82

DB2

August 1999

In this issue

update

DB2 Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38030
From USA: 01144 1635 38030
E-mail: info@xephon.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/db2update.html; you will need the user-id shown on your address label.

Editor

Robert Burgess

Disclaimer

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

Subscriptions and back-issues

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

Showing DB2 restricted tablespaces

This article gives a simple REXX EXEC to show DB2 restricted tablespaces. This was written solely as a method of checking out any restricted tablespaces and therefore it does not have any bells or whistles. It can be amended to include all sorts of WTO(R)s or UTILITY start-off code as required. It can also be run as a batch job with suitable JCL put round it.

RESTRICT

```
/*************************** REXX ********************************/
/*                                                               */
/*           RESTRICT - DB2 RESTRICTED DATASETS                  */
/*                                                               */
/*   THIS REXX ALLOWS THE USER TO INTERROGATE THE CATALOG FOR A  */
/*   GIVEN DB2 SUBSYSTEM AND LIST ANY 'RESTRICTED' DATASETS      */
/*                                                               */
/*   SCREEN 1 (REST00P) LETS THE USER INSERT WHICH DB2           */
/*   SUBSYSTEM IS REQUIRED.                                      */
/*                                                               */
/*   THIS SCREEN IS CONSTANTLY REFRESHABLE BY SIMPLY HITTING     */
/*   <ENTER> OR BY CHANGING THE SUBSYSTEM CRITERIA AND HITTING   */
/*   <ENTER>.                                                    */
/*                                                               */
/*   THE SCREEN(S) USE INFORMATION GLEANED FORM THE FOLLOWING    */
/*   DSNTXXXI MESSAGES :- 360, 362, 365, 397 & DSN9022I.         */
/*                                                               */
/***************************************************************/
/* TRACE R */
   ADDRESS "ISPEXEC" "CONTROL ERRORS RETURN"
   ADDRESS "ISPEXEC" "TBCREATE RSLIST"||,
           " KEYS(TSN)"||,
           " NAMES(DBASE DBS DBD TST TSP TSS TSLO TSHI TSC TSP)    "||,
           " NOWRITE REPLACE"
   DO FOREVER
      ADDRESS "ISPEXEC" "TBDISPL RSLIST PANEL(REST00P)"
      IF RC > 4 THEN
         DO
           ADDRESS "ISPEXEC" "TBEND RSLIST"
           EXIT
         END
      ADDRESS "ISPEXEC" "VPUT (ZZSSID) PROFILE"
      IF ZTDSELS > 0 THEN
         DO
```

```
            CALL GET_RSLIST_ROW
            CALL PROCESS_REST_ROW
          END
       CALL BUILD_RESTLIST;
    END
    EXIT
/*********************************************************************/
/* BUILD RESTRICT LIST                                               */
/*********************************************************************/
BUILD_RESTLIST:
     RESTCOUNT = Ø
     NEWSTACK
     BUFFØ1= OUTTRAP("RESTLINE.","*")
     QUEUE "-DIS DATABASE(*) SPACENAM(*) RESTRICT LIMIT(*)"
     QUEUE "END"
     "DSN SYSTEM("ZZSSID")"
     BUFFØ1 = OUTTRAP(OFF)
     IF RC > Ø THEN
        DO
          SAY 'RC='||RC
          DO I = 1 TO RESTLINE.Ø
             SAY RESTLINE.I
          END
          RETURN
        END
     N = 1
     DO I = 1 TO RESTLINE.Ø
        CALL PARSE_DETAIL
     END
     ADDRESS "ISPEXEC" "TBCREATE RSLIST"||,
         " KEYS(TSN)"||,
         " NAMES(DBASE DBS DBD TST TSP TSS TSLO TSHI TSC TSP)       "||,
         " NOWRITE REPLACE"
     IF RC > 4 THEN
        DO
         SAY 'TBCREATE ERROR '||RC
         EXIT
        END
    DO I = 1 TO RESTCOUNT
       DBASE    = DBASE.I
       DBS      = DBS.I
       DBD      = DBD.I
       TSN      = TSN.I
       TST      = TST.I
       TSP      = TSP.I
       TSS      = TSS.I
       TSLO     = TSLO.I
       TSHI     = TSHI.I
       TSC      = TSC.I
       TSP      = TSP.I
```

```
      ADDRESS "ISPEXEC" "TBADD RSLIST"
   END
   ADDRESS "ISPEXEC" "TBSORT RSLIST FIELDS(TSN,C,A)"
   ADDRESS "ISPEXEC" "TBTOP RSLIST"
   IF RC > 4 THEN
      DO
        SAY 'TBTOP ERROR '||RC
        ADDRESS "ISPEXEC" "TBEND RSLIST"
        EXIT
      END
RETURN;
/*********************************************************************/
/* PARSE_DETAIL                                                      */
/*********************************************************************/
PARSE_DETAIL:
  IF SUBSTR(RESTLINE.I,Ø1,Ø8) = 'DSNT365I' THEN
     DO
             ADDRESS "ISPEXEC" "DISPLAY PANEL (REST ØØ2P)"
             IF RC > 4 THEN
                DO
                   EXIT
                END
             ADDRESS "ISPEXEC" "CONTROL DISPLAY LINE START(45)"
             EXIT
     END
  IF SUBSTR(RESTLINE.I,Ø1,Ø8) = '        ' THEN
     DO
       RETURN
     END
  IF SUBSTR(RESTLINE.I,Ø1,Ø8) = '————' THEN
     DO
       RETURN
     END
  IF SUBSTR(RESTLINE.I,Ø1,Ø8) = '********' THEN
     DO
       RETURN
     END
  IF SUBSTR(RESTLINE.I,Ø1,Ø8) = 'DSNT36ØI' THEN
     DO
       RETURN
     END
  IF SUBSTR(RESTLINE.I,Ø1,Ø8) = 'DSNT361I' THEN
     DO
       N = 1
       DBASE.N         = ''
       DBS.N           = ''
       DBD.N           = Ø
       TSN.N           = ''
       TST.N           = ''
       TSP.N           = ''
```

```
        TSS.N           = ''
        TSLO.N          = ''
        TSHI.N          = ''
        TSC.N           = ''
        TSP.N           = ''
        RETURN
      END
    IF SUBSTR(RESTLINE.I,Ø1,Ø8) = 'DSNT362I' THEN
      DO
        J = I
        PARSE VAR RESTLINE.J S1 S2 S3 S4 S5 S6 S7 S8
        SAVE_DBASE          = S5
        SAVE_DBS            = S8
        J = J + 1
        PARSE VAR RESTLINE.J S1 S2 S3 S4
        SAVE_DBD            = S4
        RETURN
      END
    IF SUBSTR(RESTLINE.I,Ø1,Ø8) = 'DSNT397I' THEN
      DO
      I = I + 4
      J = J + 4
      PARSE VAR RESTLINE.J S1 1Ø S2 15 S3 2Ø S4 39 S5,
                           48 S6 57 S7 66 S8
      DO UNTIL SUBSTR(S1,1,7) = '*******'
         DBASE.N          = SAVE_DBASE
         DBS.N            = SAVE_DBS
         DBD.N            = SAVE_DBD
         TSN.N            = S1
         TST.N            = S2
         TSP.N            = S3
         TSS.N            = S4
         TSLO.N           = S5
         TSHI.N           = S6
         TSC.N            = S7
         TSP.N            = S8
         N = N + 1
         J = J + 1
         I = I + 1
         RESTCOUNT = RESTCOUNT + 1
         PARSE VAR RESTLINE.J S1 1Ø S2 15 S3 2Ø S4 39 S5,
                              48 S6 57 S7 66 S8
      END
RETURN;
/*********************************************************************/
/* GET_RSLIST_ROW                                                    */
/*********************************************************************/
GET_RSLIST_ROW:
  ADDRESS "ISPEXEC" "TBGET RSLIST"
RETURN;
```

```
/*********************************************************************/
/* PROCESS_REST_ROW                                                  */
/*********************************************************************/
PROCESS_REST_ROW:
  DBASE           = ''
  DBS             = ''
  DBD             = Ø
  TSN             = ''
  TST             = ''
  TSP             = ''
  TSS             = ''
  TSLO            = ''
  TSHI            = ''
  TSC             = ''
  TSP             = ''
   ADDRESS "ISPEXEC" "DISPLAY PANEL (RESTØØ1P)"
   IF RC > 4 THEN
             DO
                RETURN
             END
   ADDRESS "ISPEXEC" "CONTROL DISPLAY LINE START(45)"
RETURN;
```

REST001P

```
)ATTR
/*********************************************************************/
/* RESTØØ1P - RESTRICTED DATASETS UTILITY                            */
/*********************************************************************/
   _ TYPE(INPUT)  INTENS(HIGH)  CAPS(ON)           JUST(LEFT )
   + TYPE(TEXT)   INTENS(LOW)   COLOR(BLUE)        SKIP(ON)
   ? TYPE(TEXT)   INTENS(LOW)   COLOR(TURQUOISE)   SKIP(ON)
   % TYPE(TEXT)   INTENS(HIGH)  COLOR(YELLOW)      SKIP(ON)
   ¬ TYPE(OUTPUT) INTENS(HIGH)  COLOR(YELLOW)      CAPS(ON) JUST(RIGHT)
   # TYPE(OUTPUT) INTENS(HIGH)  COLOR(GREEN)       CAPS(ON) JUST(RIGHT)
   ! TYPE(OUTPUT) INTENS(HIGH)  COLOR(GREEN)       CAPS(ON) JUST(LEFT)
)BODY CMD(C)
%———————————+DB2 RESTRICTED DATASETS%————————————
+OPTION ===>_C                                                  +SCR-
>_AMT +
+                                                             +DB2 SSID:
_Z   +
+
+
%DATABASE TSPACE   STATUS            DBD   <    PHYERR     > CATALOG
PIECE  +
%  NAME   NAME                      LNGTH   LOW      HIGH
+
%——— ———— ———————— —— ———— ———— ———— ——— +
```

```
)MODEL
!DBASE   !TSN     !TSS              !DBD   !TSLO    !TSHI    !TSC
!TSP    +
)INIT
 .ZVARS = '(ZZSSID)'
 .CURSOR = ZZSSID
)PROC
  VER  (&C LIST,END,' ')
  VER  (&ZZSSID NONBLANK LIST,DB2P,DB2U,DB2D,DB2T,DGDD,DGDC,DGDP)
)END
```

REST002P

```
)ATTR
/*********************************************************************/
/* REST00 2P - RESTRICTED DATASETS UTILITY                           */
/*********************************************************************/
   _ TYPE(INPUT)  INTENS(HIGH)  CAPS(ON)           JUST(LEFT )
   + TYPE(TEXT)   INTENS(LOW)   COLOR(BLUE)        SKIP(ON)
   ? TYPE(TEXT)   INTENS(LOW)   COLOR(TURQUOISE)   SKIP(ON)
   % TYPE(TEXT)   INTENS(HIGH)  COLOR(YELLOW)      SKIP(ON)
   ¬ TYPE(OUTPUT) INTENS(HIGH)  COLOR(YELLOW)      CAPS(ON) JUST(RIGHT)
   # TYPE(OUTPUT) INTENS(HIGH)  COLOR(GREEN)       CAPS(ON) JUST(RIGHT)
   ! TYPE(OUTPUT) INTENS(HIGH)  COLOR(GREEN)       CAPS(ON) JUST(LEFT)
)BODY CMD(C)
%———————————+DB2 RESTRICTED DATASETS%————————————
+OPTION ===>_C                                                +SCR-
>_AMT +
+                                                           +DB2 SSID:
_Z   +
+
+
%DATABASE TSPACE   STATUS            DBD   <    PHYERR     > CATALOG
PIECE  +
%  NAME   NAME                      LNGTH   LOW      HIGH
+
%———— ———— ———————— —— ———— ———— ———— ——— +
)MODEL
+
+
+
+
+
+
+
+
+
+                         ****************************
+
+                         NO DATABASES IN RESTRICT MODE
```

```
+
+                          ****************************
+
)INIT
 .ZVARS = '(ZZSSID)'
 .CURSOR = ZZSSID
)PROC
  VER  (&C LIST,END,' ')
  VER  (&ZZSSID NONBLANK LIST,DB2P,DB2U,DB2D,DB2T,DGDD,DGDC,DGDP)
)END
```

Ian McDonald
DB2 DBA (UK)

Estimating the space needed for a Type 2 index

DESCRIPTION

The REXX EXEC listed below will estimate the space needed to create a Type 2 index on a DB2 table. It calculates the PRIQTY and a 10% SECQTY based on the number of rows, key length, number of duplicate key entries, FREEPAGE, and PCTFREE parameters and whether or not the tablespace the index is created against is defined as LARGE.

The number of levels for the index will be calculated as well as suggested PRIQTY and SECQTY values. Additional PRIQTY and SECQTY values are calculated by specifying the index space allocation in multiples of 3390 track and cylinder sizes.

IXT2SPAC

```
/* REXX *//*********************************************************/
/*                                                                 */
/* IXT2SPAC - DB2 index space calculation (Type 2)                 */
/*                                                                 */
/*******************************************************************/

    numeric digits 16
```

```
    message = 'MSG()'
    do forever
        address "ISPEXEC" "DISPLAY PANEL(IXT2SPAC)"
        if rc > 0 then exit
        address "ISPEXEC" "CONTROL DISPLAY LINE START(43)"
        call p0100_calculate
    end
    exit

/*****************************************************************/
/* p0100_calculate                                               */
/*****************************************************************/
p0100_calculate:

/*****************************************************************/
/* Calculate total leaf pages                                    */
/*****************************************************************/
    n = dups
    if large = 'Y' then
        r = 6
    else
        r = 5
    if n = 1 then
        k = keylen + r + 2
    else
        k = keylen + 4 + (r * n)
    s = ((100-pctfree)*4038)%100
    entries = (n * (s / k))%1
    remaining_space = s - (entries/n) * k
    recs_per_partial = (remaining_space - (keylen+4)) % 5
    if recs_per_partial < 1 then
        partial_entries = 0
    else
        if n // recs_per_partial = 0 then
            partial_entries = n % (n%recs_per_partial)
        else
            partial_entries = n % (n%recs_per_partial+1)
    entries = entries + partial_entries
    if entries < 1 then
        entries = 1
    if number // entries = 0 then
        leaf_pages = number % entries
    else
        leaf_pages = number % entries + 1

/*****************************************************************/
/* Calculate number of levels and total non-leaf pages           */
/*****************************************************************/
    if n = 1 then
        k = keylen + 7
```

```
    else
        k = keylen + r + 6
    f = 100-pctfree
    if f < 90 then
        f = 90
    s = (f * 4046)%100
    entries = s % k
    min_child_pages = entries + 1
    if min_child_pages < 2 then
        min_child_pages = 2
    nonleaf_pages = 0
    levels = 1
    i = leaf_pages
    do while (i > 1)
        if i > 1 then
            levels = levels + 1
        if i // min_child_pages = 0 then
            i = i % min_child_pages
        else
            i = i % min_child_pages + 1
        nonleaf_pages = nonleaf_pages + i
    end

/*****************************************************************/
/* Calculate total space required                                */
/*****************************************************************/
    if freepage > 0 then
        free_pages = leaf_pages % freepage
    else
        free_pages = 0
    tree_pages = leaf_pages + nonleaf_pages
    if tree_pages < 2 then
      tree_pages = 2
    if (tree_pages + free_pages) // 8131 = 0 then
        space_map_pages = (tree_pages + free_pages) % 8131
    else
        space_map_pages = (tree_pages + free_pages) % 8131 + 1
    total_pages = 1 + free_pages + tree_pages + space_map_pages
    if total_pages < 4 then
      total_pages = 4
    bytes = 4096 * (total_pages + 2)
    if bytes // 1024 = 0 then
        kbytes = bytes % 1024
    else
        kbytes = bytes % 1024 + 1
    if kbytes // 10 = 0 then
        kbytes2 = kbytes % 10
    else
        kbytes2 = kbytes % 10 + 1
```

```
/*****************************************************************/
/* Calculate 3390 track and cylinder multiples                   */
/*****************************************************************/

    if kbytes // 48 = 0 then
        tkbytes = kbytes
    else
        tkbytes = kbytes + ( 48 - ( kbytes // 48 ) )
    tkbytes2 = tkbytes % 10
    if tkbytes2 // 48 = 0 then
        nop
    else
        tkbytes2 = tkbytes2 + ( 48 - ( tkbytes2 // 48 ) )
    if tkbytes // 48 = 0 then
        tracks = tkbytes % 48
    else
        tracks = tkbytes % 48 + 1
    if tkbytes2 // 48 = 0 then
        tracks2 = tkbytes2 % 48
    else
        tracks2 = tkbytes2 % 48 + 1
    if kbytes // 720 = 0 then
        nop
    else
        ckbytes = kbytes + ( 720 - ( kbytes // 720 ) )
    ckbytes2 = ckbytes % 10
    if ckbytes2 // 720 = 0 then
        nop
    else
        ckbytes2 = ckbytes2 + ( 720 - ( ckbytes2 // 720 ) )
    if ckbytes // 720 = 0 then
        cyls = ckbytes % 720
    else
        cyls = ckbytes % 720 + 1
    if ckbytes2 // 720 = 0 then
        cyls2 = ckbytes2 % 720
    else
        cyls2 = ckbytes2 % 720 + 1
return
```

ISPF PANEL IXT2SPAC

```
)ATTR DEFAULT(%$_)
/*********************************************************************/
/*                                                                   */
/*  IXT2SPAC - DB2 Type 2 index space calculation.                   */
/*                                                                   */
/*********************************************************************/
```

```
    $ TYPE(TEXT) COLOR(TURQ)
    @ TYPE(OUTPUT) COLOR(WHITE) JUST(RIGHT)

)BODY EXPAND(\\)
%IXT2SPAC$\-\%TYPE 2 INDEX SPACE CALCULATION$\-\%&ZDATE  $
%COMMAND$===>_ZCMD
%&ztime
$
$Enter index information:

    %Rows       $===>_number    $
    %Key length $===>_Z  $             (+ 1 per null)
    %Cardinality$===>_Z          $     (Avg # dup keys, 1 for unique)
    %Pctfree    $===>_Z $
    %Freepage   $===>_Z  $
    %Large (y/n)$===>_Z$

$Calculated level information:

    %Levels$===>@levels    $

$Calculated space usage:
                  $   primary     secondary
      %Kilobytes$===>@kbytes   $  @kbytes2   $
      %Kilobytes$===>@tkbytes  $  @tkbytes2  $ rounded to TRK boundary
    %3390 Tracks$===>@tracks   $  @tracks2   $
      %Kilobytes$===>@ckbytes  $  @ckbytes2  $ rounded to CYL boundary
 %3390 Cylinders$===>@cyls     $  @cyls2     $
)INIT
    .zvars = '(keylen dups pctfree freepage large)'
    if (&dups = '')
        &dups = '1'
    if (&pctfree = '')
        &pctfree = '10'
    if (&freepage = '')
        &freepage = '0'
    if (&large = '')
        &large = 'N'
)PROC
    ver (&number, nonblank, range, 1, 1000000000)
    ver (&keylen, nonblank, range, 1, 254)
    ver (&dups, nonblank, range, 1, 1000000000)
    ver (&pctfree, nonblank, range, 0, 99)
    ver (&freepage, nonblank, range, 0, 255)
    ver (&large nonblank list, 'Y', 'N')
)END
```

Analysing the DSNZPARM load module – part 3

This month we conclude the program that analyses the DSNZPARM load module and creates the originating assembly macro input.

```
SYSPACCB MVC   ZPRMCL16+Ø8(48),WORKCHAR
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SMFSTAT  - SMF STATISTICS FLAGS
         MVC   ZPRMCL16(Ø8),=C'SMFSTAT='
         MVC   WORKB32,SYSPSMFS           GET SMF STATISTICS FLAGS
         BAS   R14,BIT16RTN               CONVERT FIRST 16 BITS TO NUM
         BAS   R14,BIT32RTN               CONVERT NEXT 16 BITS TO NUM
         CLI   WORKCHR1,C')'              IF ) MEANS ALL BITS ARE Ø
         BNE   SYSPSTAB                   N. GO ON
         MVC   WORKCHAR(Ø2),=C'NO'        Y. SAY NO HERE
SYSPSTAB MVC   ZPRMCL16+Ø8(48),WORKCHAR
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> STATIME  - STATISTICS TIME
         MVC   ZPRMCL16(Ø8),=C'STATIME='
         SR    R9,R9                      ZERO REGISTER
         LH    R9,SYSPSTIM                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(7),D           PACK TO ZONE
         OI    ZPRMCL16+14,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> STORPROC - STORED PROCEDURE MVS NAME
         MVC   ZPRMCL16(Ø9),=C'STORPROC='
         MVC   ZPRMCL16+Ø9(Ø8),SYSPSPPN   GET PARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> STORMXAB - ALLOWABLE ABENDS FOR STORED PROCEDURES
         MVC   ZPRMCL16(Ø9),=C'STORMXAB='
         LH    R9,SYSPSPAB                GET ALLOWABLE ABENDS FOR STP
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(7),D           PACK TO ZONE
         OI    ZPRMCL16+15,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø9       MOVE NUMBER IN HOLD AREA
```

```
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø9(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> STORTIME - STORED PROCEDURE TIMEOUT VALUE
         MVC   ZPRMCL16(Ø9),=C'STORTIME='
         LH    R9,SYSPSPAB                GET STORPROC TIMEOUT VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(7),D           PACK TO ZONE
         OI    ZPRMCL16+15,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø9       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø9(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
         AIF   (NOT D'SYSPURCK).URCKTH                          V5
*> URCHKTH  - UR CHECKPOINT THRESHOLD                           V5
         MVC   ZPRMCL16(Ø8),=C'URCHKTH='                        V5
         SR    R9,R9                                            V5
         IC    R9,SYSPURCK                UR CHECKPOINT VALUE   V5
         CVD   R9,D                       CONVERT DECIMAL       V5
         UNPK  ZPRMCL16+Ø8(3),D           PACK TO ZONE          V5
         OI    ZPRMCL16+1Ø,X'FØ'          FIX LAST DIGIT        V5
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS    V5
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK      V5
         MVI   Ø(1),C','                  PLUG COMMA HERE       V5
         BAS   R14,ZWRTRTN             DO PRINT LINE            V5
.URCKTH  AIF   (NOT D'SYSPSCER).EXTSEC                          V5
*> EXTSEC   - EXTENDED SECURITY                                 V5
         MVC   ZPRMCL16(1Ø),=C'EXTSEC=NO '                      V5
         CLI   SYSPSCER,C'Y'                                    V5
         BNE   *+1Ø                                             V5
         MVC   ZPRMCL16(1Ø),=C'EXTSEC=YES'                      V5
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK      V5
         MVI   Ø(1),C','                  PLUG COMMA HERE       V5
         BAS   R14,ZWRTRTN             DO PRINT LINE            V5
.EXTSEC  ANOP                                                   V5
*> TRACLOC  - 4K ELEMENTS IN LOCAL TRACE TRABLE
         MVC   ZPRMCL16(Ø8),=C'TRACLOC='
         LH    R9,SYSPTLSZ                NUMBER OF 4K ELEMENTS
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(7),D           PACK TO ZONE
         OI    ZPRMCL16+14,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
```

```
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> TRACSTR  - MONITOR TRACING FLAG
         MVC   ZPRMCL16(Ø8),=C'TRACSTR='
         MVC   WORKB32,SYSPTRST           GET AUTO TRACE START
         BAS   R14,BIT16RTN               CONVERT FIRST 16 BITS TO NUM
         BAS   R14,BIT32RTN               CONVERT NEXT 16 BITS TO NUM
         CLI   WORKCHR1,C')'              IF ) MEANS ALL BITS ARE Ø
         BNE   *+1Ø                       N. GO ON
         MVC   WORKCHAR(Ø2),=C'NO'        Y. SAY NO HERE
         MVC   ZPRMCL16+Ø8(48),WORKCHAR
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> TRACTBL  - NO OF 4K SEGMENTS IN LOCAL TRACETBL
         MVC   ZPRMCL16(Ø8),=C'TRACTBL='
         LH    R9,SYSPTLSZ                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(7),D           PACK TO ZONE
         OI    ZPRMCL16+14,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø8       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         MVI   ZPRMCL72,C' '              PLUG SPACE HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
         TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.1Ø.98               *
               DSN6FAC                  '
*> FORMAT DSN6FAC     ************************************************
         USING DSN6FAC,R7
         L     R7,ZPARMPTR
         LA    RØ,4
         LA    R1,255(,R7)
         CLC   =CL8'DSN6FAC',4(R7)
         BE    *+12
         BXLE  R7,RØ,*-1Ø
         B     ABEND196
         L     R7,Ø(,R7)
         L     R2,=A(DSN6FAC)             SECTION TO BE ANALYSED
         CLC   FACID,FACID-DSN6FAC(R2)
         BNE   ABEND1Ø6
         CLC   FACEID,FACEID-DSN6FAC(R2)
         BNE   ABEND1Ø6                   SECTION DSN6FAC NOT FOUND
*> FORMAT DSN6FAC     ************************************************
         MVC   ZPRMCLØ5(Ø7),=C'DSN6FAC'
*> DDF      - DDF FLAG BYTE FOR STARTUP
         MVC   ZPRMCL16(Ø4),=C'DDF='
         CLI   FACSTART,C'N'
         BNE   DDF1ØØ
         MVC   ZPRMCL16+Ø4(Ø3),=C'NO '
```

```
         B     DDF900
DDF100   CLI   FACSTART,C'C'
         BNE   DDF200
         MVC   ZPRMCL16+04(07),=C'COMMAND'
         B     DDF900
DDF200   CLI   FACSTART,C'A'
         BNE   DDF900
         MVC   ZPRMCL16+04(04),=C'AUTO'
DDF900   TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   0(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> CMTSTAT  - DDF THREAD STATUS
         MVC   ZPRMCL16(08),=C'CMTSTAT='
         MVC   ZPRMCL16+08(08),FACCMST    GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   0(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> IDTHTOIN - IDLE THREAD TIMEOUT INTERVAL
         MVC   ZPRMCL16(09),=C'IDTHTOIN='
         LH    R9,FACTOIN                 GET TIMEOUT VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZEROHOLD(7),D              PACK TO ZONE
         OI    ZEROHOLD+6,X'F0'           FIX LAST DIGIT
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+09(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   0(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN                DO PRINT LINE
*> RESYNC   - MINUTES BETWEEN RESYNC PERIODS
         MVC   ZPRMCL16(07),=C'RESYNC='
         LH    R9,FACRESYC                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+07(7),D           PACK TO ZONE
         OI    ZPRMCL16+13,X'F0'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+07       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+07(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   0(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RLFERRD  - LIMIT OF CPU SECONDS
         MVC   ZPRMCL16(16),=C'RLFERRD=NOLIMIT '
         TM    FACRLFER,128               NOLIMIT
         BO    DDFNONU
         MVC   ZPRMCL16(16),=C'RLFERRD=NORUN   '
         TM    FACRLFER,64                NORUN
         BO    DDFNONU
         L     R9,FACRLFN                 GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZEROHOLD,D                 PACK TO ZONE
```

```
         OI    ZEROHOLD+L'ZEROHOLD-1,X'FØ' FIX LAST DIGIT
         BAS   R14,DZERORTN                DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD    MOVE TRUNCATED NUMBER BACK
DDFNONU  DS    ØH
         MVI   ZPRMCL72,C' '               PLUG COMMA HERE
         BAS   R14,ZWRTRTN              DO PRINT LINE
         TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.1Ø.98                *
               DSN6GRP                   '
*> FORMAT DSN6GRP     ***********************************************
         USING DSN6GRP,R7
         L     R7,ZPARMPTR
         LA    RØ,4
         LA    R1,255(,R7)
         CLC   =CL8'DSN6GRP',4(R7)
         BE    *+12
         BXLE  R7,RØ,*-1Ø
         B     FINALIZE
         L     R7,Ø(,R7)
         L     R2,=A(DSN6GRP)           SECTION TO BE ANALYSED
         CLC   GRPID,GRPID-DSN6GRP(R2)
         BNE   ABEND1Ø7
         CLC   GRPEYE,GRPEYE-DSN6GRP(R2)
         BNE   ABEND1Ø7                 SECTION DSN6GRP NOT FOUND
*> FORMAT DSN6GRP     ***********************************************
         MVC   ZPRMCLØ5(22),=C'DSN6GRP    DSHARE=YES,'
         TM    GRPDSHR,128
         BO    *+1Ø
         MVC   ZPRMCLØ5(22),=C'DSN6GRP    DSHARE=NO, '
         MVI   ZPRMCL72,C'X'
         BAS   R14,ZWRTRTN              DO PRINT LINE
*> GROUPNAME
         MVC   ZPRMCL16(Ø8),=C'GRPNAME='
         MVC   ZPRMCL16+Ø8(8),GRPNAME
         TRT   ZPRMCL16,TRTABLE            FIND FIRST BLANK
         MVI   Ø(1),C','                   PLUG COMMA HERE
         BAS   R14,ZWRTRTN              DO PRINT LINE
         MVC   ZPRMCL16(Ø9),=C'MEMBNAME='
         MVC   ZPRMCL16+Ø9(8),GRPMNAME
         TRT   ZPRMCL16,TRTABLE            FIND FIRST BLANK
         MVI   ZPRMCL72,C' '               PLUG LAST CARD HERE
         BAS   R14,ZWRTRTN              DO PRINT LINE
         TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.1Ø.98                *
               END ROUTINE               '
FINALIZE DS    ØH
         BAS   R14,CLOSRTN
         CLC   RETCODE,=H'12'
         BE    RETCOD12
         CLC   RETCODE,=H'16'
         BE    RETCOD16
         L     R13,SAVEAREA+4
```

```
         LM    R14,R12,12(R13)
         L     R15,Ø
         BR    R14
RETCOD12 DS    ØH
         L     R13,SAVEAREA+4
         LM    R14,R12,12(R13)
         L     R15,12
         BR    R14
RETCOD16 DS    ØH
         L     R13,SAVEAREA+4
         LM    R14,R12,12(R13)
         L     R15,16
         BR    R14
         TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.1Ø.98                *
               SUB ROUTINE              '
**       S U B - R O U T I N E S
** MAINRTN - MAIN PROCESSING
**           MOVE INPUT FIELDS TO OUTPUT FIELDS
MAINRTN  DS    ØH
         ST    R14,MAINSAVE
MAINEXIT EQU   *
         L     R14,MAINSAVE
         BR    R14
** INITRTN - INITIALIZE VALUES
INITRTN  DS    ØH
         ST    R14,INITSAVE
         OPEN  (SYSPRINT,OUTPUT)
         OPEN  LOAD
         OPEN  (SNAPDUMP,OUTPUT)
INITEXIT EQU   *
         L     R14,INITSAVE
         BR    R14
** GETPRTN - GET PARMLIST
GETPRTN  DS    ØH
         ST    R14,GETPSAVE
         L     R6,Ø(R1)              GET PARMLIST ADDRESS
         MVC   PARMLEN,Ø(R6)         GET PARMLIST LENGTH
         LH    R1,Ø(,R6)
         BCTR  R1,Ø
         MVC   PARMVAL,=CL8' '
         MVC   Ø+PARMVAL,2(R6)         GET PARMLIST VAL
         EX    R1,*-6                  AND FILL WITH BLANKS TO LENGTH 8
GETPEXIT EQU   *
         L     R14,GETPSAVE
         BR    R14
** HDRLRTN - PRINT HEADER LINES
HDRLRTN  DS    ØH
         ST    R14,HDRLSAVE
         MVC   ZPRMLINE,=(&ZPRMLNE)C'='
         BAS   R14,WRITRTN
```

```
         MVC   ZPRMLINE,STITLEL1
         BAS   R14,WRITRTN
HDRLEXIT EQU   *
         L     R14,HDRLSAVE
         BR    R14
** CLOSRTN - CLOSE FILES
CLOSRTN  DS    ØH
         ST    R14,CLOSSAVE
         CLOSE SYSPRINT
         CLOSE LOAD
         CLOSE SNAPDUMP
CLOSEXIT EQU   *
         L     R14,CLOSSAVE
         BR    R14
** WRITRTN - WRITE TO SYSPRINT
WRITRTN  DS    ØH
         ST    R14,WRITSAVE
         PUT   SYSPRINT,ZPRMLINE
         MVC   ZPRMLINE,=(&ZPRMLNE)C' '
WRITEXIT EQU   *
         L     R14,WRITSAVE
         BR    R14
** ZWRTRTN - WRITE TO SYSPRINT
ZWRTRTN  DS    ØH
         ST    R14,ZWRTSAVE
         PUT   SYSPRINT,ZPRMLINE
         MVC   ZPRMLINE,=(&ZPRMLNE)C' '
         MVI   ZPRMCL72,C'X'
ZWRTEXIT EQU   *
         L     R14,ZWRTSAVE
         BR    R14
** DZERORTN - DROP LEADING ZEROS
DZERORTN DS    ØH
         ST    R14,DZERSAVE
DZERO1ØØ CLI   ZEROHOL1,X'FØ'              IF ZERO
         BNE   DZEREXIT                    N.  EXIT WE ARE DONE
         MVC   ZEROHOLD,ZEROHOLD+1         Y.  SHIFT 1BYTE LEFT
         B     DZERO1ØØ                    CHECK NEXT BYTE
DZEREXIT EQU   *
         L     R14,DZERSAVE
         BR    R14
** BIT16RTN - TEST BIT 16 AND CONVERT TO NUMERIC
BIT16RTN DS    ØH
         ST    R14,BIT16SAV
         MVC   WORKCHAR,=56CL1' '          SET TO BLANK
         LA    R9,WORKCHAR
         MVI   Ø(R9),C'('                  PLUG LEFT PARENTHESIS
BIT16CØ1 TM    WORKHEX1,B'1ØØØØØØØ'        IF BIT ON
         BNO   BIT16CØ2                    N. GO ON
         MVC   1(2,R9),=C'1,'              Y. PLUG NUMBER AND COMMA
```

```
         LA    R9,2(R9)                  BUMP 2
BIT16CØ2 TM    WORKHEX1,B'Ø1ØØØØØØ'      IF BIT ON
         BNO   BIT16CØ3                  N. GO ON
         MVC   1(2,R9),=C'2,'            Y. PLUG NUMBER AND COMMA
         LA    R9,2(R9)                  BUMP 2
BIT16CØ3 TM    WORKHEX1,B'ØØ1ØØØØØ'      IF BIT ON
         BNO   BIT16CØ4                  N. GO ON
         MVC   1(2,R9),=C'3,'            Y. PLUG NUMBER AND COMMA
         LA    R9,2(R9)                  BUMP 2
BIT16CØ4 TM    WORKHEX1,B'ØØØ1ØØØØ'      IF BIT ON
         BNO   BIT16CØ5                  N. GO ON
         MVC   1(2,R9),=C'4,'            Y. PLUG NUMBER AND COMMA
         LA    R9,2(R9)                  BUMP 2
BIT16CØ5 TM    WORKHEX1,B'ØØØØ1ØØØ'      IF BIT ON
         BNO   BIT16CØ6                  N. GO ON
         MVC   1(2,R9),=C'5,'            Y. PLUG NUMBER AND COMMA
         LA    R9,2(R9)                  BUMP 2
BIT16CØ6 TM    WORKHEX1,B'ØØØØØ1ØØ'      IF BIT ON
         BNO   BIT16CØ7                  N. GO ON
         MVC   1(2,R9),=C'6,'            Y. PLUG NUMBER AND COMMA
         LA    R9,2(R9)                  BUMP 2
BIT16CØ7 TM    WORKHEX1,B'ØØØØØØ1Ø'      IF BIT ON
         BNO   BIT16CØ8                  N. GO ON
         MVC   1(2,R9),=C'7,'            Y. PLUG NUMBER AND COMMA
         LA    R9,2(R9)                  BUMP 2
BIT16CØ8 TM    WORKHEX1,B'ØØØØØØØ1'      IF BIT ON
         BNO   BIT16CØ9                  N. GO ON
         MVC   1(2,R9),=C'8,'            Y. PLUG NUMBER AND COMMA
         LA    R9,2(R9)                  BUMP 2
BIT16CØ9 TM    WORKHEX2,B'1ØØØØØØØ'      IF BIT ON
         BNO   BIT16C1Ø                  N. GO ON
         MVC   1(2,R9),=C'9,'            Y. PLUG NUMBER AND COMMA
         LA    R9,2(R9)                  BUMP 2
BIT16C1Ø TM    WORKHEX2,B'Ø1ØØØØØØ'      IF BIT ON
         BNO   BIT16C11                  N. GO ON
         MVC   1(3,R9),=C'1Ø,'           Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                  BUMP 2
BIT16C11 TM    WORKHEX2,B'ØØ1ØØØØØ'      IF BIT ON
         BNO   BIT16C12                  N. GO ON
         MVC   1(3,R9),=C'11,'           Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                  BUMP 2
BIT16C12 TM    WORKHEX2,B'ØØØ1ØØØØ'      IF BIT ON
         BNO   BIT16C13                  N. GO ON
         MVC   1(3,R9),=C'12,'           Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                  BUMP 2
BIT16C13 TM    WORKHEX2,B'ØØØØ1ØØØ'      IF BIT ON
         BNO   BIT16C14                  N. GO ON
         MVC   1(3,R9),=C'13,'           Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                  BUMP 2
BIT16C14 TM    WORKHEX2,B'ØØØØØ1ØØ'      IF BIT ON
```

```
         BNO   BIT16C15                    N. GO ON
         MVC   1(3,R9),=C'14,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT16C15 TM    WORKHEX2,B'00000010'        IF BIT ON
         BNO   BIT16C16                    N. GO ON
         MVC   1(3,R9),=C'15,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT16C16 TM    WORKHEX2,B'00000001'        IF BIT ON
         BNO   BIT16END                    N. GO ON
         MVC   1(3,R9),=C'16,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT16END MVI   0(R9),C')'                 PLUG RIGHT PAREN
         L     R14,BIT16SAV
         BR    R14
** BIT32RTN - TEST BIT 17 - 32 AND CONVERT TO NUMERIC
BIT32RTN DS    0H
         ST    R14,BIT32SAV
BIT32C17 TM    WORKHEX3,B'10000000'         IF BIT ON
         BNO   BIT32C18                    N. GO ON
         MVC   1(3,R9),=C'17,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C18 TM    WORKHEX3,B'01000000'         IF BIT ON
         BNO   BIT32C19                    N. GO ON
         MVC   1(3,R9),=C'18,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C19 TM    WORKHEX3,B'00100000'         IF BIT ON
         BNO   BIT32C20                    N. GO ON
         MVC   1(3,R9),=C'19,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C20 TM    WORKHEX3,B'00010000'         IF BIT ON
         BNO   BIT32C21                    N. GO ON
         MVC   1(3,R9),=C'20,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C21 TM    WORKHEX3,B'00001000'         IF BIT ON
         BNO   BIT32C22                    N. GO ON
         MVC   1(3,R9),=C'21,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C22 TM    WORKHEX3,B'00000100'         IF BIT ON
         BNO   BIT32C23                    N. GO ON
         MVC   1(3,R9),=C'22,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C23 TM    WORKHEX3,B'00000010'         IF BIT ON
         BNO   BIT32C24                    N. GO ON
         MVC   1(3,R9),=C'23,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C24 TM    WORKHEX3,B'00000001'         IF BIT ON
         BNO   BIT32C25                    N. GO ON
         MVC   1(3,R9),=C'24,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C25 TM    WORKHEX4,B'10000000'         IF BIT ON
```

```
         BNO   BIT32C26                    N. GO ON
         MVC   1(3,R9),=C'25,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C26 TM    WORKHEX4,B'01000000'        IF BIT ON
         BNO   BIT32C27                    N. GO ON
         MVC   1(3,R9),=C'26,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C27 TM    WORKHEX4,B'00100000'        IF BIT ON
         BNO   BIT32C28                    N. GO ON
         MVC   1(3,R9),=C'27,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C28 TM    WORKHEX4,B'00010000'        IF BIT ON
         BNO   BIT32C29                    N. GO ON
         MVC   1(3,R9),=C'28,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C29 TM    WORKHEX4,B'00001000'        IF BIT ON
         BNO   BIT32C30                    N. GO ON
         MVC   1(3,R9),=C'29,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C30 TM    WORKHEX4,B'00000100'        IF BIT ON
         BNO   BIT32C31                    N. GO ON
         MVC   1(3,R9),=C'30,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C31 TM    WORKHEX4,B'00000010'        IF BIT ON
         BNO   BIT32C32                    N. GO ON
         MVC   1(3,R9),=C'31,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32C32 TM    WORKHEX4,B'00000001'        IF BIT ON
         BNO   BIT32END                    N. GO ON
         MVC   1(3,R9),=C'32,'             Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                    BUMP 2
BIT32END MVI   0(R9),C')'                  PLUG RIGHT PAREN
         L     R14,BIT32SAV
         BR    R14
** ABENDRTN - HANDLE SOFT ABEND
ABNDRTN  DS    0H
         SNAP  DCB=SNAPDUMP,STORAGE=(WSSTART,WSEND)
         MVC   RETCODE,=H'16'
         B     FINALIZE
         TITLE 'LOEBEN - ZPARM DB2 V4 CREATE - 27.10.98                *
               CONSTANTS                '
**       W O R K I N G   S T O R A G E
ZPARMGV@ CSECT
SAVEAREA DC    18F'0'
WSSTART  DS    0F
         DC    CL16'WORKING STORAGE '
WRKPFLG1 DC    AL1(0)
         DS    0D
D        DS    D
FULLWORD DC    F'1024'
```

```
** PARMLIST VALUES
PARMLEN  DS    XL2
PARMVAL  DS    CL8
** SQUEEZE ZERO HOLD AREA
ZEROHOLD DS    ØCL16
ZEROHOL1 DC    C' ',CL15' ',C'  '
** ZPARM OUTPUT LINE
ZPRMLINE DS    ØCL(&ZPRMLNE)        SYSPRINT RECORD
         DS    CLØ4                 FIELD STARTING IN COL 1
ZPRMCLØ5 DS    CL11                 FIELD STARTING IN COL 5
ZPRMCL16 DS    CL56                                   COL 16
ZPRMCL72 DS    CLØ1                                   COL 72
ZPRMCL73 DS    CLØ8                                   COL 8Ø
RETCODE  DS    H'Ø'
         DC    (&ZPRMLNE-*+ZPRMLINE)C' '
** BIT TEST WORK AREA
WORKCHAR DS    ØCL56
WORKCHR1 DS    C
WORKCHRS DS    CL55
WORKB16  DS    ØXL2
WORKB32  DS    ØXL4
WORKHEX1 DS    XL1
WORKHEX2 DS    XL1
WORKHEX3 DS    XL1
WORKHEX4 DS    XL1
**       R O U T I N E   S A V E A R E A S
MAINSAVE DS    F'Ø'
INITSAVE DS    F'Ø'
GETPSAVE DS    F'Ø'
CLOSSAVE DS    F'Ø'
WRITSAVE DS    F'Ø'
ZWRTSAVE DS    F'Ø'
DZERSAVE DS    F'Ø'
HDRLSAVE DS    F'Ø'
BIT16SAV DS    F'Ø'
BIT32SAV DS    F'Ø'
         PRINT NOGEN
ABEND1ØØ ABEND 1ØØ                    LOAD FAILED
ABEND1Ø1 ABEND 1Ø1,DUMP
ABEND1Ø2 ABEND 1Ø2,DUMP
ABEND1Ø3 ABEND 1Ø3,DUMP               DSN6ARVP EYE CATCHER NOT FOUND
ABEND1Ø4 ABEND 1Ø4,DUMP               DSN6LOGP EYE CATCHER NOT FOUND
ABEND1Ø5 ABEND 1Ø5,DUMP               DSN6SYSP EYE CATCHER NOT FOUND
ABEND1Ø6 ABEND 1Ø6,DUMP               DSN6FAC  EYE CATCHER NOT FOUND
ABEND1Ø7 ABEND 1Ø7,DUMP               DSN6GRP  EYE CATCHER NOT FOUND
ABEND19Ø ABEND 19Ø,DUMP               DSN6SYSP NOT FOUND IN DIRECTORY
ABEND191 ABEND 191,DUMP               DSN6ARVP NOT FOUND IN DIRECTORY
ABEND192 ABEND 192,DUMP               DSN6LOGP NOT FOUND IN DIRECTORY
ABEND193 ABEND 193,DUMP               DSN6SYSP NOT FOUND IN DIRECTORY
ABEND194 ABEND 194,DUMP               DSN6LOGP NOT FOUND IN DIRECTORY
```

```
ABEND195 ABEND 195,DUMP                DSN6SYSP NOT FOUND IN DIRECTORY
ABEND196 ABEND 196,DUMP                DSN6FAC  NOT FOUND IN DIRECTORY
ABEND197 ABEND 197,DUMP                DSN6SYSP NOT FOUND IN DIRECTORY
ABEND198 ABEND 198,DUMP                DSN6SYSP NOT FOUND IN DIRECTORY
ABEND199 ABEND 199,DUMP                DSN6SYSP NOT FOUND IN DIRECTORY
SYSPRINT DCB
DSORG=PS,RECFM=F,MACRF=(PM),DDNAME=SYSPRINT,LRECL=&ZPRMLN18830006
               NE
LOAD     DCB   DSORG=PO,DDNAME=LOAD,MACRF=R
SNAPDUMP DCB   DSORG=PS,RECFM=VBA,MACRF=W,LRECL=125,BLKSIZE=882,       X
               DDNAME=SNAPDUMP
**       STRING CONSTANTS
STITLEL1 DC    CL&ZPRMLNE'                DISPLAY DSNZPARM VALUES'
WSEND    DC    CL08'WSEND>>'
ZPARMPTR DS    3A                   ZPARM LOAD ADDRESS, LENGTH, END
         LTORG
TRTABLE  DC    XL256'0'
         ORG   TRTABLE+X'40'
         DC    X'40'                 BLANK
         ORG
         END
//SYSLIB   DD  DISP=SHR,DSN=DSN.SKA.I410.SDSNMACS
//*YSLIB   DD  DISP=SHR,DSN=DSN.I510.DSN510.SDSNMACS
//*        DD  DISP=SHR,DSN=DSN.I510.DSN510.SDSNSAMP
//         DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSLIN   DD  DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(800,(500,500)),DCB=(BLKSIZE=800)
//SYSPRINT DD  SYSOUT=* 2,OUTPUT=*.BA09
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(800,(500,500),,,ROUND)
//SYSUT2   DD  UNIT=SYSDA,SPACE=(800,(500,500),,,ROUND)
//SYSUT3   DD  UNIT=SYSDA,SPACE=(800,(500,500),,,ROUND)
//*
//L       EXEC PGM=IEWL,PARM='LIST,XREF,NCAL,RENT',COND=(4,LT,A)
//SYSLIN   DD  DISP=(OLD,DELETE),DSN=&&LOADSET
//SYSLMOD  DD  DISP=(,PASS),DSN=&&LOAD(ZPARMREE),
//             SPACE=(TRK,(50,50,2)),UNIT=SYSDA
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(1024,(50,50))
//X       EXEC PGM=*.L.SYSLMOD,COND=(4,LT),PARM=DB2P
//LOAD     DD  DISP=SHR,DSN=DB24.DSNLOAD
//*LOAD    DD  DISP=SHR,DSN=DB25.SDSNLOAD
//SYSPRINT DD  SYSOUT=*
//SNAPDUMP DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//
```

Rolf Loeben (Germany) © Xephon 1999

Analysing and displaying RI structures

It is not easy to get an overview of relationships among tables defined with referential constraints. This EXEC analyses parent–child dependencies from the catalog table SYSIBM.SYSRELS and presents the referential structure in a hierarchical manner. Referential cycles will be detected and displayed with an error message.

For example, for a relationship such as that shown in Figure 1, the RI structure shown in Figure 2 will be displayed.

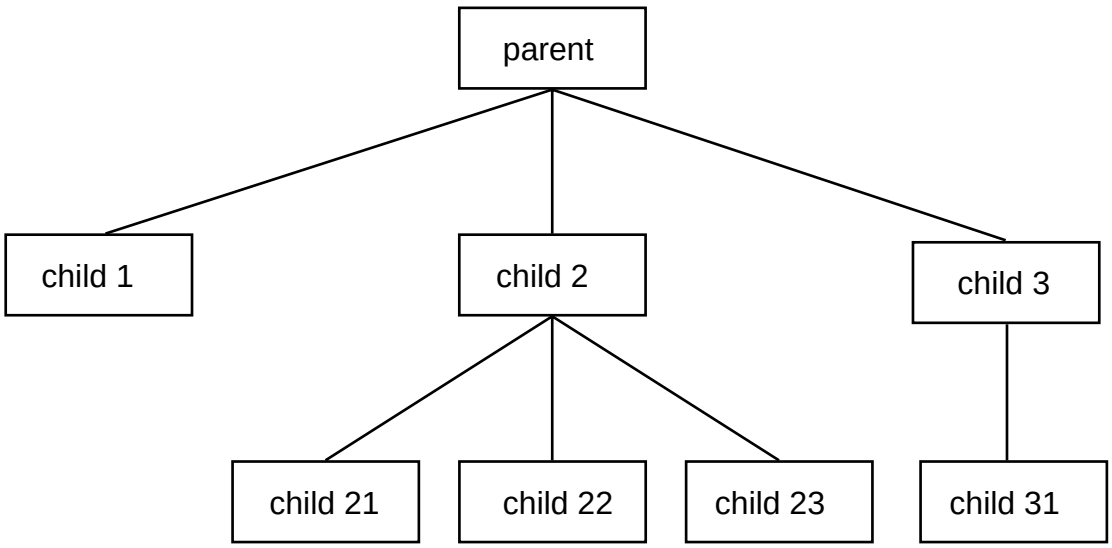

Figure 1: Parent–child relationship

The application comprises one REXX EXEC and three ISPF panels for displaying the required data. EXEC DB2RELS receives control and retrieves the SSID from a select service menu that has the DB2 libraries (SDSNLOAD and RUNLIB.LOAD) already allocated. Initially a selection panel (DB2RELS) asks for the table owner, table name (DB2 wildcards are allowed), and the display version. After receiving the search conditions, parent tables of referential constraints

```
                                                    Row 1 to 8 of 8
   DISPLAY REFERENTIAL INTEGRITY STRUCTURE    SSID  - ssid
                                              User  - user
                                              Date  - Ø3.Ø3.99
                                              Time  - 13:41
  COMMAND ===>                               SCROLL ===> PAGE
 ──────────────────────────────────────────────────────────────────────
 LEVEL               TABLE         RELNAME           DELETERULE
  Ø.............     PARENT
  .1............     CHILD1        CHILD1PARENT      CASCADE
  .2............     CHILD2        CHILD2PARENT      RESTRICT
  ..1...........     CHILD21       CHILD21CHILD2     CASCADE
  ..2...........     CHILD22       CHILD22CHILD2     CASCADE
  ..3...........     CHILD23       CHILD23CHILD2     CASCADE
  .3............     CHILD3        CHILD3PARENT      RESTRICT
  ..1...........     CHILD31       CHILD31CHILD3     RESTRICT
****************************** Bottom of data ***********************
```

Figure 2: Example of display

are retrieved from the SYSIBM.SYSRELS table via a TSO/DSN session using the DSNTIAUL sample unload table program. Temporary datasets will be allocated for the input and output datasets used by the unload program. The SQL result set is stored in a temporary ISPF table named PTAB. Having selected Version 1, the result set contains only top level parent tables.

For Version 2, all matching parent tables are displayed. A table display for panel DB2RELS2 is then opened. After entering one or more selection marks, the display process for the RI structure is initiated. The structure is analysed using the SELECT statement recursively against the SYSIBM.SYSRELS table until no further dependent tables are found. The RI cycle check is performed for every result row.

The result rows are then sorted, using a bubble sort, into hierarchical order and stored in ISPF table RITAB and displayed. Initially, the number of digits for variable LEVEL is set to one (a maximum of nine parent–child relationships) but can be increased by setting variable LD to two (a maximum of 99 parent–child relationships). Returning (PF3 – End) to the selection screen (DB2RELS) will delete all ISPF tables and datasets.

REXX EXEC DB2RELS

```
/* REXX */
/*====================================================================-/
/- Function: Display hierarchical view of DB2 referential table       -/
/-           constraints. RI-cycles will be displayed with an         -/
/-           error message.                                           -/
/-====================================================================-*/
arg dbs .                                        /* parm=DB2 subsystem */
db2id = dbs                                      /* init panel SSID fld*/
lp    = 2Ø               /* max pos available for level display     */
ld    = 1                /* level digits                            */
                         /* 1 = max  9 parent-child hierarchy depth */
                         /* 2 = max 99 parent-child hierarchy depth */
own = ,OWNER'                                 /* init table owner       */
version = ,1'                                 /* init display version   */
                                              /* 1 = start on top level */
                                              /* 2 = start on any level */

tbn     = ,'
msg     = ,'
amt     = ,PAGE'                                 /* init scroll amount */
do forever
    ADDRESS ISPEXEC ,DISPLAY PANEL (DB2RELS)'
    if rc = 8 then
       leave                                                /* pf3 - end  */
    else
     do
       allocrc = Ø                 /* rc of TSO ALLOC cmds           */
       counter = Ø                 /* no of table descriptions displ */
       x = outtrap(var.,'*')       /* trap TSO command output        */
       call alloc                  /* alloc temp data sets for SELECT*/
       if allocrc = Ø then
          do
            call db2select1        /* select parent tables           */
            if counter > Ø then    /* tables for owner found         */
               do
                 ADDRESS ISPEXEC ,TBTOP   PTAB'
                 ADDRESS ISPEXEC ,TBDISPL PTAB PANEL(DB2RELS2)'
                 if rc  < 8 then                        /* tab select */
                    do
                      ADDRESS ISPEXEC ,CONTROL DISPLAY SAVE'
                      call ristructure    /* analyse RI structure     */
                      call ritabloop      /* first parent tab select  */
                      call ptabloop       /* more parent tabs select  */
                    end
                 ADDRESS ISPEXEC ,TBEND   PTAB'
                 call dealloc                /* delete temp datasets  */
                 msg = , , || counter ,table(s) displayed |'
               end
            else
```

```
               do
                 call dealloc                /* delete temp datasets  */
                 msg = , Table(s) not found |'
               end
          end
     end
end
exit
/*-------------------------------------------------------------------- /
/- subroutine: alloc temp datasets for DB2 select processing          -/
/-             dsntemp1 = SYSIN                                       -/
/-             dsntemp2 = SYSREC00                                    -/
/-             dsntemp3 = SYSPRINT                                    -/
/-------------------------------------------------------------------- */
alloc:
/* time = TIME(,s')  */
time = ,00000'
dsntemp1 = USERID() || ,.DB2RELS' || ,.A' || time
dsntemp2 = USERID() || ,.DB2RELS' || ,.B' || time
dsntemp3 = USERID() || ,.DB2RELS' || ,.C' || time
/*                                          allocate temp dataset 1    */
ADDRESS TSO "ALLOC FI("temp1") DA(,"dsntemp1"') OLD REUSE"
if rc = 0 then
   ADDRESS TSO "EXECIO 0 DISKW "temp1" (OPEN FINIS"
else
   ADDRESS TSO "ALLOC FI("temp1") DA(,"dsntemp1"'),
   NEW CAT REUSE UNIT(SYSTS),
   LRECL(80) BLKSIZE(2792) RECFM(F B) SPACE(1,1) TRACKS"
   if rc ¬= 0 then
      do
        msg = ,Temporary dataset 1 cannot be allocated |'
        allocrc = 1
        return
      end
/*                                          allocate temp dataset 2    */
ADDRESS TSO "ALLOC FI("temp2") DA(,"dsntemp2"') OLD REUSE"
if rc = 0 then
   ADDRESS TSO "EXECIO 0 DISKW "temp2" (OPEN FINIS"
else
   ADDRESS TSO "ALLOC FI("temp2") DA(,"dsntemp2"'),
   NEW CAT REUSE UNIT(SYSTS),
   LRECL(100) BLKSIZE(2790) RECFM(F B) SPACE(1,1) TRACKS"
   if rc ¬= 0 then
      do
        msg = ,Temporary dataset 2 cannot be allocated |'
        allocrc = 1
        return
      end
/*                                          allocate temp dataset 3    */
ADDRESS TSO "ALLOC FI("temp3") DA(,"dsntemp3"') OLD REUSE"
```

```
if rc = Ø then
   ADDRESS TSO "EXECIO Ø DISKW "temp3" (OPEN FINIS"
else
   ADDRESS TSO "ALLOC FI("temp3") DA(,"dsntemp3"'),
   NEW CAT REUSE UNIT(SYSTS),
   LRECL(133) BLKSIZE(2793Ø) RECFM(F B) SPACE(1,1) TRACKS"
   if rc ¬= Ø then
      do
        msg = ,Temporary dataset 3 cannot be allocated |'
        allocrc = 1
        return
      end
return
/*———————————————————————————————————————————————————————————————— /
/- subroutine: dealloc and delete temp data sets                       -/
/———————————————————————————————————————————————————————————————— */
dealloc:
ADDRESS TSO "FREE  FI("temp1")"
ADDRESS TSO "FREE  FI("temp2")"
ADDRESS TSO "FREE  FI("temp3")"
ADDRESS TSO "DELETE (,"dsntemp1"')"
ADDRESS TSO "DELETE (,"dsntemp2"')"
ADDRESS TSO "DELETE (,"dsntemp3"')"
return
/*———————————————————————————————————————————————————————————————— /
/- subroutine: display parent table(s) of RI                           -/
/———————————————————————————————————————————————————————————————— */
ptabloop:
do forever
   ADDRESS ISPEXEC ,CONTROL DISPLAY RESTORE'
   ADDRESS ISPEXEC ,TBDISPL PTAB'
   select
     when rc = 8 then leave                          /* pf3 - end  */
     when rc = 4 then                                /* more rows  */
        do
          ADDRESS ISPEXEC ,CONTROL DISPLAY SAVE'
          call ristructure
          call ritabloop
        end
     when rc = Ø then                                /* last row   */
        do
          ADDRESS ISPEXEC ,CONTROL DISPLAY SAVE'
          call ristructure
          call ritabloop
        end
   end
end
return
/*———————————————————————————————————————————————————————————————— /
/- subroutine: display RI structure  of selected hierarchy            -/
```

```
/------------------------------------------------------------------------- */
ritabloop:
ADDRESS ISPEXEC ,TBTOP   RITAB'
do forever
   ADDRESS ISPEXEC ,TBDISPL RITAB PANEL(DB2RELS3)'
   if rc = 8 then  leave                              /* pf3 - end  */
end
ADDRESS ISPEXEC ,TBEND   RITAB'
return
/*------------------------------------------------------------------------- /
/- subroutine: analyse RI structure                                        -/
/------------------------------------------------------------------------- */
ristructure:
ristruc. = ,'                           /* clear structure table     */
j = 1                                   /* init structure depth      */
level1 = ,'                             /* init parent-child concat  */
level2 = right(,Ø',ld,'Ø')              /* init level depth          */
ristruc.j =  level2 tname , , , , , ,  /* ins top level parent tab  */
do k = 1 until ristruc.k = ,'           /* loop until eot            */
   parse var ristruc.k level reftbname . /* next arg from struc tab   */
   level1 = level                       /* parent-child concatenation*/
   call db2select2                      /* select from SYSRELS       */
end
call ristrucsort                        /* sort structure table      */
call ritabcreate                        /* create ISPF table         */
return
/*------------------------------------------------------------------------- /
/- subroutine: sort RI structure via bubble sort                           -/
/------------------------------------------------------------------------- */
ristrucsort:
do i = 1 to j - 1
   do k = 1 to j - i
      parse var ristruc.k levelk .
      n = k + 1
      parse var ristruc.n leveln .
      if left(levelk,lp,'Ø') > left(leveln,lp,'Ø') then
         do
           temp     = ristruc.k             /* exchange entries   */
           ristruc.k = ristruc.n            /* via temp field     */
           ristruc.n = temp
         end
   end
end
return
/*------------------------------------------------------------------------- /
/- subroutine: create ISPF table for RI structure table                    -/
/------------------------------------------------------------------------- */
ritabcreate:
ADDRESS ISPEXEC ,TBCREATE RITAB,
                NAMES(LEVEL TABLE RELNAME RULE),
```

```
                NOWRITE REPLACE'
do i=1 to j
   parse var ristruc.i level table relname rule .
   l = length(level) - ld
   level = overlay(,.',level,,l,'.')            /* overlay leading  */
   level       = left(level,lp,'.')             /* numbers with ,.' */
   select
       when rule = ,R' then rule = ,RESTRICT'
       when rule = ,C' then rule = ,CASCADE'
       when rule = ,N' then rule = ,SET NULL'
       otherwise nop
   end
   ADDRESS ISPEXEC ,TBADD RITAB MULT(, || j || ,)'
end
return
/*-------------------------------------------------------------------- /
/- subroutine: get parent table(s) of RI from SYSRELS                  -/
/- version=1: display begins on parent tables that are not dependent  -/
/- version=2: display begins on any parent tables                     -/
/-------------------------------------------------------------------- */
db2select1:
/*                                        store select into SYSIN */
V.    = ,'
if version = ,1' then                     /* start on  top level  */
  do
    V.1   = "  SELECT                                          "
    V.2   = "  DISTINCT (SUBSTR(A.REFTBNAME, 1, 18))          "
    V.3   = "  FROM SYSIBM.SYSRELS A                          "
    V.4   = "  WHERE A.REFTBCREATOR = ," || own || ","
    if tbn = ,' then
       V.5   = "  AND NOT EXISTS                             "
    else
       V.5   = "  AND A.REFTBNAME LIKE ," || tbn || ", AND NOT EXISTS"
    V.6   = "  (SELECT 1                                      "
    V.7   = "   FROM SYSIBM.SYSRELS B                         "
    V.8   = "   WHERE A.REFTBNAME = B.TBNAME)                 "
    V.9   = "  ORDER BY 1;                                    "
  end
if version = ,2' then                     /* start on any level   */
  do
    V.1   = "  SELECT                                          "
    V.2   = "  DISTINCT (SUBSTR(A.REFTBNAME, 1, 18))          "
    V.3   = "  FROM SYSIBM.SYSRELS A                          "
    V.4   = "  WHERE A.REFTBCREATOR = ," || own || ","
    if tbn = ,' then
       V.5   = " "
    else
       V.5   = "  AND A.REFTBNAME LIKE ," || tbn || ","
    V.6   = "  ORDER BY 1;                                    "
  end
```

```
ADDRESS TSO "EXECIO * DISKW "temp1" (STEM V. "
ADDRESS TSO "EXECIO Ø DISKW "temp1" (FINIS"
/*                                                    call DB2 via DSN */
ADDRESS TSO "ALLOC FI("sysin") DA(,"dsntemp1"') OLD REUSE"
ADDRESS TSO "ALLOC FI("sysrecØØ") DA(,"dsntemp2"') OLD REUSE"
ADDRESS TSO "ALLOC FI("sysprint") DA(,"dsntemp3"') OLD REUSE"
ADDRESS TSO "ALLOC FI("syspunch") DUMMY"
push "END"
push "RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) PARMS(,SQL')"
ADDRESS TSO "DSN SYSTEM(" || dbs || ")"
ADDRESS TSO "EXECIO * DISKR "sysrecØØ" (STEM rowin. FINIS"
ADDRESS TSO "FREE FI("sysin")"
ADDRESS TSO "FREE FI("sysrecØØ")"
ADDRESS TSO "FREE FI("syspunch")"
ADDRESS TSO "FREE FI("sysprint")"
                                        /* create table and add rows */
if rowin.Ø > Ø then
   do
     ADDRESS ISPEXEC ,TBCREATE PTAB KEYS(TNAME) NOWRITE REPLACE'
     do i=1 to rowin.Ø
        parse var rowin.i tname 19 .
        ADDRESS ISPEXEC ,TBADD PTAB MULT(, || rowin.Ø || ,)'
        counter = counter + 1
     end
   end
return
/*———————————————————————————————————————————————————————————————————— /
/- subroutine: get dependent table(s) of RI from SYSIBM.SYSRELS       -/
/———————————————————————————————————————————————————————————————————— */
db2select2:
/*                                          store select into SYSIN */
V.    = ,'
V.1   = "  SELECT                        "
V.2   = "  SUBSTR(TBNAME, 1, 18),        "
V.3   = "  RELNAME,                      "
V.4   = "  DELETERULE                    "
V.5   = "  FROM SYSIBM.SYSRELS           "
V.6   = "  WHERE REFTBNAME = ," || reftbname || ","
V.7   = "  ORDER BY 1;"
ADDRESS TSO "EXECIO * DISKW "temp1" (STEM V. "
ADDRESS TSO "EXECIO Ø DISKW "temp1" (FINIS"
/*                                                    call DB2 via DSN */
ADDRESS TSO "ALLOC FI("sysin") DA(,"dsntemp1"') OLD REUSE"
ADDRESS TSO "ALLOC FI("sysrecØØ") DA(,"dsntemp2"') OLD REUSE"
ADDRESS TSO "ALLOC FI("sysprint") DA(,"dsntemp3"') OLD REUSE"
ADDRESS TSO "ALLOC FI("syspunch") DUMMY"
push "END"
push "RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) PARMS(,SQL')"
ADDRESS TSO "DSN SYSTEM(" || dbs || ")"
ADDRESS TSO "EXECIO * DISKR "sysrecØØ" (STEM rowin. FINIS"
```

```
ADDRESS TSO "FREE FI("sysin")"
ADDRESS TSO "FREE FI("sysrecØØ")"
ADDRESS TSO "FREE FI("syspunch")"
ADDRESS TSO "FREE FI("sysprint")"
                                        /* stem rows into            */
                                        /* structure table           */
if rowin.Ø > Ø then
   do
     level2 = Ø                         /* init level depth          */
     do i=1 to rowin.Ø
        parse var rowin.i table 18 relname 27 rule 28 .
        call cyclecheck
        level2 = level2 + 1                  /* next level depth     */
        level2 = right(level2,ld,'Ø')        /* format level digit   */
        level  = level1 || level2            /* parent-child concat  */
        j = j + 1
        ristruc.j = level table relname rule reftbname
     end
   end
return
/*-------------------------------------------------------------------- /
/- subroutine: check for RI cycles                                    -/
/-------------------------------------------------------------------- */
cyclecheck:
do c = 1 to j
   parse var ristruc.c . . . . ctable .
   if table = ctable then
      do
        table   = strip(table) || ,_HAS_RI-CYCLE|'
        leave
      end
end
return
```

PANEL DB2RELS

```
)Body
%               DISPLAY REFERENTIAL TABLE CONSTRAINTS        SSID   -
+&db2id
%                                                            User   -
+&ZUSER
%                                                            Date   -
+&date
% COMMAND ===> _ZCMD                                        +%Time   -
+&ZTIME
%-------------------------------------------------------------------------
+
+
+             %Table owner ......................... :+ _OWN     +
```

```
+
+              %Table name (DB2 wildcards allowed) ....:+ _TBN
+
+
+              %Version ..............................:+ _VERSION +
+              %(1 = start on top parent level)+
+              %(2 = start on any parent level)+
+
+
+
+              &MSG
%-------------------------------------------------------------------------
%
%
)INIT
   .CURSOR = TBN
   &DATE    = ,&ZDAY..&ZMONTH..&ZYEAR'
)PROC
   VER (&own,nb,len,'<=',8)
   VER (&version,nb,list,1,2)
   IF  (&tbn > ,')
       VER (&tbn,nb,len,'<=',18)
)END
```

PANEL DB2RELS2

```
)ATTR
 § TYPE(OUTPUT) INTENS(LOW)
)BODY
%
%               DISPLAY PARENT TABLE(S) OF RI                 SSID   -
+&db2id
%                                                             User   -
+&ZUSER
%                                                             Date   -
+&date
%                                                            +%Time   -
+&ZTIME
% COMMAND ===> _ZCMD                                         +%SCROLL
===>_AMT +
%-------------------------------------------------------------------------
% S TNAME
)MODEL
 _Z§tname
)INIT
   .ZVARS  = ,(SEL)'
   &SEL    = ,'
   .CURSOR = ZCMD
   &DATE   = ,&ZDAY..&ZMONTH..&ZYEAR'
```

```
)REINIT
   IF   (.MSG = , ,)
        &SEL  = , ,
        REFRESH(SEL)
)PROC
   IF  (&ZTDSELS = Ø)
       .MSG = DBMØØØ
   else
       VER (&SEL,NB,LIST,S,s)
)END
```

PANEL DB2RELS3

```
)ATTR
 § TYPE(OUTPUT) INTENS(LOW)
)BODY
%
%               DISPLAY REFERENTIAL INTEGRITY STRUCTURE      SSID   -
+&db2id
%                                                            User   -
+&ZUSER
%                                                            Date   -
+&date
%                                                           +%Time   -
+&ZTIME
% COMMAND ===> _ZCMD                                        +%SCROLL
===>_AMT +
%———————————————————————————————————————————————————————————————————
% LEVEL                 TABLE                    RELNAME
DELETERULE
)MODEL
 §level               §table                   §relname        §rule
)INIT
   .CURSOR = ZCMD
   &DATE   = ,&ZDAY..&ZMONTH..&ZYEAR'
)PROC
)END
```

MESSAGE MEMBER DBM00

```
DBMØØØ  ,NO SELECTION ENTERED' .TYPE=WARNING
```

Raimund Kleebaur
DBA
Hugo Boss (Germany)

An extent checker – part 2

This month we conclude the article giving a tool that lists the VSAM DB2 LDSs belonging to a special DB2 instance or subsystem along with their extents in sorted form.

```
/* ROUTINE TO GET THE INCREASED PQTY AND SECQTY FOR */
/* THE PARTICULAR TABLESPACE OR A PART OF THE       */
/* TABLESPACE                                       */
/*   THE PQTY AND THE SECQTY IS PUMPED BY 30% AND   */
/*     20% RESPECTIVELY                             */
ALTERTS:

/* QUEUE THE JCL AND SUBMIT IT                      */
"ALLOC DS(USERID.JCLOUT)  FI(OUT) SHR"
NEWSTACK
QUEUE "//XXXXXJOB  JOB (ACCT PARAMETER),'COMMENTS',CLASS=A,         "
QUEUE "//  MSGCLASS=X,REGION=4096K,MSGLEVEL=(1,1),NOTIFY=&SYSUID    "
QUEUE "//*************************************************************"
QUEUE "//* TO GET THE PQTY AND SQTY OF THE TABLESPACE               "
QUEUE "//*************************************************************"
QUEUE "//STEP010 EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)          "
QUEUE "//STEPLIB DD DSN=SYS2.DB2.XXXX.DSNLOAD,DISP=SHR              "
QUEUE "//SYSTSPRT DD SYSOUT=*                                       "
QUEUE "//SYSPRINT DD SYSOUT=*                                       "
QUEUE "//SYSOUT   DD SYSOUT=*                                       "
QUEUE "//SYSREC00 DD DSN=XXXTEST.USERID.JCLOUT.DATA,DISP=SHR        "
QUEUE "//SYSPUNCH DD DUMMY                                          "
QUEUE "//SYSTSIN  DD *                                              "
QUEUE "    DSN SYSTEM(XXXX)                                         "
QUEUE "   RUN PROGRAM(DSNTIAUL) PARMS('SQL')                        "
QUEUE "   END                                                       "
QUEUE "/*                                                           "
IF PART = 1
 THEN
  DO
    QUEUE "//SYSIN    DD *                                            "
    QUEUE "   SET CURRENT DEGREE = 'ANY' ;                            "
    QUEUE "   SELECT '"VT"', ' ', '"DB"', ' ', '"SP"', ' ',           "
    QUEUE "          DIGITS(PARTITION), ' ',                          "
    QUEUE "          DIGITS(INTEGER(PQTY * 1.2) + (PQTY * 4)), ' ',   "
    QUEUE "          DIGITS(INTEGER(SQTY * .8) + (SQTY * 4))          "
    QUEUE "   FROM SYSIBM.SYSTABLEPART                                "
    QUEUE "   WHERE TSNAME = '"SP"' AND PARTITION IN (0,1);           "
    QUEUE "/*                                                         "
  END
 ELSE
  DO
```

```
    QUEUE "//SYSIN    DD *                                              "
    QUEUE "   SET CURRENT DEGREE = 'ANY' ;                              "
    QUEUE "   SELECT '"VT"', ' ', '"DB"', ' ', '"SP"', ' ',             "
    QUEUE "          DIGITS(PARTITION), ' ',                            "
    QUEUE "          DIGITS(INTEGER(PQTY * 1.2) + (PQTY * 4)), ' ',     "
    QUEUE "          DIGITS(INTEGER(SQTY * .8) + (SQTY * 4))            "
    QUEUE "   FROM SYSIBM.SYSTABLEPART                                  "
    QUEUE "   WHERE TSNAME = '"SP"' AND PARTITION = "PART";             "
    QUEUE "/*                                                           "
  END
QUEUE "//*************************************************************"
QUEUE "//* TO EXEC THE ALTER                                          "
QUEUE "//*************************************************************"
QUEUE "//STEP2Ø   EXEC PGM=IKJEFTØ1,REGION=4Ø96K                      "
QUEUE "//SYSEXEC  DD DSN=XXXTEST.USERID.REXXLIB,DISP=SHR              "
QUEUE "//SYSTSPRT DD SYSOUT=*                                         "
QUEUE "//SYSTSIN DD *                                                 "
QUEUE "   EXECUTIL SEARCHDD(YES)                                      "
QUEUE "   %ALTERTS                                                    "
 QUEUE "//*                                                           "
"EXECIO 36 DISKW OUT (FINIS"
"SUBMIT 'XXXTEST.USERID.JCLOUT'"
"FREE DD(OUT)"
DELSTACK
RETURN
```

ALTERTS

```
/* REXX TO ALTER THE TABLESPACE                      */
/* ALLOCATE THE FILE                                 */
/*  THE OUTPUT OF THE EXTENTS EXEC IS IN             */
/*   XXXTEST.USERID.JCLOUT.DATA                      */
   "ALLOC DS(USERID.JCLOUT.DATA)  FI(INP) SHR REU"
/* INITIALIZE THE ARRAY                              */
   DSNAME. = ''
/* READ THE FILE INTO THE ARRAY                      */
   "EXECIO * DISKR INP (STEM DSNAME."
/* SEGREGATE THE INPUT TO DIFFERENT VARIABLES        */
/* THE INPUT IS IN THE FORMAT AS GIVEN BELOW :       */
/* ..XXXTEST.. ..DATABASE.. ..SPACENAM.. ØØZZZ..     */
/*                          PQTY... SECQTY           */
/*   WHERE VCAT      - VCATNAME OF THE STGROUP       */
/*         DB        - NAME OF THE DATABASE          */
/*         SP        - SPACENAME                     */
/*         PART      - NUM OF THE PARTITION          */
/*         PQTY      - INCREASED PQTY WHICH IS       */
/*                       A MULTIPLE OF 4KB PAGES     */
/*         SECQTY    - INCREASED SECQTY WHICH IS     */
/*                       A MULTIPLE OF 4KB PAGES     */
   VCAT = SUBSTR(SUBWORD(DSNAME.1,1,1),3,7)
   DB   = SUBSTR(SUBWORD(DSNAME.1,2,1),3,8)
```

```
   SP   = SUBSTR(SUBWORD(DSNAME.1,3,1),3,8)
   PART = SUBSTR(SUBWORD(DSNAME.1,4,1),3,3)
   PQTY = SUBSTR(SUBWORD(DSNAME.1,5,1),1,1Ø)
   SQTY = SUBSTR(SUBWORD(DSNAME.1,6,1),1,1Ø)

/* TO CHECK IF THE SPACENAME IS IN THE CATALOG       */
   IF DB = ' '
      THEN DO
            SAY 'SPACENAME IS NOT FOUND IN THE CATALOG'
            EXIT
           END
   ELSE
      NOP

/* TO DISPLAY THE PARAMETERS TO BE USED IN THE JCL    */
   SAY 'VCAT      ' VCAT
   SAY 'DBNAME    ' DB
   SAY 'SPACENM   ' SP
   SAY 'PART      ' PART
   SAY 'PQTY      ' PQTY
   SAY 'SQTY      ' SQTY

/* TO CHECK FOR THE APPROPRIATE ROUTINE TO BE CALLED  */
   IF PART = Ø
      THEN
        DO
         SAY 'TO ALTER THE WHOLE TABLESPACE'
         CALL NPARTTS
        END
   ELSE
        DO
         SAY 'TO ALTER THE PARTITIONED TABLESPACE'
         CALL PARTTS
        END
   EXIT

/* ROUTINE TO ALTER THE WHOLE TABLESPACE              */
NPARTTS:
/* QUEUE THE JCL AND SUBMIT                          */
/*  THE JCL IS WRITTEN INTO XXXTEST.USERID.ALTER.JCL  */
/*    WHICH IS ALLOCATED WITH LRECL 8Ø                */

"ALLOC DS(USERID.ALTER.JCL)  FI(OUT) SHR"
NEWSTACK
QUEUE "//XXXXXJOB  JOB (ACCT PARAMETER),'COMMENTS',CLASS=A,         "
QUEUE "//  MSGCLASS=X,REGION=4Ø96K,MSGLEVEL=(1,1),NOTIFY=&SYSUID    "
QUEUE "//*************************************************************"
QUEUE "//*  CREATES A BASE GDG WITH 8 GENERATIONS                   *"
QUEUE "//*************************************************************"
QUEUE "//STEPØ1   EXEC PGM=IDCAMS                                    "
QUEUE "//SYSPRINT DD  SYSOUT=*                                       "
QUEUE "//SYSIN    DD  *                                              "
```

```
QUEUE "  DEF GDG (NAME("VCAT"."DB"."SP".F)LIM (8) SCR)              "
QUEUE "/*                                                          "
QUEUE "//*********************************************************"
QUEUE "//*  CREATE AN IMAGE COPY OF THE TABLESPACE                *"
QUEUE "//*   THE JOB IS EXECUTED EVEN IF THE BASE                 *"
QUEUE "//*       GDG IS IN THE CATALOG (RC 12)                    *"
QUEUE "//*********************************************************"
QUEUE "//IMAGCPY  EXEC PGM=DSNUTILB,PARM='XXXX,UTILID',TIME=SSSS,   "
QUEUE "//         COND=((Ø,GT,STEPØ1),(12,LT,STEPØ1))              "
QUEUE "//SYSCOPY  DD UNIT=CART,DISP=(,CATLG),                      "
QUEUE "//         DCB=(DSCB1,RECFM=FB,LRECL=4Ø96,BLKSIZE=28672),    "
QUEUE "//         VOL=(,,,99),                                     "
QUEUE "//         DSNAME="VCAT"."DB"."SP".F(+1),                   "
QUEUE "//         LABEL=(1,SL,EXPDT=99ØØØ)                         "
QUEUE "//SYSPRINT DD SYSOUT=*                                      "
QUEUE "//SYSUDUMP DD SYSOUT=*                                      "
QUEUE "//SYSIN    DD *                                             "
QUEUE "     COPY TABLESPACE "DB"."SP" COPYDDN SYSCOPY              "
QUEUE "          FULL YES SHRLEVEL REFERENCE                       "
QUEUE "//*********************************************************"
QUEUE "//*  STOP THE TABLESPACE                                   *"
QUEUE "//*********************************************************"
QUEUE "//STEPØ1Ø  EXEC PGM=IKJEFTØ1,REGION=4Ø96K                   "
QUEUE "//SYSTSPRT DD SYSOUT=*                                      "
QUEUE "//SYSPRINT DD SYSOUT=*                                      "
QUEUE "//SYSTSIN DD *                                              "
QUEUE "   DSN SYSTEM(XXXX)                                         "
QUEUE "   -STOP DB("DB") SPACENAM("SP")                            "
QUEUE "/*                                                          "
QUEUE "//*********************************************************"
QUEUE "//*  ALTER THE TABLESPACE                                  *"
QUEUE "//*********************************************************"
QUEUE "//STEPØ2Ø EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø,COND=(4,LT)         "
QUEUE "//STEPLIB DD DSN=SYS2.DB2.XXXX.DSNLOAD,DISP=SHR             "
QUEUE "//SYSTSPRT DD SYSOUT=*                                      "
QUEUE "//SYSPRINT DD SYSOUT=*                                      "
QUEUE "//SYSOUT   DD SYSOUT=*                                      "
QUEUE "//SYSTSIN  DD *                                             "
QUEUE "    DSN SYSTEM(XXXX)                                        "
QUEUE "   RUN PROGRAM(DSNTIAD)                                     "
QUEUE "   END                                                      "
QUEUE "//*                                                         "
QUEUE "//SYSIN    DD *                                             "
QUEUE "   SET CURRENT DEGREE = 'ANY' ;                             "
QUEUE "   ALTER TABLESPACE "DB"."SP"                               "
QUEUE "          PRIQTY "PQTY"                                     "
QUEUE "          SECQTY "SQTY";                                    "
QUEUE "   COMMIT;                                                  "
QUEUE "/*                                                          "
QUEUE "//*********************************************************"
QUEUE "//*  START THE TABLESPACE IN UT MODE                       *"
```

```
QUEUE "//*************************************************************"
QUEUE "//STEP3Ø   EXEC PGM=IKJEFTØ1,REGION=4Ø96K,COND=(4,LT)        "
QUEUE "//SYSTSPRT DD SYSOUT=*                                       "
QUEUE "//SYSPRINT DD SYSOUT=*                                       "
QUEUE "//SYSTSIN DD *                                               "
QUEUE "   DSN SYSTEM(XXXX)                                          "
QUEUE "   -STA DB("DB") SPACENAM("SP") ACCESS(UT)                   "
QUEUE "/*                                                           "
QUEUE "//*************************************************************"
QUEUE "//*  RECOVER THE TABLESPACE                                  *"
QUEUE "//*************************************************************"
QUEUE "//RECVTS  EXEC PROC=DSNUPROC,TIME=SSSS,COND=(4,LT),          "
QUEUE "//        SYSTEM=XXXX,UID='UTILID',UTPROC=''                 "
QUEUE "//DSNUPROC.SYSUT1 DD DISP=(MOD,DELETE,DELETE),               "
QUEUE "//        DSNAME=XXXXWRK."DB"."SP".SYSUT1,                   "
QUEUE "//        SPACE=(CYL,(15Ø,5Ø),RLSE)                          "
QUEUE "//DSNUPROC.SYSIN DD *                                        "
QUEUE "  RECOVER TABLESPACE "DB"."SP"                               "
QUEUE "/*                                                           "
QUEUE "//*************************************************************"
QUEUE "//*  START THE TABLESPACE IN RW MODE                         *"
QUEUE "//*************************************************************"
QUEUE "//STARTA   EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø,COND=(4,LT)         "
QUEUE "//SYSTSPRT DD SYSOUT=*                                       "
QUEUE "//SYSPRINT DD SYSOUT=*                                       "
QUEUE "//SYSTSIN  DD *                                              "
QUEUE "  DSN SYSTEM(XXXX)                                           "
QUEUE "  -STA DATABASE("DB") SPACENAM("SP") ACCESS(RW)              "
QUEUE "/*                                                           "
QUEUE "//*                                                          "
"EXECIO 89 DISKW OUT (FINIS"
"SUBMIT 'XXXTEST.USERID.ALTER.JCL'"
"FREE DD(OUT)"
DELSTACK
RETURN

/* ROUTINE TO ALTER A PART OF A TABLESPACE           */
PARTTS:
/* QUEUE THE JCL AND SUBMIT                          */
/*  THE JCL IS WRITTEN INTO XXXTEST.USERID.ALTER.JCL  */
/*    WHICH IS ALLOCATED WITH LRECL 8Ø                */

"ALLOC DS(USERID.ALTER.JCL)  FI(OUT) SHR"
NEWSTACK
QUEUE "//XXXXXJOB  JOB (ACCT PARAMETER),'COMMENTS',CLASS=A,         "
QUEUE "//  MSGCLASS=X,REGION=4Ø96K,MSGLEVEL=(1,1),NOTIFY=&SYSUID    "
QUEUE "//*************************************************************"
QUEUE "//*  CREATE A BASE GDG WITH 8 GENERATIONS                   *"
QUEUE "//*************************************************************"
QUEUE "//STEPØ1   EXEC PGM=IDCAMS                                   "
QUEUE "//SYSPRINT DD  SYSOUT=*                                      "
```

```
QUEUE "//SYSIN    DD  *                                                 "
QUEUE "  DEF GDG (NAME("VCAT"."DB"."SP".PART"PART".F)LIM (8) SCR)   "
QUEUE "/*                                                             "
QUEUE "//*************************************************************"
QUEUE "//*  CREATE AN IMAGE COPY OF THE PARTITION                   *"
QUEUE "//*   THE JOB IS EXECUTED EVEN IF THE BASE                   *"
QUEUE "//*       GDG IS IN THE CATALOG (RC 12)                      *"
QUEUE "//*************************************************************"
QUEUE "//IMAGCPY  EXEC PGM=DSNUTILB,PARM='XXXX,UTILID',TIME=SSSS,   "
QUEUE "//         COND=((Ø,GT,STEPØ1),(12,LT,STEPØ1))               "
QUEUE "//SYSCOPY  DD UNIT=CART,DISP=(,CATLG),                       "
QUEUE "//         DCB=(DSCB1,RECFM=FB,LRECL=4Ø96,BLKSIZE=28672),    "
QUEUE "//         VOL=(,,,99),                                      "
QUEUE "//         DSNAME="VCAT"."DB"."SP".PART"PART".F(+1),         "
QUEUE "//         LABEL=(1,SL,EXPDT=99ØØØ)                          "
QUEUE "//SYSPRINT DD SYSOUT=*                                       "
QUEUE "//SYSUDUMP DD SYSOUT=*                                       "
QUEUE "//SYSIN    DD *                                              "
QUEUE "     COPY TABLESPACE "DB"."SP" DSNUM "PART" COPYDDN SYSCOPY  "
QUEUE "          FULL YES SHRLEVEL REFERENCE                        "
QUEUE "//*************************************************************"
QUEUE "//*  STOP THE REQ'ED PARTITION OF THE TABLESPACE            *"
QUEUE "//*************************************************************"
QUEUE "//STEPØ1Ø  EXEC PGM=IKJEFTØ1,REGION=4Ø96K                    "
QUEUE "//SYSTSPRT DD SYSOUT=*                                       "
QUEUE "//SYSPRINT DD SYSOUT=*                                       "
QUEUE "//SYSTSIN DD *                                               "
QUEUE "   DSN SYSTEM(XXXX)                                          "
QUEUE "   -STOP DB("DB") SPACENAM("SP") PART("PART")                "
QUEUE "/*                                                           "
QUEUE "//*************************************************************"
QUEUE "//*  ALTER THE PARTITION                                     *"
QUEUE "//*************************************************************"
QUEUE "//STEPØ2Ø EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø,COND=(4,LT)          "
QUEUE "//STEPLIB DD DSN=SYS2.DB2.XXXX.DSNLOAD,DISP=SHR              "
QUEUE "//SYSTSPRT DD SYSOUT=*                                       "
QUEUE "//SYSPRINT DD SYSOUT=*                                       "
QUEUE "//SYSOUT   DD SYSOUT=*                                       "
QUEUE "//SYSTSIN  DD *                                              "
QUEUE "    DSN SYSTEM(XXXX)                                         "
QUEUE "   RUN PROGRAM(DSNTIAD)                                      "
QUEUE "   END                                                       "
QUEUE "//*                                                          "
QUEUE "//SYSIN    DD *                                              "
QUEUE "   SET CURRENT DEGREE = 'ANY' ;                              "
QUEUE "   ALTER TABLESPACE "DB"."SP" PART "PART"                    "
QUEUE "          PRIQTY "PQTY"                                      "
QUEUE "          SECQTY "SQTY";                                     "
QUEUE "   COMMIT;                                                   "
QUEUE "/*                                                           "
QUEUE "//*************************************************************"
```

```
QUEUE "//*  START THE PARTITION IN UT MODE                          *"
QUEUE "//*************************************************************"
QUEUE "//STEP3Ø   EXEC PGM=IKJEFTØ1,REGION=4Ø96K,COND=(4,LT)        "
QUEUE "//SYSTSPRT DD SYSOUT=*                                       "
QUEUE "//SYSPRINT DD SYSOUT=*                                       "
QUEUE "//SYSTSIN DD *                                               "
QUEUE "   DSN SYSTEM(XXXX)                                          "
QUEUE "   -STA DB("DB") SPACENAM("SP") PART("PART") ACCESS(UT)      "
QUEUE "/*                                                           "
QUEUE "//*************************************************************"
QUEUE "//*  RECOVER THE PARTITION OF THE TABLESPACE                *"
QUEUE "//*************************************************************"
QUEUE "//RECVTS  EXEC PROC=DSNUPROC,TIME=SSSS,COND=(4,LT),          "
QUEUE "//        SYSTEM=XXXX,UID='UTILID',UTPROC=''                 "
QUEUE "//DSNUPROC.SYSUT1 DD DISP=(MOD,DELETE,DELETE),               "
QUEUE "//        DSNAME=XXXXWRK."DB"."SP".PART"PART".SYSUT1,        "
QUEUE "//        SPACE=(CYL,(15Ø,5Ø),RLSE)                          "
QUEUE "//DSNUPROC.SYSIN DD *                                        "
QUEUE "  RECOVER TABLESPACE "DB"."SP" DSNUM "PART"                  "
QUEUE "//*                                                          "
QUEUE "//*************************************************************"
QUEUE "//*  START THE PARTITION IN RW MODE                         *"
QUEUE "//*************************************************************"
QUEUE "//STARTA   EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø,COND=(4,LT)         "
QUEUE "//SYSTSPRT DD SYSOUT=*                                       "
QUEUE "//SYSPRINT DD SYSOUT=*                                       "
QUEUE "//SYSTSIN  DD *                                              "
QUEUE "  DSN SYSTEM(XXXX)                                           "
QUEUE "  -STA DATABASE("DB") SPACENAM("SP") PART("PART") ACCESS(RW) "
QUEUE "/*                                                           "
QUEUE "//*                                                          "
"EXECIO 89 DISKW OUT (FINIS"
"SUBMIT 'XXXTEST.USERID.ALTER.JCL'"
"FREE DD(OUT)"
DELSTACK
RETURN
ALTERIX
/* REXX TO ALTER THE INDEXSPACE                    */
/* THE OUTPUT OF THE EXTENTS EXEC IS IN            */
/*   XXXTEST.USERID.JCLOUT.DATA                    */
   "ALLOC DS(USERID.JCLOUT.DATA)  FI(INP) SHR REU"
/* INITIALIZE THE ARRAY                            */
   DSNAME. = ''
/* READ THE FILE INTO AN ARRAY                     */
   "EXECIO * DISKR INP (STEM DSNAME."
/* SEGREGATE THE INPUT TO DIFFERENT VARIABLES      */
/* THE INPUT IS IN THE FORMAT AS GIVEN BELOW :     */
/* ..QUALIFIER ..DATABASE.. ..SPACENAM.. ØØZZZ..   */
/*                          PQTY... SECQTY         */
/*   WHERE QUALIFIER – CREATOR OF THE INDEX        */
/*         DB        – NAME OF THE DATABASE        */
```

```
/*         SP        - SPACENAME                     */
/*         PART      - NUM OF THE PARTITION          */
/*         PQTY      - INCREASED PQTY WHICH IS       */
/*                       A MULTIPLE OF 4KB PAGES     */
/*         SECQTY    - INCREASED SECQTY WHICH IS     */
/*                       A MULTIPLE OF 4KB PAGES     */

   QUALIFIER = SUBWORD(DSNAME.1,1,1)
   DB        = SUBSTR(SUBWORD(DSNAME.1,2,1),3,8)
   SP        = SUBSTR(SUBWORD(DSNAME.1,3,1),3,8)
   PART      = SUBSTR(SUBWORD(DSNAME.1,4,1),3,3)
   PQTY      = SUBSTR(SUBWORD(DSNAME.1,5,1),1,10)
   SQTY      = SUBSTR(SUBWORD(DSNAME.1,6,1),1,10)

/* TO CHECK IF THE SPACENAME IS IN THE CATALOG       */
   IF DB = ' '
      THEN DO
            SAY 'SPACENAME IS NOT FOUND IN THE CATALOG'
            EXIT
           END
   ELSE
      NOP

/* TO DISPLAY THE PARAMETERS TO BE USED IN THE JCL    */
   SAY 'DBNAME    ' DB
   SAY 'SPACENM   ' SP
   SAY 'PART      ' PART
   SAY 'QUALIFIER ' QUALIFIER
   SAY 'PQTY      ' PQTY
   SAY 'SQTY      ' SQTY

/* TO CHECK FOR THE APPROPRIATE ROUTINE TO BE CALLED  */
   IF PART = 0
      THEN
        DO
          SAY 'TO ALTER THE WHOLE INDEXSPACE'
          CALL NPARTIX
        END
   ELSE
        DO
          SAY 'TO ALTER THE PARTITIONED INDEXSPACE'
          CALL PARTIX
        END
   EXIT
/* ROUTINE TO ALTER THE WHOLE INDEXSPACE              */
NPARTIX:
/* QUEUE THE JCL AND SUBMIT                           */
/*  THE JCL IS WRITTEN INTO XXXTEST.USERID.ALTER.JCL  */
/*    WHICH IS ALLOCATED WITH LRECL 80                */

"ALLOC DS(USERID.ALTER.JCL)  FI(OUT) SHR"
```

```
NEWSTACK
QUEUE "//XXXXXJOB  JOB (ACCT PARAMETER),'COMMENTS',CLASS=A,         "
QUEUE "//  MSGCLASS=X,REGION=4Ø96K,MSGLEVEL=(1,1),NOTIFY=&SYSUID    "
QUEUE "//*************************************************************"
QUEUE "//* STOP THE INDEXSPACE                                       "
QUEUE "//*************************************************************"
QUEUE "//STEPØ1Ø  EXEC PGM=IKJEFTØ1,REGION=4Ø96K                     "
QUEUE "//SYSTSPRT DD SYSOUT=*                                       "
QUEUE "//SYSPRINT DD SYSOUT=*                                       "
QUEUE "//SYSTSIN DD *                                               "
QUEUE "   DSN SYSTEM(XXXX)                                          "
QUEUE "   -STOP DB("DB") SPACENAM("SP")                             "
QUEUE "/*                                                           "
QUEUE "//*************************************************************"
QUEUE "//* ALTER THE INDEXSPACE                                      "
QUEUE "//*************************************************************"
QUEUE "//STEPØ2Ø EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø,COND=(4,LT)          "
QUEUE "//STEPLIB DD DSN=SYS2.DB2.XXXX.DSNLOAD,DISP=SHR              "
QUEUE "//SYSTSPRT DD SYSOUT=*                                       "
QUEUE "//SYSPRINT DD SYSOUT=*                                       "
QUEUE "//SYSOUT   DD SYSOUT=*                                       "
QUEUE "//SYSTSIN  DD *                                              "
QUEUE "    DSN SYSTEM(XXXX)                                         "
QUEUE "   RUN PROGRAM(DSNTIAD)                                      "
QUEUE "   END                                                       "
QUEUE "//*                                                          "
QUEUE "//SYSIN    DD *                                              "
QUEUE "   SET CURRENT DEGREE = 'ANY' ;                              "
QUEUE "   ALTER INDEX "QUALIFIER"."SP"                              "
QUEUE "           PRIQTY "PQTY"                                     "
QUEUE "           SECQTY "SQTY";                                    "
QUEUE "   COMMIT;                                                   "
QUEUE "/*                                                           "
QUEUE "//*************************************************************"
QUEUE "//* START THE INDEXSPACE IN UT MODE                          "
QUEUE "//*************************************************************"
QUEUE "//STEP3Ø   EXEC PGM=IKJEFTØ1,COND=(4,LT)                     "
QUEUE "//SYSTSPRT DD SYSOUT=*                                       "
QUEUE "//SYSPRINT DD SYSOUT=*                                       "
QUEUE "//SYSTSIN DD *                                               "
QUEUE "   DSN SYSTEM(XXXX)                                          "
QUEUE "   -STA DB("DB") SPACENAM("SP") ACCESS(UT)                   "
QUEUE "/*                                                           "
QUEUE "//*************************************************************"
QUEUE "//* RECOVER THE INDEX                                        "
QUEUE "//*************************************************************"
QUEUE "//RECVIX  EXEC PROC=DSNUPROC,TIME=SSSS,COND=(4,LT),          "
QUEUE "//        SYSTEM=XXXX,UID='UTILID',UTPROC=''                 "
QUEUE "//DSNUPROC.SYSUT1 DD DISP=(MOD,DELETE,DELETE),               "
QUEUE "//        DSNAME=XXXXWRK."DB"."SP".SYSUT1,                   "
QUEUE "//        SPACE=(CYL,(15Ø,5Ø),RLSE)                          "
```

```
QUEUE "//DSNUPROC.SYSIN DD *                                              "
QUEUE "  RECOVER INDEX ("QUALIFIER"."SP")                                 "
QUEUE "//*                                                                "
QUEUE "//*****************************************************************"
QUEUE "//* START THE INDEXSPACE IN RW MODE                               "
QUEUE "//*****************************************************************"
QUEUE "//STARTA   EXEC PGM=IKJEFT01,COND=(4,LT)                           "
QUEUE "//SYSTSPRT DD SYSOUT=*                                             "
QUEUE "//SYSPRINT DD SYSOUT=*                                             "
QUEUE "//SYSTSIN  DD *                                                    "
QUEUE "  DSN SYSTEM(XXXX)                                                 "
QUEUE "  -STA DATABASE("DB") SPACENAM("SP") ACCESS(RW)                    "
QUEUE "/*                                                                 "
QUEUE "//*                                                                "
"EXECIO 63 DISKW OUT (FINIS"
"SUBMIT 'XXXTEST.USERID.ALTER.JCL'"
"FREE DD(OUT)"
DELSTACK
RETURN

/* ROUTINE TO ALTER THE PART OF AN INDEXSPACE         */
PARTIX:
/* QUEUE THE JCL AND SUBMIT                          */
/*  THE JCL IS WRITTEN INTO XXXTEST.USERID.ALTER.JCL  */
/*    WHICH IS ALLOCATED WITH LRECL 80                */

"ALLOC DS(USERID.ALTER.JCL)  FI(OUT) SHR"
NEWSTACK
QUEUE "//XXXXXJOB  JOB (ACCT PARAMETER),'COMMENTS',CLASS=A,               "
QUEUE "//  MSGCLASS=X,REGION=4096K,MSGLEVEL=(1,1),NOTIFY=&SYSUID          "
QUEUE "//*****************************************************************"
QUEUE "//* STOP THE INDEXSPACE                                           "
QUEUE "//*****************************************************************"
QUEUE "//STEP010  EXEC PGM=IKJEFT01,REGION=4096K                          "
QUEUE "//SYSTSPRT DD SYSOUT=*                                             "
QUEUE "//SYSPRINT DD SYSOUT=*                                             "
QUEUE "//SYSTSIN DD *                                                     "
QUEUE "   DSN SYSTEM(XXXX)                                                "
QUEUE "   -STOP DB("DB") SPACENAM("SP") PART("PART")                      "
QUEUE "/*                                                                 "
QUEUE "//*****************************************************************"
QUEUE "//* ALTER THE INDEXSPACE                                          "
QUEUE "//*****************************************************************"
QUEUE "//STEP020 EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)                "
QUEUE "//STEPLIB DD DSN=SYS2.DB2.XXXX.DSNLOAD,DISP=SHR                    "
QUEUE "//SYSTSPRT DD SYSOUT=*                                             "
QUEUE "//SYSPRINT DD SYSOUT=*                                             "
QUEUE "//SYSOUT   DD SYSOUT=*                                             "
QUEUE "//SYSTSIN  DD *                                                    "
QUEUE "    DSN SYSTEM(XXXX)                                               "
```

```
QUEUE "   RUN PROGRAM(DSNTIAD)                                          "
QUEUE "   END                                                           "
QUEUE "//*                                                              "
QUEUE "//SYSIN    DD *                                                  "
QUEUE "   SET CURRENT DEGREE = 'ANY' ;                                  "
QUEUE "   ALTER INDEX "QUALIFIER"."SP" PART "PART"                      "
QUEUE "          PRIQTY "PQTY"                                          "
QUEUE "          SECQTY "SQTY";                                         "
QUEUE "/*                                                               "
QUEUE "//*************************************************************"
QUEUE "//* START THE INDEXSPACE IN UT MODE                             "
QUEUE "//*************************************************************"
QUEUE "//STEP3Ø   EXEC PGM=IKJEFTØ1,REGION=4Ø96K,COND=(4,LT)            "
QUEUE "//SYSTSPRT DD SYSOUT=*                                           "
QUEUE "//SYSPRINT DD SYSOUT=*                                           "
QUEUE "//SYSTSIN DD *                                                   "
QUEUE "   DSN SYSTEM(XXXX)                                              "
QUEUE "   -STA DB("DB") SPACENAM("SP") PART("PART") ACCESS(UT)          "
QUEUE "/*                                                               "
QUEUE "//*************************************************************"
QUEUE "//* RECOVER THE INDEX                                           "
QUEUE "//*************************************************************"
QUEUE "//RECVIX  EXEC PROC=DSNUPROC,TIME=SSSS,COND=(4,LT),              "
QUEUE "//        SYSTEM=XXXX,UID='UTILID',UTPROC=''                     "
QUEUE "//DSNUPROC.SYSUT1 DD DISP=(MOD,DELETE,DELETE),                   "
QUEUE "//        DSNAME=XXXXWRK."DB"."SP".SYSUT1,                       "
QUEUE "//        SPACE=(CYL,(15Ø,5Ø),RLSE)                              "
QUEUE "//DSNUPROC.SYSIN DD *                                            "
QUEUE "  RECOVER INDEX ("QUALIFIER"."SP" PART "PART")                   "
QUEUE "//*                                                              "
QUEUE "//*************************************************************"
QUEUE "//* START THE INDEXSPACE IN RW MODE                             "
QUEUE "//*************************************************************"
QUEUE "//STARTA   EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø,COND=(4,LT)             "
QUEUE "//SYSTSPRT DD SYSOUT=*                                           "
QUEUE "//SYSPRINT DD SYSOUT=*                                           "
QUEUE "//SYSTSIN  DD *                                                  "
QUEUE "  DSN SYSTEM(XXXX)                                               "
QUEUE "  -STA DATABASE("DB") SPACENAM("SP") PART ("PART") ACCESS(RW)"
QUEUE "/*                                                               "
QUEUE "//*                                                              "

"EXECIO 63 DISKW OUT (FINIS"
"SUBMIT 'XXXTEST.USERID.ALTER.JCL'"
"FREE DD(OUT)"
DELSTACK
RETURN
```

Kiran Haryadi and K R Swaminaathan
DBA
Wipro Infotech (India)

Brio Techology has announced Brio Enterprise: IBM DB2 Edition, which has been optimized for DB2 across all platforms. Featuring native support for DB2 UDB, DB2 UDB for AS/400, and DB2 OLAP Server, a 'snap in' meta data connection for Visual Warehouse Information Catalog, customized look-and-feel, packaging, support, and other capabilities, it is available in Web and Microsoft Windows client implementations and for both AIX and Windows NT server environments.

For further information contact:
Brio Technology, 3950 Fabian Way, Suite 200, Palo Alto, CA 94303, USA.
Tel: (650) 856 8000.
Brio Technology, Europa House Church Street, Isleworth, TW7 6EB, UK.
Tel: (0181) 569 8999.
URL: http://www.brio.com.

* * *

StarQuest has announced Version 3.0 of its StarSQL product for accessing DB2 data from Windows applications. It provides access to data in IBM databases on mainframe and mid-range systems from ODBC-enabled PC or Unix CLI applications using either TCP/IP or SNA.

New features of the driver, which runs entirely on the desktop, include two-phase commit for distributed on-line transaction processing support, plus support for ODBC Version 3 and Unix, as well as enhanced static SQL support and a new database connection wizard. The ODBC connectivity wizard detects the appropriate configuration parameters for the DB2 database being accessed. Other features include support for Remote Data Objects (RDO) and Active Data Objects (ADO), remote password management, and new accounting information providing data to the DB2 trace facility for use in data centre chargebacks. There's also a new installer, support for native security mechanisms of DB2, and support for Java via SunSoft JDK ODBC-to-JDBC bridge.

For further information contact:
StarQuest Software, 1288 Ninth Street, Berkeley, CA 94710, USA.
Tel: (510) 528 2900.
URL: http://www.starquest.com.

* * *

DB2 users can benefit from Release 1.5 of Sterling Software's VISION:Phaseshift date-masking tool for MVS and OS/390, insulating legacy applications from Y2K date issues without making any changes to the code. The new release reduces installation time by supporting import of mass data definitions for DB2, IMS, and flat files, and it has better DB2 support and provides more flexible test facilities.

For further information contact:
Sterling Software, 1800 Alexander Bell Drive, Reston, VA 22091, USA.
Tel: (703) 264 8000.
Sterling Software, 1 Longwalk Road, Stockley Park, Uxbridge, Middlesex, UB11 1DB, UK.
Tel: (0181) 867 8000.
URL: http://www.sterling.com.