'
isredit delete x all
isredit change P'=====' '     ' 59 all
isredit change '|'  ' '  all
isredit save
isredit end
```

MDB2026 EDIT MACRO

```
isredit macro
isredit exclude all
isredit find '_|' all
isredit find ' Ø ROW'
isredit delete x all
isredit change P'=====' '     ' 49 all
isredit change '|'  ' '  all
isredit save
isredit end
```

SAMPLE BATCH SUBMIT PROCEDURE

```
//???????J JOB (account),'SAMPLE JOB',CLASS=?,MSGCLASS=X,
//       REGION=3M,MSGLEVEL=(1,1),NOTIFY=&SYSUID
/*JOBPARM BYTES=999999,LINES=999999
//*  -------------------------------------------------  *
//* -    * Parameter List description  * Default *    - *
//* -  +--------------------------------------------+  - *
//* -     DB2 Subsystem name ---:       no default    - *
//* -     Creator/Owner ------:         no default    - *
//* -     Submit job -------:           * = no        - *
//* -     Rebind job name -----:        * = Auto Build- *
//* -     Hi-level work areas ---:      USERID        - *
//* -     Esoteric unit name ---:       WORKA         - *
//* -     Account job name ----:        no default    - *
//* -  +--------------------------------------------+  - *
//*  -------------------------------------------------  *
//JOBLIB    DD  DSN=PLI.V2R3M0.PLILINK,DISP=SHR
//          DD  DSN=PLI.V2R3M0.SIBMLINK,DISP=SHR
//          DD  DSN=SYS1.DSN310.SDSNLOAD,DISP=SHR
//* ------------------------- *
//DELVIEW  EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SYSTSIN   DD  *
 DSN SYSTEM(DSNZ)
 RUN PROGRAM(DSNTEP2) PLAN(DSNTEP31) LIB('DSN310.RUNLIB.LOAD')
//SYSIN    DD  *
DROP VIEW VDBA000_RBPLAN   ;
DROP VIEW VDBA000_RBPACK   ;
//* ------------------------- *
//LAB0     IF (DELVIEW.RC LT 9) THEN
//CREVIEW  EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SYSTSIN   DD  *
 DSN SYSTEM(DSNZ)
 RUN PROGRAM(DSNTEP2) PLAN(DSNTEP31) LIB('DSN310.RUNLIB.LOAD')
//SYSIN    DD  *

   CREATE VIEW VDBA000_RBPLAN (PLAN) AS
   SELECT  NAME
           FROM SYSIBM.SYSPLAN
           WHERE NAME     LIKE '%'    AND
                 CREATOR  LIKE 'DBA000%' ;

   CREATE VIEW VDBA000_RBPACK (PACKAGE,COLLECT) AS
   SELECT  NAME,COLLID
           FROM SYSIBM.SYSPACKAGE
           WHERE COLLID   LIKE '%'    AND
```

```
                 NAME     LIKE '%'    AND
                 OWNER    LIKE 'DBAØØØ%' ;
//* -------------------------------------------------- *
//* --         VALID    =    'N'    AND
//* -------------------------------------------------- *
//LAB1      IF (CREVIEW.RC EQ Ø) THEN
//COPYTMP EXEC PGM=IEBCOPY
//SYSPRINT  DD  SYSOUT=*
//SYSUT3    DD  UNIT=VIO,SPACE=(CYL,(5,1))
//SYSUT4    DD  UNIT=VIO,SPACE=(CYL,(5,1))
//INP1      DD  DISP=SHR,DSN=ISP.V3R5MØ.ISPTENU
//OUT1      DD  DSN=&&TENU,DISP=(,PASS),SPACE=(CYL,(1,1,1Ø)),
//          UNIT=WORKA
//SYSIN     DD *
 COPY OUTDD=OUT1,INDD=INP1
//* -------------------------------------------------- *
//LAB2      IF (COPYTMP.RC EQ Ø) THEN
//REXXØØ   EXEC PGM=IKJEFTØ1,DYNAMNBR=15Ø
//SYSTSPRT  DD  SYSOUT=*
//* ----->  Clist and Macro library name  <---- *
//SYSPROC   DD  DSN=USER.CLIST.LIBRARY,DISP=SHR
//ISPLOG    DD  DUMMY
//ISPPROF   DD  DISP=(,DELETE,DELETE),UNIT=WORKA,SPACE=(CYL,(1,1,1Ø)),
//          DCB=(LRECL=8Ø,BLKSIZE=312Ø,RECFM=FB,DSORG=PO),DSN=&&PROF
//ISPPLIB   DD  DISP=SHR,DSN=ISP.V3R5MØ.ISPPENU
//ISPMLIB   DD  DISP=SHR,DSN=ISR.V3R5MØ.ISRMENU
//          DD  DISP=SHR,DSN=ISP.V3R5MØ.ISPMENU
//ISPTLIB   DD  DISP=(OLD,PASS),DSN=&&TENU
//ISPSLIB   DD  DISP=SHR,DSN=ISP.V3R5MØ.ISPSLIB
//SYSTSIN   DD  *
PROF WTPMSG MSGID
 ISPSTART CMD(@DB2RBND DSNZ,DBSØØØ,*,*,*,*,account)
//LABØEND  ENDIF
//LAB1END  ENDIF
//LAB2END  ENDIF
//* -------------------------------------------------- *
```

Giuseppe Rendano
DB2 System Programmer (Italy)

Code published in *DB2 Update* is available from our Web site,www.xephon.com. Once you have registered, you can select an article containing code that you want e-mailed to you. Remember to have your copy of the issue containing the article with you when you access our Web site.

DB2 utility services – part 2

This month we continue the DB2 services that provide an easy way to execute any DB2 utility.

```
   Call Error 'ddate'
end
if dage=' '
then del='D'||right(ddate,8)
else del='A'||right(dage,8)
if dage=' ' then ddated=substr(ddate,3)
parm=substr(dbnc,1,8)||substr(tsnc,1,8)||del
messg = "Accessing  db2 system "db2""
messg = time() || " " || messg
Call Send_messg
messg = 'Select     syscopy        information'
messg = time() || " " || messg
Call Send_messg
ADDRESS TSO
QUEUE "RUN PROGRAM("program") PLAN("plan"),
       LIBRARY ('"llib"'),
       PARMS ('/"parm"')"
QUEUE "END "
"DSN SYSTEM("db2")"
if rc=12 then do
   "delstack"
   Call Free_proc
   Call Aloc
   address ispexec 'tbend messdb'
   Call Create_messg
   message = 'Error.   'db2||' ssid is not valid  |'
   Call Error 'db2'
END
"EXECIO * DISKR SYSPRINT (STEM ROW."
if substr(row.1,2) = 'NO CATALOG ENTRIES FOUND' then do
   Call Free_proc
   Call Aloc
   address ispexec 'tbend messdb'
   Call Create_messg
   message = 'No catalog entries found, check Search Fields.'
   Call Error 'dbnc'
end
else do
   address ispexec 'addpop row(1) column(5)'
   address ispexec 'tbcreate "mlist" names(detail db ts icd dsn)'
   address ispexec 'tbcreate "dlist" names(db ts)'
   count=Ø
   num=row.Ø
   db1=' '
   ts1=' '
   do i=1 to row.Ø
      count=count+1
      db = strip(substr(row.i,2,8))
      ts = strip(word(row.i,2))
      icd= word(row.i,3)
      dsn= strip(word(row.i,4))
      detail=right(count,3)||' '||left(db,9)||,
             left(ts,9)||icd||' '||dsn
      if db1=db & ts1=ts then nop
      else address ispexec 'tbadd "dlist"'
      db1=db
      ts1=ts
      address ispexec 'tbadd "mlist"'
   end
   address ispexec 'tbtop "mlist"'
   address ispexec 'tbtop "dlist"'
   address ispexec 'tbdispl "mlist" panel(dbmlist)'
   if rc=8 then do
      Call Free_proc
      address ispexec 'tbend "mlist"'
      address ispexec 'tbend "dlist"'
      Call Aloc
      address ispexec rempop all
      address ispexec 'tbend messdb'
      Call Create_messg
      signal top
   end
end
messg = 'Building a modify recovery job'
messg = time() || " " || messg
Call Send_messg
Call Free_proc
/* JCL Skeleton DB2 Modify Recovery    */
address ispexec rempop all
title = 'MODIFY RECOVERY UTILITY'
date=date()
time=time(c)
user=userid()
tempfile=userid()||'.UTIL.MODIFY'
address tso
"delete '"tempfile"'"
"free dsname('"tempfile"')"
"free ddname(ispfile)"
"free attrlist(formfile)"
"attrib formfile blksize(8ØØ) lrecl(8Ø) recfm(f b) dsorg(ps)"
"alloc ddname(ispfile) dsname('"tempfile"')",
       "new using (formfile) unit(339Ø) space(1 1) cylinders"
address ispexec
```

```
 "ftopen"
 "ftincl DBMODI"
 "ftclose"
 zedsmsg = "JCL shown"
 zedlmsg = "JCL DB2 Modify shown"
 "setmsg msg(isrzØØ1)"
 "edit dataset('"tempfile"')"
 address ispexec 'tbend "mlist"'
 address ispexec 'tbend "dlist"'
 address ispexec  "tbclose "messdb""
 exit
 Aloc:
   ADDRESS TSO "DELETE '"SYSVAR(SYSUID)".UTIL.DBUT'"
   "ALLOC DD(SYSPRINT) DSN('"SYSVAR(SYSUID)".UTIL.DBUT') SPACE(24 8),
   TRACK MOD UNIT(339Ø) RECFM(F,B) LRECL(8Ø) BLKSIZE(8ØØ) ,
   F(SYSPRINT) CATALOG REUSE "
 Return
 Error:
   ARG cur_par
   cur=cur_par
   address ispexec "setmsg msg(dbutØØ1)"
   signal top
 Return
 Free_proc:
   "execio Ø diskr sysprint (finis"
   address tso "free f(sysprint)"
 Return
 Create_messg:
  messg = "s"||userid()
  address ispexec "tbcreate "messdb" names(messg) write replace"
 Return
 Send_messg:
  address ispexec "tbadd " messdb
  address ispexec "control display lock "
  address ispexec "addpop row(13) column(6)"
  address ispexec "tbdispl "messdb" panel(dbutum)"
  address ispexec rempop
 Return
```

DBSTO

```
/* REXX */
/* STOSPACE: Build a stospace job                */
/* trace r */
 zpfctl = 'OFF'
 Y=MSG("OFF")
 /*****************************************************/
 /* Change to your convention standards              */
 program = 'PDBSTOS'
```

```
plan    = 'PDBSTOS'
llib    = 'SKUPNI.BATCH.LOADLIB'
/*****************************************************/
address ispexec 'vput (zpfctl) profile'
Call Aloc
cur='scre'
Call Create_messg
TOP:
address ispexec "display panel(dbstom) cursor("CUR")"
if rc=8 then do
   Call Free_proc
   address ispexec "tbclose "messdb""
   exit
end
/* Check input parameters                            */
if sga='YES' | sga='NO' then nop
else do
   message='Invalid All parameter. Valid values are: YES, NO.'
   Call Error 'sga'
end
if scre=' ' & sgn=' ' then do
   message='At least one Catalog search field must be entered.'
   Call Error 'scre'
end
parm=substr(scre,1,8)||substr(sgn,1,8)||substr(sga,1,3)
messg = "Accessing  db2 system "db2""
messg = time() || " " || messg
Call Send_messg
messg = 'Select     sysstogroup    information'
messg = time() || " " || messg
Call Send_messg
ADDRESS TSO
QUEUE "RUN PROGRAM("program") PLAN("plan"),
       LIBRARY ('"llib"'),
       PARMS ('/"parm"')"
QUEUE "END "
"DSN SYSTEM("db2")"
if rc=12 then do
   "delstack"
   Call Free_proc
   Call Aloc
   address ispexec 'tbend messdb'
   Call Create_messg
   message = 'Error.   'db2||' ssid is not valid  |'
   Call Error 'db2'
END
"EXECIO * DISKR SYSPRINT (STEM ROW."
if substr(row.1,2) = 'NO CATALOG ENTRIES FOUND' then do
   Call Free_proc
   Call Aloc
   address ispexec 'tbend messdb'
   Call Create_messg
   message = 'No catalog entries found, check Search Fields.'
   Call Error 'dbnc'
end
else do
   address ispexec 'tbcreate "slist" names(cr sgname)'
   num=row.0
   do i=1 to row.0
      cr     = strip(substr(row.i,2,8))
      sgname = strip(word(row.i,2))
      address ispexec 'tbadd "slist"'
   end
   address ispexec 'tbtop "slist"'
   if sga ='NO' then do
      address ispexec 'addpop row(1) column(5)'
      address ispexec 'tbdispl "slist" panel(dbslist)'
   end
   if rc=8 then do
      Call Free_proc
      address ispexec 'tbend "slist"'
      Call Aloc
      address ispexec rempop all
      address ispexec 'tbend messdb'
      Call Create_messg
      signal top
   end
end
if sga ='NO' then  address ispexec rempop all
messg = 'Building a stospace job'
messg = time() || " " || messg
Call Send_messg
Call Free_proc
/* JCL Skeleton DB2 Stospace    */
title = 'STOSPACE UTILITY'
date=date()
time=time(c)
user=userid()
tempfile=userid()||'.UTIL.STOSPACE'
address tso
"delete '"tempfile"'"
"free dsname('"tempfile"')"
"free ddname(ispfile)"
"free attrlist(formfile)"
"attrib formfile blksize(800) lrecl(80) recfm(f b) dsorg(ps)"
"alloc ddname(ispfile) dsname('"tempfile"')",
       "new using (formfile) unit(3390) space(1 1) cylinders"
address ispexec
"ftopen"
"ftincl DBSTOS"
```

```
 "ftclose"
 zedsmsg = "JCL shown"
 zedlmsg = "JCL DB2 Stospace shown"
 "setmsg msg(isrz001)"
 "edit dataset('"tempfile"')"
 address ispexec 'tbend "slist"'
 address ispexec  "tbclose "messdb""
 exit
 Aloc:
   ADDRESS TSO "DELETE '"SYSVAR(SYSUID)".UTIL.DBUT'"
   "ALLOC DD(SYSPRINT) DSN('"SYSVAR(SYSUID)".UTIL.DBUT') SPACE(24 8),
   TRACK MOD UNIT(3390) RECFM(F,B) LRECL(80) BLKSIZE(800) ,
   F(SYSPRINT) CATALOG REUSE "
 Return
 Error:
   ARG cur_par
   cur=cur_par
   address ispexec "setmsg msg(dbut001)"
   signal top
 Return
 Free_proc:
   "execio 0 diskr sysprint (finis"
   address tso "free f(sysprint)"
 Return
 Create_messg:
  messg = "s"||userid()
  address ispexec "tbcreate "messdb" names(messg) write replace"
 Return
 Send_messg:
  address ispexec "tbadd " messdb
  address ispexec "control display lock "
  address ispexec "addpop row(13) column(6)"
  address ispexec "tbdispl "messdb" panel(dbutum)"
  address ispexec rempop
 Return
```

DBUTILM

```
)attr default(%+_)
   [ type (output) intens(low)  color(green)  caps(off)
   # type (output) intens(low)  color(white)  caps(off)
   _ type (input)  intens(low)  color(yellow) caps(off) pad('_')
   + type (text)   intens(low)  color(green)
   / type (text)   intens(low)  color(yellow)
   ~ type (text)   intens(high) color(turquoise)
   @ type (text)   intens(high) color(red)   caps(off) hilite(reverse)
)body window(42,19) expand ($$)
/.........................................
+          @ DB2 Utilities V4 +
/.........................................
+
+            _z+[che   +
+            _z+[cop   +
+            _z+[mod   +
+            _z+[qui   +
+
+            _z+[rec   +
+            _z+[reo   +
+            _z+[rep   +
+            _z+[run   +
+            _z+[sto   +
+
/.........................................
+
#msg                                     +
+             / PF3 - End +     ~1998,"ZB"
)init
  .ZVARS = '(ch co mo qu re ro rp ru st)'
  &che = 'Check'
  &cop = 'Copy'
  &mod = 'Modify'
  &qui = 'Quiesce'
  &rec = 'Recover'
  &reo = 'Reorg'
  &rep = 'Report'
  &run = 'Runstat'
  &sto = 'Stospace'
  &msg = 'Place cursor on choice and press <Enter>'
  IF (&kurs = CH,CHE)
      .attr (che) = 'color (yellow) caps(on)'
  IF (&kurs = CO,COP)
      .attr (cop) = 'color (yellow) caps(on)'
  IF (&kurs = MO,MOD)
      .attr (mod) = 'color (yellow) caps(on)'
  IF (&kurs = QU,QUI)
      .attr (qui) = 'color (yellow) caps(on)'
  IF (&kurs = RE,REC)
      .attr (rec) = 'color (yellow) caps(on)'
  IF (&kurs = RO,REO)
      .attr (reo) = 'color (yellow) caps(on)'
  IF (&kurs = RP,REP)
      .attr (rep) = 'color (yellow) caps(on)'
  IF (&kurs = RU,RUN)
      .attr (run) = 'color (yellow) caps(on)'
  IF (&kurs = ST,STO)
      .attr (sto) = 'color (yellow) caps(on)'
)proc
  &kurs = .CURSOR
  if (.pfkey = pf03) &pf3 = exit
)end
```

DBUTUM

```
)attr default($+_)
| type (text)   intens(low)  color(white)
@ type (text)   intens(high) color(red)   caps(off) hilite(reverse)
| type (input)  intens(non)  color(green) caps(on)  just(left)
# type (output) intens(high) color(yellow)  caps(off)
)body default($~\) window(53,3)
|zcmd +     @ Message display |amt  |
|-----------50%----------100% |
)model clear(messg)
#z                                                  +
)init
     .zvars = '(messg)'
)reinit
)proc
  if (.pfkey = pf03) &pf3 = exit
)end
```

DBCHEM

```
)Attr Default(%+_)
 | type(text)   intens(high) caps(on ) color(yellow)
 $ type(output) intens(high) caps(off) color(yellow)
 ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
 # type(text)   intens(high) caps(off) hilite(reverse)
 } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
 [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ? DB2 CHECK - Parameter Entry +|-/-/-
%Command ===>_zcmd                                                      +
+
#PARAMETER  #PARAMETER VALUE                 #PROMPT                    +
+
+SSID     =>[db2   +                         DB2 Sub-System Identifier
+Type     =>[typ   +                         Check Index, Data or Both
+Scope    =>[scop  +                         PENDING or ALL
+Creator  =>[crec  +                         Table Creator
+Name     =>[tabc  +                         Table Name
+Tsname   =>[tsnc  +                         Tablespace Name
+Dbname   =>[dbnc  +                         Database   Name
+Volume   =>[vol   +                         Volser
+Catname  =>[catn  +                         Catalog name
+Tracks   =>[trk+                            Tracks per Cylinder - 15
+
            $msg                                                         +
+
                                } PF3 Return +
)Init
  &msg = 'Enter parameters values for the CHECK service |'
  if (&db2 ¬= ' ')
      .attr (db2) = 'pad(nulls)'
  if (&typ ¬= ' ')
      .attr (typ) = 'pad(nulls)'
  if (&scop ¬= ' ')
      .attr (scop) = 'pad(nulls)'
  if (&crec ¬= ' ')
      .attr (crec) = 'pad(nulls)'
  if (&tabc ¬= ' ')
      .attr (tabc) = 'pad(nulls)'
  if (&tsnc ¬= ' ')
      .attr (tsnc) = 'pad(nulls)'
  if (&dbnc ¬= ' ')
      .attr (dbnc) = 'pad(nulls)'
  if (&vol ¬= ' ')
      .attr (vol) = 'pad(nulls)'
  if (&catn ¬= ' ')
      .attr (catn) = 'pad(nulls)'
  if (&trk ¬= ' ')
      .attr (trk) = 'pad(nulls)'
)Reinit
)Proc
  &ctyp = TRUNC(&typ,' ')
  if (&ctyp='I' | &ctyp='IN') &typ = 'INDEX'
  if (&ctyp='D' | &ctyp='DA') &typ = 'DATA'
  if (&ctyp='B' | &ctyp='BO') &typ = 'BOTH'
  &sc = TRUNC(&scop,' ')
  if (&sc='A' | &sc='AL') &scop = 'ALL'
  if (&sc='P' | &sc='PE') &scop = 'PENDING'
  VPUT (db2 typ scop crec tabc tsnc dbnc vol catn trk) PROFILE
)End
```

DBCOPM

```
)Attr Default(%+_)
  | type(text)   intens(high) caps(on ) color(yellow)
  $ type(output) intens(high) caps(off) color(yellow)
  ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
  # type(text)   intens(high) caps(off) hilite(reverse)
  } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
  [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ? DB2 COPY - Parameter Entry +|-/-/-
%Command ===>_zcmd                                                       +
+
#PARAMETER  #PARAMETER VALUE                #PROMPT                +
+
+SSID     =>[db2   +                        DB2 Sub-System Identifier
+Creator  =>[crec  +                        Table Creator
+Name     =>[tabc  +                        Table Name
+Tsname   =>[tsnc  +                        Tablespace Name
+Dbname   =>[dbnc  +                        Database   Name
+Card      >[card  +                        Total number of rows
+Full     =>[ful   +                        Full image copy|YES+or|NO
+Shrlevel =>[ref   +                        |Reference+or|Change
+Copypref =>[pref  +                        Dataset PREFIX for IC's
+Stopts   =>[sto+                           Stop tablespace|YES+or|NO
+Quiesce  =>[que+                           Quiesce tspace |YES+or|NO
+Offsite  =>[off+                           Create OFFSITE |YES+or|NO
+Devt     =>[dev   +                        Device Type  (339Ø, Tape)
+Volume   =>[vol   +                        Volser
+Catname  =>[catn  +                        Catalog name
+Tracks   =>[trk+                           Tracks per Cylinder - 15
+
            $msg                                                   +
                              } PF3 Return +
)Init
  if (&db2 ¬= ' ')
      .attr (db2) = 'pad(nulls)'
  if (&crec ¬= ' ')
      .attr (crec) = 'pad(nulls)'
  if (&tabc ¬= ' ')
      .attr (tabc) = 'pad(nulls)'
  if (&tsnc ¬= ' ')
      .attr (tsnc) = 'pad(nulls)'
  if (&dbnc ¬= ' ')
      .attr (dbnc) = 'pad(nulls)'
  if (&card ¬= ' ')
      .attr (card) = 'pad(nulls)'
  if (&ful ¬= ' ')
      .attr (ful) = 'pad(nulls)'
  if (&ref ¬= ' ')
      .attr (ref) = 'pad(nulls)'
  if (&pref ¬= ' ')
      .attr (pref) = 'pad(nulls)'
  if (&sto ¬= ' ')
      .attr (sto) = 'pad(nulls)'
  if (&que ¬= ' ')
      .attr (que) = 'pad(nulls)'
  if (&off ¬= ' ')
      .attr (off) = 'pad(nulls)'
  if (&dev ¬= ' ')
      .attr (dev) = 'pad(nulls)'
  if (&vol ¬= ' ')
```

```
      .attr (vol) = 'pad(nulls)'
  if (&catn ¬= ' ')
      .attr (catn) = 'pad(nulls)'
  if (&trk ¬= ' ')
      .attr (trk) = 'pad(nulls)'
  &msg = 'Enter parameters values for the COPY service |'
)Reinit
)Proc
  &sr = TRUNC(&ref,' ')
  if (&sr='R'  | &sr='RE'   ) &ref = 'REFERENCE'
  if (&sr='C'  | &sr='CH'   ) &ref = 'CHANGE'
  VPUT (db2 crec tabc tsnc dbnc card pref ful) PROFILE
  VPUT (ref sto que off dev vol catn trk) PROFILE
)End
```

DBMODM

```
)Attr Default(%+_)
  | type(text)   intens(high) caps(on ) color(yellow)
  $ type(output) intens(high) caps(off) color(yellow)
  ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
  # type(text)   intens(high) caps(off) hilite(reverse)
  } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
  [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ? DB2 MODIFY - Parameter Entry +|-/-/-
%Command ===>_zcmd                                                     +
+
#PARAMETER  #PARAMETER VALUE                #PROMPT                    +
+
+SSID     =>[db2     +                      DB2 Sub-System Identifier
+Delage   =>[dage    +                      Delete AGE in days
+Deldate  =>[ddate   +                      Delete DATE, YYYYMMDD form
+Dbname   =>[dbnc    +                      Database   Name
+Tsname   =>[tsnc    +                      Tablespace Name
+
            $msg                                                       +
+
                               } PF3 Return +
)Init
  &msg = 'Enter parameters values for the MODIFY service |'
  if (&db2 ¬= ' ')
      .attr (db2) = 'pad(nulls)'
  if (&dage ¬= ' ')
      .attr (dage) = 'pad(nulls)'
  if (&ddate ¬= ' ')
      .attr (ddate) = 'pad(nulls)'
  if (&dbnc ¬= ' ')
      .attr (dbnc) = 'pad(nulls)'
  if (&tsnc ¬= ' ')
      .attr (tsnc) = 'pad(nulls)'
)Reinit
)Proc
  VPUT (db2 tsnc dbnc dage ddate) PROFILE
)End
```

DBQUIM

```
)Attr Default(%+_)
   | type(text)   intens(high) caps(on ) color(yellow)
   $ type(output) intens(high) caps(off) color(yellow)
   ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
   # type(text)   intens(high) caps(off) hilite(reverse)
   } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
   [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ? DB2 QUIESCE - Parameter Entry +|-/-/-
%Command ===>_zcmd                                                     +
+
#PARAMETER  #PARAMETER VALUE                 #PROMPT                   +
+
+SSID     =>[db2     +                       DB2 Sub-System Identifier
+Creator  =>[crec    +                       Table Creator
+Name     =>[tabc    +                       Table Name
+Tsname   =>[tsnc    +                       Tablespace Name
+Dbname   =>[dbnc    +                       Database   Name
+
            $msg                                                       +
+
                               } PF3 Return +
)Init
  &msg = 'Enter parameters values for the QUIESCE service |'
  if (&db2 ¬= ' ')
      .attr (db2) = 'pad(nulls)'
  if (&crec ¬= ' ')
      .attr (crec) = 'pad(nulls)'
  if (&tabc ¬= ' ')
      .attr (tabc) = 'pad(nulls)'
  if (&tsnc ¬= ' ')
      .attr (tsnc) = 'pad(nulls)'
  if (&dbnc ¬= ' ')
      .attr (dbnc) = 'pad(nulls)'
)Reinit
)Proc
  VPUT (db2 crec tabc tsnc dbnc) PROFILE
)End
```

DBRECM

```
)Attr Default(%+_)
   | type(text)   intens(high) caps(on ) color(yellow)
   $ type(output) intens(high) caps(off) color(yellow)
   ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
   # type(text)   intens(high) caps(off) hilite(reverse)
   } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
   [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ? DB2 RECOVER - Parameter Entry +|-/-/-
%Command ===>_zcmd                                                      +
+
#PARAMETER  #PARAMETER VALUE                 #PROMPT                    +
+
+SSID     =>[db2     +                       DB2 Sub-System Identifier
+Creator  =>[crec    +                       Table Creator
+Name     =>[tabc    +                       Table Name
+Tsname   =>[tsnc    +                       Tablespace Name
+Dbname   =>[dbnc    +                       Database   Name
+Rectype   >[rtyp    +                       CURrent, RBA or TOCOPY
+Datefrom =>[dfr     +                       From YYYYMMDD
+Dateto   =>[dto     +                       To   YYYYMMDD
+Volume   =>[vol     +                       Volser
+Volsort  =>[vola    +                       Volser for SORTWORK
+Volsort  =>[volb    +                       Volser for SORTWORK
+Catname  =>[catn    +                       Catalog name
+Tracks   =>[trk     +                       Tracks per Cylinder - 15
+
            $msg                                                        +
+
                               } PF3 Return +
)Init
  if (&db2 ¬= ' ')
      .attr (db2) = 'pad(nulls)'
  if (&crec ¬= ' ')
      .attr (crec) = 'pad(nulls)'
  if (&tabc ¬= ' ')
      .attr (tabc) = 'pad(nulls)'
  if (&tsnc ¬= ' ')
      .attr (tsnc) = 'pad(nulls)'
  if (&dbnc ¬= ' ')
      .attr (dbnc) = 'pad(nulls)'
  if (&rtyp ¬= ' ')
      .attr (rtyp) = 'pad(nulls)'
  if (&dfr ¬= ' ')
      .attr (dfr) = 'pad(nulls)'
  if (&dto ¬= ' ')
      .attr (dto) = 'pad(nulls)'
  if (&vol ¬= ' ')
```

```
      .attr (vol) = 'pad(nulls)'
  if (&vola ¬= ' ')
      .attr (vola) = 'pad(nulls)'
  if (&volb ¬= ' ')
      .attr (volb) = 'pad(nulls)'
  if (&catn ¬= ' ')
      .attr (catn) = 'pad(nulls)'
  if (&trk ¬= ' ')
      .attr (trk) = 'pad(nulls)'
  &msg = 'Enter parameters values for the RECOVER service |'
)Reinit
)Proc
  &rt = TRUNC(&rtyp,' ')
  if (&rt='R' | &rt='RB') &rtyp = 'RBA'
  if (&rt='C' | &rt='CU') &rtyp = 'CUR'
  if (&rt='T' | &rt='TO') &rtyp = 'TOCOPY'
  VPUT (db2 crec tabc tsnc dbnc rtyp) PROFILE
  VPUT (dfr dto vol vola volb catn trk) PROFILE
)End
```

DBREPM

```
)Attr Default(%+_)
   | type(text)   intens(high) caps(on ) color(yellow)
   $ type(output) intens(high) caps(off) color(yellow)
   ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
   # type(text)   intens(high) caps(off) hilite(reverse)
   } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
   [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ? DB2 REPORT - Parameter Entry +|-/-/-
%Command ===>_zcmd                                                      +
+
#PARAMETER  #PARAMETER VALUE                 #PROMPT                    +
+
+SSID     =>[db2     +                       DB2 Sub-System Identifier
+Creator  =>[crec    +                       Table Creator
+Name     =>[tabc    +                       Table Name
+Tsname   =>[tsnc    +                       Tablespace Name
+Dbname   =>[dbnc    +                       Database   Name
+Type     =>[rtype   +                       Recover-Tablespaceset-Both
+Current  =>[rcu     +                       Last recoverable point
+Summary  =>[rsu     +                       VOLSER numbers ONLY
+
            $msg                                                        +
+
                               } PF3 Return +
)Init
  &msg = 'Enter parameters values for the REPORT service |'
  if (&db2 ¬= ' ')
```

```
        .attr (db2) = 'pad(nulls)'
    if (&crec ¬= ' ')
        .attr (crec) = 'pad(nulls)'
    if (&tabc ¬= ' ')
        .attr (tabc) = 'pad(nulls)'
    if (&tsnc ¬= ' ')
        .attr (tsnc) = 'pad(nulls)'
    if (&dbnc ¬= ' ')
        .attr (dbnc) = 'pad(nulls)'
    if (&rtype ¬= ' ')
        .attr (rtype) = 'pad(nulls)'
    if (&rcu ¬= ' ')
        .attr (rcu) = 'pad(nulls)'
    if (&rsu ¬= ' ')
        .attr (rsu) = 'pad(nulls)'
)Reinit
)Proc
    &rt = TRUNC(&rtype,' ')
    IF (&rt='R' | &rt='RE' | &rt='REC' ) &rtype = 'RECOVERY'
    IF (&rt='T' | &rt='TA' | &rt='TAB' ) &rtype = 'TABLESPACESET'
    IF (&rt='B' | &rt='BO' | &rt='BOT' ) &rtype = 'BOTH'
    VPUT (db2 crec tabc tsnc dbnc rtype rcu rsu) PROFILE
)End
```

DBRUNM

```
)Attr Default(%+_)
  | type(text)   intens(high) caps(on ) color(yellow)
  $ type(output) intens(high) caps(off) color(yellow)
  ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
  # type(text)   intens(high) caps(off) hilite(reverse)
  } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
  [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ? DB2 RUNSTATS - Parameter Entry +|-/-/-
%Command ===>_zcmd                                                  +
+
#PARAMETER  #PARAMETER VALUE                #PROMPT                 +
+
+SSID     =>[db2     +                      DB2 Sub-System Identifier
+Creator  =>[crec    +                      Table Creator
+Name     =>[tabc    +                      Table Name
+Tsname   =>[tsnc    +                      Tablespace Name
+Dbname   =>[dbnc    +                      Database   Name
+Shrlevel =>[ref     +                      |Reference+or|Change
+Report   =>[rre     +                      Report on statistics
+Update   =>[upd     +                      ALL-SPACE-ACCESSPATH-NONE
+
            $msg                                                    +
+
                               } PF3 Return +
)Init
  &msg = 'Enter parameters values for the RUNSTATS service |'
  if (&db2 ¬= ' ')
      .attr (db2) = 'pad(nulls)'
  if (&crec ¬= ' ')
      .attr (crec) = 'pad(nulls)'
  if (&tabc ¬= ' ')
      .attr (tabc) = 'pad(nulls)'
  if (&tsnc ¬= ' ')
      .attr (tsnc) = 'pad(nulls)'
  if (&dbnc ¬= ' ')
      .attr (dbnc) = 'pad(nulls)'
  if (&ref ¬= ' ')
      .attr (ref) = 'pad(nulls)'
  if (&rre ¬= ' ')
      .attr (rre) = 'pad(nulls)'
  if (&upd ¬= ' ')
      .attr (upd) = 'pad(nulls)'
)Reinit
)Proc
  &sr = TRUNC(&ref,' ')
  if (&sr='R'  | &sr='RE'   ) &ref = 'REFERENCE'
  if (&sr='C'  | &sr='CH'   ) &ref = 'CHANGE'
  &up = TRUNC(&upd,' ')
  if (&up='A'  | &up='AL'   ) &upd = 'ALL'
  if (&up='AC' | &up='ACC'  ) &upd = 'ACCESSPATH'
  if (&up='S'  | &up='SP'   ) &upd = 'SPACE'
  if (&up='N'  | &up='NO'   ) &upd = 'NONE'
  VPUT (db2 crec tabc tsnc dbnc ref rre upd) PROFILE
)End
```

DBREOM

```
)Attr Default(%+_)
  | type(text)   intens(high) caps(on ) color(yellow)
  $ type(output) intens(high) caps(off) color(yellow)
  ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
  # type(text)   intens(high) caps(off) hilite(reverse)
  } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
  [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ? DB2 REORG - Parameter Entry +|-/-/-
%Command ===>_zcmd                                                     +
+
#PARAMETER  #PARAMETER VALUE                 #PROMPT                   +
+
+SSID     =>[db2     +                       DB2 Sub-System Identifier
+Creator  =>[crec    +                       Table Creator
+Name     =>[tabc    +                       Table Name
```

```
+Tsname   =>[tsnc    +                       Tablespace Name
+Dbname   =>[dbnc    +                       Database   Name
+Icopy    =>[ico     +                       Before-After-Both-None
+Copypref =>[pref    +                       Dataset PREFIX for IC's
+Log      =>[log+                            Log option|YES+or|NO
+Sortdata =>[sor+                            Sortdata|YES+or|NO
+Keepdict =>[dic+                            Keepdictionary|YES+or|NO
+Stopts   =>[sto+                            Stop tablespace|YES+or|NO
+Quiesce  =>[que+                            Quiesce tspace |YES+or|NO
+Runstats =>[rru+                            Runstats tspace |YES+or|NO
+Volume   =>[vol      + [vola  + [volb  +    Volser
+Catname  =>[catn     +                      Catalog name
+Tracks   =>[trk+                            Tracks per Cylinder - 15
+
            $msg                                                       +
                               } PF3 Return +
)Init
  if (&db2 ¬= ' ')
      .attr (db2) = 'pad(nulls)'
  if (&crec ¬= ' ')
      .attr (crec) = 'pad(nulls)'
  if (&tabc ¬= ' ')
      .attr (tabc) = 'pad(nulls)'
  if (&tsnc ¬= ' ')
      .attr (tsnc) = 'pad(nulls)'
  if (&dbnc ¬= ' ')
      .attr (dbnc) = 'pad(nulls)'
  if (&ico ¬= ' ')
      .attr (ico) = 'pad(nulls)'
  if (&pref ¬= ' ')
      .attr (pref) = 'pad(nulls)'
  if (&log ¬= ' ')
      .attr (log) = 'pad(nulls)'
  if (&sor ¬= ' ')
      .attr (sor) = 'pad(nulls)'
  if (&sto ¬= ' ')
      .attr (sto) = 'pad(nulls)'
  if (&que ¬= ' ')
      .attr (que) = 'pad(nulls)'
  if (&rru ¬= ' ')
      .attr (rru) = 'pad(nulls)'
  if (&dic ¬= ' ')
      .attr (dic) = 'pad(nulls)'
  if (&vol ¬= ' ')
      .attr (vol) = 'pad(nulls)'
  if (&vola¬= ' ')
      .attr (vola) = 'pad(nulls)'
  if (&volb¬= ' ')
      .attr (volb) = 'pad(nulls)'
  if (&catn ¬= ' ')
      .attr (catn) = 'pad(nulls)'
  if (&trk ¬= ' ')
      .attr (trk) = 'pad(nulls)'
  &msg = 'Enter parameters values for the REORG service |'
)Reinit
)Proc
  &sr = TRUNC(&ico,' ')
  if (&sr='A' | &sr='AF') &ico = 'AFTER'
  if (&sr='B' | &sr='BE') &ico = 'BEFORE'
  if (          &sr='BO') &ico = 'BOTH'
  if (&sr='N' | &sr='NO') &ico = 'NONE'
  VPUT (db2 crec tabc tsnc dbnc ico pref log sor) PROFILE
  VPUT (sto que rru dic vol vola volb catn trk) PROFILE
)End
```

DBSTOM

```
)Attr Default(%+_)
  | type(text)   intens(high) caps(on ) color(yellow)
  $ type(output) intens(high) caps(off) color(yellow)
  ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
  # type(text)   intens(high) caps(off) hilite(reverse)
  } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
  [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ? DB2 STOSPACE - Parameter Entry +|-/-/-
%Command ===>_zcmd                                                  +
+
#PARAMETER  #PARAMETER VALUE                #PROMPT                 +
+
+SSID     =>[db2     +                      DB2 Sub-System Identifier
+All      =>[sga     +                      All stogroups Yes or No
+Creator  =>[scre    +                      Creator  Name
+Sgname   =>[sgn     +                      Stogroup Name
+
            $msg                                                    +
+
                               } PF3 Return +
)Init
  &msg = 'Enter parameters values for the STOSPACE service |'
  if (&db2 ¬= ' ')
      .attr (db2) = 'pad(nulls)'
  if (&sga ¬= ' ')
      .attr (sga) = 'pad(nulls)'
  if (&scre ¬= ' ')
      .attr (scre) = 'pad(nulls)'
  if (&sgn ¬= ' ')
      .attr (sgn) = 'pad(nulls)'
)Reinit
)Proc
```

```
  VPUT (db2 sga scre sgn) PROFILE
)End
```

DBULIST

```
)Attr Default(%+_)
   ( type(text  ) intens(high) hilite(reverse)
   ] type(text  ) intens(high) hilite(reverse) color(green)
   / type(text  ) intens(high) hilite(reverse) color(yellow)
   ~ type(output) intens(high)                 color(red)
   [ type(output) intens(high) hilite(reverse) color(green) caps(off)
   + type(text  ) intens(low )
   _ type( input) intens(high) caps(on ) just(left )
   ¬ type(output) intens(low ) caps(off) just(asis )
)Body window(74,19)
[head                              +
+
+Command ===>_zcmd                                    +Scroll ===>_amt +
+
+Press]Enter+to have this service continue.
+Press]End  +to  respecify your PARAMETERS.
+
]Dbname  +]Tsname  +]Table           +]Creator +       ]  Card+/Message+
+
)Model
¬z       +¬z       +¬z               +¬z       +¬z             +~z      +
)Init
  .ZVARS = '(v1 v2 v3 v4 v5 v6)'
  &amt = PAGE
)Reinit
)Proc
)End
```

DBMLIST

```
)Attr Default(%+_)
   ( type(text  ) intens(high) hilite(reverse)
   ] type(text  ) intens(high) hilite(reverse) color(green)
   / type(text  ) intens(high) hilite(reverse) color(yellow)
   ~ type(output) intens(high)                 color(red)
   + type(text  ) intens(low )
   _ type( input) intens(high) caps(on ) just(left )
   ¬ type(output) intens(low ) caps(off) just(asis )
)Body window(76,19)
(DB2 Modify - Selection Result+
+
+Command ===>_zcmd                                    +Scroll ===>_amt +
+
+Press]Enter+to have this service continue.
+Press]End  +to  respecify your PARAMETERS.
```

```
+
]Dbname  ]Tsname  ]Icdate    ]Dsname                                  +
+
)Model
¬z       ¬z       ¬z         ¬z                                       +
)Init
  .ZVARS = '(db ts icd dsn)'
  &amt = PAGE
)Reinit
)Proc
)End
```

DBRLIST

```
)Attr Default(%+_)
  | type(text)   intens(high) caps(on ) color(yellow)
  $ type(output) intens(high) caps(off) color(yellow) hilite(reverse)
  ? type(text)   intens(high) caps(on ) color(red) hilite(uscore)
  # type(text)   intens(high) caps(off) hilite(reverse)
  } type(text)   intens(high) caps(off) color(white)
  ] type( input) intens(high) caps(on ) just(left ) pad('-')
  { type(output) intens(low ) caps(off) color(white) hilite(reverse)
  ¬ type(output) intens(low ) caps(off) just(asis ) color(turquoise)
)Body  Expand(//)
?Select Image Copy for Recovery$dbts                 +
%Command ===>_zcmd                                 / /%Scroll ===>_amt +
|Select with an 'S', press ENTER to skip RECOVERY of this TABLESPACE
|or press PF3, to return to Parameter Entry.
+---------------------------------------
+Valid sel:|S+Select
+---------------------------------------
                       # Ic +                                         +
#S #Icdate    #Ictime  #type#Dsnum{rectype                            +
)Model
]z+¬z         ¬z       ¬z   ¬z    ¬z                                  +
)Init
  .ZVARS = '(sel icd ict ity idsn rbadsn)'
  &amt = PAGE
  &sel = ''
  IF (&rtyp = 'RBA') &rectype = 'Start_rba'
  IF (&rtyp = 'TOCOPY') &rectype = 'Dsname'
)Reinit
)Proc
)End
```

DBSLIST

```
)Attr Default(%+_)
  ( type(text  ) intens(high) hilite(reverse)
  ] type(text  ) intens(high) hilite(reverse) color(green)
  / type(text  ) intens(high) hilite(reverse) color(yellow)
  ~ type(output) intens(high)                 color(red)
  + type(text  ) intens(low )
  _ type( input) intens(high) caps(on ) just(left )
  ¬ type(output) intens(low ) caps(off) just(asis )
)Body window(74,19)
(DB2 Stospace - Selection Result+
+
+Command ===>_zcmd                                    +Scroll ===>_amt +
+
+Press]Enter+to have this service continue.
+Press]End  +to  respecify your PARAMETERS.
+
]Creator +]Sgname  +                                                   +
+
)Model
¬z       +¬z       +                                                   +
)Init
  .ZVARS = '(cr sgname)'
  &amt = PAGE
)Reinit
)Proc
)End
```

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

Bernard Zver
Database Administrator
Informatika Maribor (Slovenia)

Avoiding SPUFI panels for SQL

This EXEC allows you to execute some SQL without having to go through the SPUFI panels.

The EXEC is an edit macro and functions in two modes. In the first mode, the EXEC is invoked from the command line and reads in the entire member. In the second mode, it will only process the lines selected by doing a QQ/QQ block command in the prefix area, and typing the command RUNSQL on the command line.

In either mode, the EXEC then invokes DSNTEP2 to run whatever has been read in. It then views the output file.

To tailor the EXEC to run on other systems, see the tailoring section.

SOURCE CODE

```
/* rexx */
  trace n
 Address ISPEXEC
 'ISREDIT MACRO (BIGLIN) NOPROCESS'
 'ISREDIT PROCESS RANGE Q'
 rcc = rc
 /* If rc=Ø then a  Q  was entered. */
 /* If rc=4 then no Q was entered.  */

 /*******************************************************************/
 /*  db2id   is the DB2 sub-system.                                 */
 /*******************************************************************/

 upper biglin
 parse var biglin db2id

 born = db2id
 If(pos(born,'HELP') > Ø   born = '?'   born = ' ') then Do
   ff1 = center('===> This EXEC executes an SQL query. <===',78)
   say copies('*',78)
   say ff1
   say copies('*',78)
   say ' '
   say 'This EXEC allows you to execute any SQL within a normal PDS'
   say 'member.  You do not have to be in SPUFI.'
   say ' '
   say ' '
   say 'On the command line type:  RUNSQL DB2-system-id.'
   say ' '
   say 'e.g.:                      RUNSQL DSN'
   say ' '
   say ' '
   say 'If you want to run just a few lines in the member, then '
   say 'use the delimiters Q/QQ in the prefix area, and the same'
   say 'RUNSQL DBA1 command on the command line.'
   say ' '
   say ' '
   ms1 = 'Press PF1'
   ms2 = ,
   'RUNSQL db2id  '
   ZEDSMSG = ms1
   ZEDLMSG = ms2
   "ISPEXEC SETMSG MSG(ISRZ000)"
   exit
 End /* If(pos(born,'HELP') > 0   born = '?'   born = ' ') then Do */

 /*******************************************************************/
 /* To roll this EXEC out to other systems:                         */
 /*                                                                 */
 /*                        Required changes:                        */
 /* (1) Update the array www. with the relevant info (see example). */
 /*                                                                 */
 /*                        Optional changes:                        */
 /* (1) The EXEC uses userid.TEP2.SYSIN/SYSPRINT as work files.  If */
 /*     you want to change these names, then look for dsn.1 and     */
 /*     dsn.2.                                                      */
 /*                                                                 */
 /*******************************************************************/

 /************************************************************/
 /* Get the system on which the EXEC is running.             */
 /************************************************************/

 /************************************************************/
 /* Get the correct prog name from the system/DB2-id.        */
 /* The MVS-id (the JESNODE) is obtained by doing a          */
 /* sysvar(sysnode).                                         */
 /************************************************************/

 xx = outtrap('gvar.')
 wai = sysvar(sysnode)
 xx = outtrap(off)
 /* www = mvsid,db2id,program,plan,librun  */
 www.0 = 1
 www.1 = 'MVSID,DSN,DSNTEP2,DSNTEP2,DSN.RUNLIB.LOAD'
 ipass = 0
 Do jk = 1 to www.0
```

```
  www.jk = translate(www.jk,' ',',')
  If(subword(www.jk,1,1)=wai & subword(www.jk,2,1)=db2id) then Do
    ipass = 1
    nam1 = subword(www.jk,3,1)
    nam2 = subword(www.jk,4,1)
    nam3 = subword(www.jk,5,1)
    libreq =  "RUN PROGRAM("  nam1   ") PLAN("  nam2   ") " ,
              "LIB('"  nam3    "')"
    leave jk
  End /* subword(www.jk,1,1)=wai & subworrd(www.jk,2,1)=db2id) */
End /* Do jk = 1 to www.Ø */

If(ipass = Ø) then Do
  say 'Invalid DB2 system ' db2id 'on machine' wai
  say 'Am exiting ....'
  exit
End /* If(ipass = Ø) then Do */

/************************************************************/
/* Check if required datasets exist - if not create them.   */
/************************************************************/

Address TSO
usern = USERID()
dsn.1 = usern    ".TEP2.SYSIN"
dsn.2 = usern    ".TEP2.SYSPRINT"

Do jk = 1 to 2
  dsnt = "'"    dsn.jk    "'"
  xx = SYSDSN(dsnt)
  If(xx = 'OK') then Do
    Nop
  End /* If(xx = 'OK') then Do */
  Else Do
    If(jk = 1) then Do
      "ATTR OUT LRECL(8Ø) BLKSIZE(616Ø) RECFM(F B) DSORG(PS)"
      "ALLOC FI(WRITEO) DA('"dsn.jk"') " ,
      "UNIT(SYSDA) USING(OUT) SPACE(1,1) CYL" ,
      "CATALOG"
      "FREE ATTRLIST(OUT)"
      "FREE FI(WRITEO)"
    End /* If(jk = 1) then Do */
    If(jk = 2) then Do
      "ATTR OUT LRECL(133) BLKSIZE(133) RECFM(F B A) DSORG(PS)"
      "ALLOC FI(WRITEO) DA('"dsn.jk"') " ,
      "UNIT(SYSDA) USING(OUT) SPACE(1Ø,5) CYL" ,
      "CATALOG"
      "FREE ATTRLIST(OUT)"
      "FREE FI(WRITEO)"
    End /* If(jk = 2) then Do */
  End /* If(xx = 'OK') then Do */
End /* Do jk = 1 to 2 */

Select
  When (rcc = Ø) then Do

    /***********************************************/
    /* Read in just the lines selected by Q/QQ.    */
    /***********************************************/

    'ISREDIT (CMD) = RANGE_CMD'
    'ISREDIT (LINE1) = LINENUM .ZFRANGE'
    'ISREDIT (LINE2) = LINENUM .ZLRANGE'
    linestocut = line2 - line1 + 1
    icr = Ø /* The number of lines to process. */
    say 'You are running the following query:'
    say ' '
    Do jk = line1  to line2
      icr = icr + 1
      'ISREDIT (STM) = LINE' jk
      upper stm
      linj.icr = strip(stm,T,' ')
      say linj.icr
    End /* Do jk = 1 to line1 to line2 */
    linj.Ø = icr
    rowr = line1
    colr = 1
  End /* When (rcc = Ø) then Do */

  When (rcc = 4) then Do

    /***********************************************/
    /* Read in the whole member.                   */
    /***********************************************/

    'ISREDIT (LRECL) = DATA_WIDTH'
    'ISREDIT (FIRP) = LINENUM .ZF'
    'ISREDIT (LASP) = LINENUM .ZL'
    'ISREDIT (ROW,COL) = CURSOR'
    'ISREDIT NULLS = ON'
    'ISREDIT (ROW,COL) = CURSOR'
    'ISREDIT (LASP) = LINENUM .ZL'
    rowr = row
    colr = col
    row = 1
    Do jk = 1 to lasp
      'ISREDIT CURSOR = (ROW,COL)'
      'ISREDIT (STM) = LINE .ZCSR'
      upper stm
      linj.jk = stm
      row = row + 1
    End /* Do jk = 1 to lasp */
    linj.Ø = lasp
  End /* When (rcc = 4) then Do */

  Otherwise Do
    Nop
  End
End /* Select */

jm = Ø
xx = outtrap('gvar.')
address TSO "FREE F(DD1)"
xx = outtrap(OFF)
Address TSO  "ALLOC F(DD1) DA('"  dsn.1    "') SHR "
/************************************************************/
/* Issue query.                                             */
/************************************************************/
Address TSO
"EXECIO " linj.Ø "DISKW DD1 (STEM LINJ. FINIS"
gvar. = ''
xx = outtrap('gvar.')
Address TSO "FREE F(DD1)"
Address TSO "FREE F(SYSIN)"
Address TSO "FREE F(SYSPRINT)"
xx = outtrap(OFF)
address TSO  "ALLOC F(SYSIN)    DA('"    dsn.1    "') SHR "
address TSO  "ALLOC F(SYSPRINT) DA('"    dsn.2    "') SHR "
push "END"
push libreq
Address TSO "DSN SYSTEM("    db2id   ")"
/************************************************************/
/* TSO FREE all the files.                                  */
/************************************************************/
gvar. = ''
xx = outtrap('gvar.')
address TSO "FREE F(DD1)"
address TSO "FREE F(SYSIN)"
address TSO "FREE F(SYSPRINT)"
xx = outtrap(OFF)
/************************************************************/
/* View the sysprint output file.                           */
/************************************************************/
"ISPEXEC VIEW DATASET('"    dsn.2    "')"
"ISREDIT CURSOR = (ROWR,COLR)"
exit
```

DB2 news

DB2 users can benefit from BMC's new products in its Enterprise Data Propagation (EDP) portfolio. Recovery Plus for DB2 and IMS is a high-speed utility that recovers tablespaces and indexspaces and reduces CPU time by up to 70%. It also includes operational improvements such as partial recovery keywords, plus a new DB2 physical back-out capability.

Recovery Manager for DB2 automates the entire recovery process and also eliminates the need to identify manually which applications are affected and to locate a common point of consistency from which to recover.

Copy Plus for DB2 ensures valid copies for recovery purposes and ensures that more frequent image copies can be taken.

For further information contact:
BMC Software, 2101 CityWest Boulevard, Houston, TX 77042-2827, USA.
Tel: (713) 918 8800.
BMC Software, Compass House, 207-215 London Road, Camberley, Surrey, GU15 3EY, UK.
Tel: (01276) 24622.
URL: http://www.bmc.com.

* * *

IBM has announced the latest update to DB2 for OS/390 Version 5. Enhancements include the capability to run DB2 utilities from a stored procedure and improved performance for ODBC applications, including multiple thread support and a shadow of the catalog to reduce lock contention.

Other enhancements include improved accounting information for monitoring DDF performance, the ability to add end-user parameters to RRS attachment sign-on, WLM support for distributed connection load balancing, and parallel dataset allocation on opening. Also new is an ability to alter tablespace attributes without stopping the tablespace, plus performance enhancements to RUNSTATS and improved storage handling in IRLM.

For further information contact your local IBM representative.

* * *

AMI Software has announced quikDATE/DB2, for multi-tester, dynamic data ageing of DB2 databases in preparation for 1 January 2000. DB2 date content identified by the software can be tested or dynamically changed either globally, specifically to a table, or tailored to a single or set of columns. Date rules can also be modified at any time during the testing process.

Identification of all hidden dates, or those not defined as DATE, DAT, or DD in the DB2 table definition, is possible by coupling quikDATE/DB2 with AMI's Legacy Directory product.

For further information contact:
AMI Software, Fernlea House, Newby, Penrith, Cumbria, CA10 3EX, UK.
Tel: (01931) 714053.
URL: http://www.amisoftware.com

* * *