74

DB2

December 1998

In this issue

© Xephon plc 1998

update

DB2 Update

Published by
Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38030
From USA: 01144 1635 38030
E-mail: xephon@compuserve.com

North American office
Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75077-2150
USA
Telephone: 940 455 7050

Contributions
Articles published in *DB2 Update* are paid for at the rate of £170 ($250) per 1000 words and £90 ($140) per 100 lines of code for original material. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*.

DB2 Update on-line
Code from *DB2 Update* can be downloaded from our Web site at http://www.xephon.com; you will need the user-id shown on your address label.

Editor
Robert Burgess

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

Subscriptions and back-issues
A year's subscription to *DB2 Update*, comprising twelve monthly issues, costs £245.00 in the UK; $365.00 in the USA and Canada; £251.00 in Europe; £257.00 in Australasia and Japan; and £255.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1994 issue, are available separately to subscribers for £21.00 ($31.00) each including postage.

© Xephon plc 1998. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs $36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

Printed in England.

Storage impact of Type 2 indexes

Type 2 indexes provide numerous benefits for a DB2 subsystem. The primary benefit is the elimination of index locking. However, many newer DB2 features such as row-level locking and uncommitted reads require Type 2 indexes. Furthermore, with DB2 Version 6, available in 1999, Type 1 indexes are eliminated altogether, so you will be forced to move to Type 2 indexes.

However, before rushing off to implement them, you would be wise to investigate the storage impact of migrating from Type 1 to Type 2 indexes.

Just what impact will Type 2 indexes have on storage requirements? The answer, not surprisingly, is 'it depends!'. There are quite a few differences between Type 1 and Type 2 indexes that impact on storage.

The first difference is in the amount of usable space on an index page. A Type 2 leaf page has 4,038 bytes of usable space; a Type 2 non-leaf page has 4,046 bytes. Type 1 leaf and non-leaf pages have 4,050 usable bytes per page. So, Type 2 indexes have less usable space per page.

Additionally, Type 2 indexes require an additional one-byte RID prefix in addition to the four-byte RID found in both Type 1 and Type 2 indexes. The new one-byte RID prefix found in a Type 2 index contains three flags – pseudo-deleted, possibly uncommitted, and RID hole follows.

Because Type 2 indexes have a different internal structure, two pieces of header information needed on Type 1 indexes are no longer required – the subpage header and the non-unique key header. Since Type 2 indexes do not use subpages, the 17-byte logical subpage header required of a Type 1 index is not in Type 2 indexes.

Non-unique Type 1 indexes have a six-byte header and will repeat an entry (header and key) if a key has more than 255 RIDs. Type 2 indexes have a two-byte header and can have more than 255 RIDs in each entry. The entry is only repeated if there is not enough room in a leaf page to hold all the RIDs; the same is true for a Type 1 index.

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

Type 2 indexes also have a two-byte MAPID for each key at the end of the page, so the total saving per key is two bytes (six bytes for the Type 1 header, minus two bytes for the Type 2 header and two bytes for the MAPID).

Type 2 indexes store truncated keys instead of the complete key. Only the portion of the key required to make it uniquely identifiable is stored on non-leaf pages. However, if there are many duplicate keys so that the same key is on more than one leaf page, a Type 2 index will have RIDs stored in the non-leaf pages, causing more space to be used rather than less. This is because Type 2 indexes keep the RIDs in sequence.

Finally, Type 2 indexes are required for large tablespaces. In this case the RID is five bytes (plus the one-byte RID prefix, which is still required).

As you can see, there is no clear-cut answer as to whether a Type 1 or Type 2 index will utilize more storage. In general, though, you should favour creating Type 2 indexes instead of Type 1 because of the advantages they offer:

- There is no index locking with Type 2 indexes. Index locking is one of the predominant causes of contention in pre-Version 4 DB2 applications.
- Type 2 indexes are the only type supported for ASCII encoded tables.
- As touched upon earlier, many newer DB2 features cannot be used unless Type 2 indexes are used – these features include large object support, row-level locking, data sharing, full partition independence, uncommitted reads, UNIQUE WHERE NOT NULL, and CPU and Sysplex parallelism.
- Furthermore, IBM is promoting Type 2 indexes as the standard and will remove support for Type 1 indexes in DB2 Version 6. However, as of DB2 Version 5, both Type 1 and Type 2 indexes are still supported.

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

RULES-OF-THUMB FOR INDEX STORAGE

Taking all of the points above into consideration, here are some general rules-of-thumb on index storage requirements that you can apply when developing DB2 databases:

- A Type 1 index with a subpage of 16 usually wastes a lot of space. A Type 2 index will almost always use less space than a Type 1 with 16 subpages (but so will a Type 1 index with a subpage of 1).
- A Type 1 index with a subpage of 1 will usually require slightly less space than a Type 2 index for both unique and non-unique keys. For the average user, the space difference is relatively small and should not be a factor.
- Beware of Type 2 space usage if numerous row deletes occur. Type 1 indexes clean up after a delete, while DB2 pseudo-deletes index RID entries. A pseudo-delete is when DB2 marks the index entry for deletion, but does not physically delete it. When high levels of activity occur, you could encounter numerous pages of nothing but pseudo-deleted RIDs. DB2 should periodically clean up the pseudo-deleted entries, but in some cases users have reported seeing them staying around for weeks at a time wasting space. A reorganization or rebuild will clean up the pseudo-deleted RIDs and free the wasted space.
- Beware of space usage when numerous inserts occur. Type 1 index entries move around in the page and finally, when a split occurs, one half of the index entries are moved to another page, usually causing the one half page to be wasted. This is known as the ‘half-full’ problem. Type 2 index pages will also split, but provision has been made at the end of a dataset to avoid the ‘half-full’ problem.

  In addition, Type 2 indexes with non-unique keys will chain RIDs within a page. Each chain entry requires a chain pointer and the normal RID. The additional overhead is two bytes plus the Type 2 RID. All these problems can be solved by reorganizing the index.
- The user should monitor the disk space usage of both Type 1 and

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

Type 2 indexes and reorganize the indexes when they grow too large or when performance problems arise.

Be sure to factor all of these issues into your index storage requirement exercises. The actual index sizing formulas are contained in the IBM DB2 manuals and you should use these calculations (or an automated space calculator) to arrive at actual index space requirements. Good luck planning the space requirements for your Type 2 index conversion.

Craig S Mullins
VP Operations
Platinum Technology (USA)
© Platinum Technology 1998

PLAN and PACKAGE management – part 2

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

```
/*————————————————————————————————————————*/
/*–      RELEASE   "Deallocate/Commit"     ——*/
/*————————————————————————————————————————*/
    if rel_pk = D then
       rele  = DEALLOCATE
    else
       rele  = COMMIT
/*————————————————————————————————————————*/
/*–      EXPLAIN   "YES/NO"                ——*/
/*————————————————————————————————————————*/
    if exp_pk = Y then
       expl  = YES
    else
       expl  = NO
/*————————————————————————————————————————*/
/*–    Test Package name already created   ——*/
/*————————————————————————————————————————*/
    wrkok = outdspk'('pak_pk')'
    dsn = sysdsn''''wrkok''''
    if dsn = OK then do
       #pk = #pk + 1
       wrkok =  pk || right(#pk,6,'0')
       end
    else
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
          wrkok = pak_pk
   /*———————————————————————————————————————————*/
   /*—      Build Bind job                      ———*/
   /*———————————————————————————————————————————*/
       if swsyspr = on then do
sb.#x='//SYSTSIN   DD  DSN='outdspk'('wrkok'),DISP=SHR                '
          #x = #x + 1
          swsyspr = off
          end
       else do
sb.#x='//          DD  DSN='outdspk'('wrkok'),DISP=SHR                '
          #x = #x + 1
          end
       if #x1 > 8Ø then do
sb.#x='//* ————————————————————————— *      '
          #x = #x + 1
          call hdrbnd
          #x1 = Ø
          end
       #x1 = #x1 + 1
       jobw = fipk
       "alloc da('"outdspk"("wrkok")') f("jobw") shr reuse"
sk.1='DSN SYSTEM('subsys')                                            '
       lcol = length('BIND PACKAGE('col_pk')')
       wrk = 5Ø - lcol
sk.2='BIND PACKAGE('col_pk')'copies(' ',wrk)'-'
       lpds = length('LIBRARY('''pds_pk''')')
       wrk = 5Ø - lpds
sk.3='LIBRARY('''pds_pk''')'copies(' ',wrk)'-'
       lpak = length('MEMBER('pak_pk')')
       wrk = 5Ø - lpak
sk.4='MEMBER('pak_pk')'copies(' ',wrk)'-'
       lown = length('OWNER('own_pk')')
       wrk = 5Ø - lown
sk.5='OWNER('own_pk')'copies(' ',wrk)'-'
       lqua = length('QUALIFIER('qua_pk')')
       wrk = 5Ø - lqua
sk.6='QUALIFIER('QUA_pk')'copies(' ',wrk)'-'
sk.7='ACT(REPLACE)                                          -'
       lvalid = length('VALIDATE('valid')')
       wrk = 5Ø - lvalid
sk.8='VALIDATE('valid')'copies(' ',wrk)'-'
       lisol = length('ISOLATION('isol')')
       wrk = 5Ø - lisol
sk.9='ISOLATION('isol')'copies(' ',wrk)'-'
       lrele = length('RELEASE('rele')')
       wrk = 5Ø - lrele
sk.1Ø='RELEASE('rele')'copies(' ',wrk)'-'
sk.11='EXPLAIN('expl')'
       sk.Ø =11
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
      call Writeout
      ctrpk = ctrpk + 1
      end
   /*-------------------------------------------------------*/
   /*-      Write Bind job                                 --*/
   /*-------------------------------------------------------*/
sb.#x='//* ------------------------- *      '
    sb.0 = #x
    jobw = fiupd
    "alloc da('"outdsupd"(bindpk)') f("jobw") mod reuse"
    call Wrindx
   /*-------------------------------------------------------*/
   /*-    IPOUPDTE job for .PACKAGE library   --*/
   /*-------------------------------------------------------*/
    jobw = fiupd
    "alloc da('"outdsupd"(ipouppk)') f("jobw") mod reuse"
    call hdripo
sk.19='//IPOUPPK  EXEC PGM=IPOUPDTE,                                '
sk.20='//*       PARM='UPDATE'              /* Run with UPDATE      '
sk.21='//*       PARM='CHECK'               /* Run without UPDATE   '
sk.22='//STEPLIB  DD  DSN=IPO1.LINKLIB,DISP=SHR                     '
sk.23='//SYSPRINT DD  SYSOUT=*                                      '
sk.24='//@INST    DD  DSN='outdspk',DISP=SHR                        '
sk.25='//SYSIN    DD  DATA,DLM=@@                                   '
sk.26='</*                                                          '
sk.27=''subsys'<+                                                   '
sk.28=' ????<                           /* New DB2 subsystem       '
sk.29=''col_pk'<+                                                   '
sk.30=' ??????????????????<             /* New Collection          '
sk.31=''own_pk'<+                                                   '
sk.32=' ????????<                       /* New Owner               '
sk.33=''qua_pk'<+                                                   '
sk.34=' ????????<                       /* New Qualifier           '
sk.35=''valid'<+                                                    '
sk.36=' ????<                           /* Validate RUN/BIND       '
sk.37=''isol'<+                                                     '
sk.38=' ??<                             /* Isolation RR/CS         '
sk.39='REPLACE<+                                                    '
sk.40=' ???????<                        /* Action Replace/Add/Retain'
sk.41=''rele'<+                                                     '
sk.42=' ??????????<                     /* Release Commit/Deallocate'
sk.43=''expl'<+                                                     '
sk.44=' ???<                            /* Explain YES/NO          '
sk.45=''pds_pk'<+                                                   '
sk.46=' ????????????????<               /* New PDS for dbrms       '
sk.47='</*                                                          '
    sk.0=47
    Call Writeout
    return
   /*-------------------------------------------------------*/
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
   /*-   Write bind plan members with Package   -*/
   /*-------------------------------------------------*/
 Wrbppack:
    say '>>>>>>>>  Building library 'outdsppk
    jobw = fippk
    "alloc da('"outdsppk"($$$coibm)') f("jobw") shr reuse"
sk.1='     ////////////////////////////////////////////////////// '
sk.2='     ****** Do not erase this member !!!!     Thanks.  ****** '
sk.3='     ////////////////////////////////////////////////////// '
    sk.Ø = 3
    call Writeout
    yesppk = on
    #x  = 1
    #x1 = Ø
    Call Hdrbnd
    DO #c = 1 to sysprint.Ø
       seq_ppk = word(sysprint.#c,1)
       pla_ppk = word(sysprint.#c,2)
       pak_ppk = word(sysprint.#c,3)
       col_ppk = word(sysprint.#c,4)
       cre_ppk = word(sysprint.#c,5)
       qua_ppk = word(sysprint.#c,6)
       val_ppk = word(sysprint.#c,7)
       acq_ppk = word(sysprint.#c,8)
       iso_ppk = word(sysprint.#c,9)
       rel_ppk = word(sysprint.#c,1Ø)
       exp_ppk = word(sysprint.#c,11)
   /*-------------------------------------------------*/
   /*-      VALIDATE  "Bind/Run"               ----*/
   /*-------------------------------------------------*/
       if val_ppk = B then
          valid_ppk = BIND
       else
          valid_ppk = RUN
   /*-------------------------------------------------*/
   /*-     ACQIRE    "Use/Allocate"           ----*/
   /*-------------------------------------------------*/
       if acq_ppk = A then
          acqu_ppk = ALLOCATE
       else
          acqu_ppk = USE
   /*-------------------------------------------------*/
   /*-  ISOLATION "Rep.Read/Cursor Stability    -*/
   /*-------------------------------------------------*/
       if iso_ppk = R then
          isol_ppk  = RR
       else
          isol_ppk  = CS
   /*-------------------------------------------------*/
   /*-    RELEASE   "Deallocate/Commit"       ----*/
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
   /*-------------------------------------------------------*/
       if rel_ppk = D then
          rele_ppk  = DEALLOCATE
       else
          rele_ppk  = COMMIT
   /*-------------------------------------------------------*/
   /*-    EXPLAIN   "YES/NO"                  ---*/
   /*-------------------------------------------------------*/
       if exp_ppk = Y then
          expl_ppk  = YES
       else
          expl_ppk  = NO
   /*-------------------------------------------------------*/
   /*-    Build Bind job                      ---*/
   /*-------------------------------------------------------*/
       if swsyspr = on then do
sb.#x='//SYSTSIN   DD  DSN='outdsppk'('pla_ppk'),DISP=SHR            '
          #x = #x + 1
          swsyspr = off
          end
       else do
sb.#x='//          DD  DSN='outdsppk'('pla_ppk'),DISP=SHR            '
          #x = #x + 1
          end
       if #x1 > 8Ø then do
sb.#x='//* ------------------------- *      '
          #x = #x + 1
          call hdrbnd
          #x1 = Ø
          end
       #x1 = #x1 + 1
       jobw = fippk
       "alloc da('"outdsppk"("pla_ppk")') f("jobw") shr reuse"
sk.1='DSN SYSTEM('subsys')'
       lpla_ppk = length('BIND PLAN('pla_ppk')')
       wrk = 5Ø - lpla_ppk
sk.2='BIND PLAN('pla_ppk')'copies(' ',wrk)'-'
sk.3='PKLIST (                                          -'
       lpak_ppk = length('   'col_ppk'.'pak_ppk' ,')
       wrk = 5Ø - lpak_ppk
sk.4='   'col_ppk'.'pak_ppk' ,'copies(' ',wrk)'-'
       #c1 = #c + 1
       pla_com = word(sysprint.#c1,2)
       pak_ppk = word(sysprint.#c1,3)
       col_ppk = word(sysprint.#c1,4)
       #g = 4
       sw = off
       do while pla_ppk = pla_com
          sw = on
          #c = #c + 1
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
          pla_com = word(sysprint.#c,2)
          pak_ppk = word(sysprint.#c,3)
          col_ppk = word(sysprint.#c,4)
          if pla_ppk = pla_com then do
             #g = #g + 1
             lpak_ppk = length('   'col_ppk'.'pak_ppk' ,')
             wrk = 5Ø - lpak_ppk
sk.#g='   'col_ppk'.'pak_ppk' ,'copies(' ',wrk)'-'
             end
          end
       if sw = off then
          #c = #c + 1
       #g = #g + 1
sk.#g='       )                                              - '
       lcre_ppk = length('OWNER('cre_ppk')')
       wrk = 5Ø - lcre_ppk
       #g = #g + 1
sk.#g='OWNER('cre_ppk')'copies(' ',wrk)'-'
       lqua_ppk = length('QUALIFIER('qua_ppk')')
       wrk = 5Ø - lqua_ppk
       #g = #g + 1
sk.#g='QUALIFIER('qua_ppk')'copies(' ',wrk)'-'
       #g = #g + 1
sk.#g='ACT(REPLACE)                                          -'
       lvalid_ppk = length('VALIDATE('valid_ppk')')
       wrk = 5Ø - lvalid_ppk
       #g = #g + 1
sk.#g='VALIDATE('valid_ppk')'copies(' ',wrk)'-'
       lisol_ppk = length('ISOLATION('isol_ppk')')
       wrk = 5Ø - lisol_ppk
       #g = #g + 1
sk.#g='ISOLATION('isol_ppk')'copies(' ',wrk)'-'
       lacqu_ppk = length('ACQUIRE('acqu_ppk')')
       wrk = 5Ø - lacqu_ppk
       #g = #g + 1
sk.#g='ACQUIRE('acqu_ppk')'copies(' ',wrk)'-'
       lrele_ppk = length('RELEASE('rele_ppk')')
       wrk = 5Ø - lrele_ppk
       #g = #g + 1
sk.#g='RELEASE('rele_ppk')'copies(' ',wrk)'-'
       #g = #g + 1
sk.#g='EXPLAIN('expl_ppk')'
       sk.Ø = #g
       call Writeout
       ctrppk = ctrppk + 1
       #c = #c - 1
       end
   /*————————————————————————————————————————————*/
   /*—      Build Bind job                       ———*/
   /*————————————————————————————————————————————*/
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
sb.#x='//* ------------------------------ *      '
    sb.Ø = #x
    jobw = fiupd
    "alloc da('"outdsupd"(bindppk)') f("jobw") mod reuse"
    call Wrindx
   /*-------------------------------------------------*/
   /*-    IPOUPDTE job for .PLANPACK library   --*/
   /*-------------------------------------------------*/
    jobw = fiupd
    "alloc da('"outdsupd"(ipoupppk)') f("jobw") mod reuse"
    call hdripo
sk.19='//IPOBPPPK EXEC PGM=IPOUPDTE,                                    '
sk.2Ø='//*       PARM='UPDATE'              /* Run with  UPDATE       '
sk.21='//*       PARM='CHECK'               /* Run without UPDATE     '
sk.22='//STEPLIB  DD  DSN=IPO1.LINKLIB,DISP=SHR                         '
sk.23='//SYSPRINT DD  SYSOUT=*                                          '
sk.24='//@INST    DD  DSN='outdsppk',DISP=SHR                           '
sk.25='//SYSIN    DD  DATA,DLM=@@                                       '
sk.26='</*                                                              '
sk.27=''subsys'<+                                                       '
sk.28=' ????<                           /* New subsystem DB2          '
sk.29=''cre_ppk'<+                                                      '
sk.3Ø=' ?????????<                      /* New Owner                  '
sk.31=''qua_ppk'<+                                                      '
sk.32=' ?????????<                      /* New Qualifier              '
sk.33='REPLACE<+                                                        '
sk.34=' ????????<                       /* Action Replace/Add/Retain'
sk.35=''valid_ppk'<+                                                    '
sk.36=' ????<                           /* Validate RUN/BIND          '
sk.37=''isol_ppk'<+                                                     '
sk.38=' ??<                             /* Isolation RR/CS            '
sk.39=''rele_ppk'<+                                                     '
sk.4Ø=' ??????????<                     /* Release Commit/Deallocate'
sk.41=''expl_ppk'<+                                                     '
sk.42=' ???<                            /* Explain YES/NO             '
sk.43=''acqu_ppk'<+                                                     '
sk.44=' ??????????<                     /* Acquire Use/Dealloacte     '
sk.45='</*                                                              '
    sk.Ø=45
    Call Writeout
    return
   /*-------------------------------------------------*/
   /*-    Write bind plan members with DBRM   ---*/
   /*-------------------------------------------------*/
 Wrbpmemb:
    say '>>>>>>>>  Building library 'outdsppm
    jobw = fippm
    "alloc da('"outdsppm"($$$coibm)') f("jobw") shr reuse"
sk.1='     ////////////////////////////////////////////////////// '
sk.2='     ****** Do not erase this member !!!!     Thanks.  ****** '
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
sk.3='     ////////////////////////////////////////////////////// '
    sk.Ø = 3
    call Writeout
    #x  = 1
    #x1 = Ø
    Call Hdrbnd
    DO #e = 1 to sysprint.Ø
       pla_ppm = word(sysprint.#e,1)
       dbr_ppm = word(sysprint.#e,2)
       cre_ppm = word(sysprint.#e,3)
       qua_ppm = word(sysprint.#e,4)
       val_ppm = word(sysprint.#e,5)
       acq_ppm = word(sysprint.#e,6)
       iso_ppm = word(sysprint.#e,7)
       rel_ppm = word(sysprint.#e,8)
       exp_ppm = word(sysprint.#e,9)
       pds_ppm = word(sysprint.#e,1Ø)
   /*———————————————————————————————————————————*/
   /*—     VALIDATE  "Bind/Run"                ———*/
   /*———————————————————————————————————————————*/
       if val_ppm = B then
          valid_ppm = BIND
       else
          valid_ppm = RUN
   /*———————————————————————————————————————————*/
   /*—    ACQIRE    "Use/Allocate"             ———*/
   /*———————————————————————————————————————————*/
       if acq_ppm = A then
          acqu_ppm = ALLOCATE
       else
          acqu_ppm = USE
   /*———————————————————————————————————————————*/
   /*—   ISOLATION "Rep.Read/Cursor Stability   —*/
   /*———————————————————————————————————————————*/
       if iso_ppm = R then
          isol_ppm  = RR
       else
          isol_ppm  = CS
   /*———————————————————————————————————————————*/
   /*—   RELEASE   "Deallocate/Commit"         ———*/
   /*———————————————————————————————————————————*/
       if rel_ppm = D then
          rele_ppm  = DEALLOCATE
       else
          rele_ppm  = COMMIT
   /*———————————————————————————————————————————*/
   /*—    EXPLAIN   "YES/NO"                   ———*/
   /*———————————————————————————————————————————*/
       if exp_ppm = Y then
          expl_ppm  = YES
       else
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
          expl_ppm  = NO
   /*————————————————————————————————————————————*/
   /*—    Build Bind job                       ———*/
   /*————————————————————————————————————————————*/
       if swsyspr = on then do
sb.#x='//SYSTSIN   DD  DSN='outdsppm'('pla_ppm'),DISP=SHR             '
          #x = #x + 1
          swsyspr = off
          end
       else do
sb.#x='//          DD  DSN='outdsppm'('pla_ppm'),DISP=SHR             '
          #x = #x + 1
          end
       if #x1 > 8Ø then do
sb.#x='//* ————————————————————— *      '
          #x = #x + 1
          call hdrbnd
          #x1 = Ø
          end
       #x1 = #x1 + 1
       jobw = fippm
       "alloc da('"outdsppm"("pla_ppm")') f("jobw") shr reuse"
sk.1='DSN SYSTEM('subsys')'
       lpla_ppm = length('BIND PLAN('pla_ppm')')
       wrk = 5Ø - lpla_ppm
sk.2='BIND PLAN('pla_ppm')'copies(' ',wrk)'-'
       lpds_ppm = length('LIBRARY(''pds_ppm'')')
       wrk = 5Ø - lpds_ppm
sk.3='LIBRARY(''pds_ppm'')'copies(' ',wrk)'-'
sk.4='MEMBER (                                              -'
       ldbr_ppm = length('   'dbr_ppm' ,')
       wrk = 5Ø - ldbr_ppm
sk.5='   'dbr_ppm' ,'copies(' ',wrk)'-'
       #e1 = #e + 1
       pla_com = word(sysprint.#e1,1)
       dbr_ppm = word(sysprint.#e1,2)
       #g = 5
       sw = off
       do while pla_ppm = pla_com
          sw = on
          #e = #e + 1
          pla_com = word(sysprint.#e,1)
          dbr_ppm = word(sysprint.#e,2)
          if pla_ppm = pla_com then do
             #g = #g + 1
             ldbr_ppm = length('   'dbr_ppm' ,')
             wrk = 5Ø - ldbr_ppm
sk.#g='   'dbr_ppm' ,'copies(' ',wrk)'-'
             end
          end
       if sw = off then
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
          #e = #e + 1
        #g = #g + 1
sk.#g='       )                                                - '
        lcre_ppm = length('OWNER('cre_ppm')')
        wrk = 5Ø - lcre_ppm
        #g = #g + 1
sk.#g='OWNER('cre_ppm')'copies(' ',wrk)'-'
        lqua_ppm = length('QUALIFIER('qua_ppm')')
        wrk = 5Ø - lqua_ppm
        #g = #g + 1
sk.#g='QUALIFIER('qua_ppm')'copies(' ',wrk)'-'
        #g = #g + 1
sk.#g='ACT(REPLACE)                                            -'
        lvalid_ppm = length('VALIDATE('valid_ppm')')
        wrk = 5Ø - lvalid_ppm
        #g = #g + 1
sk.#g='VALIDATE('valid_ppm')'copies(' ',wrk)'-'
        lisol_ppm = length('ISOLATION('isol_ppm')')
        wrk = 5Ø - lisol_ppm
        #g = #g + 1
sk.#g='ISOLATION('isol_ppm')'copies(' ',wrk)'-'
        lacqu_ppm = length('ACQUIRE('acqu_ppm')')
        wrk = 5Ø - lacqu_ppm
        #g = #g + 1
sk.#g='ACQUIRE('acqu_ppm')'copies(' ',wrk)'-'
        lrele_ppm = length('RELEASE('rele_ppm')')
        wrk = 5Ø - lrele_ppm
        #g = #g + 1
sk.#g='RELEASE('rele_ppm')'copies(' ',wrk)'-'
        #g = #g + 1
sk.#g='EXPLAIN('expl_ppm')'
        sk.Ø = #g
        call Writeout
        ctrppm = ctrppm + 1
        #e = #e - 1
        end
    /*————————————————————————————————————————————*/
    /*–      Build Bind job                      ———*/
    /*————————————————————————————————————————————*/
sb.#x='//* ————————————————————————— *      '
     sb.Ø = #x
     jobw = fiupd
     "alloc da('"outdsupd"(bindppm)') f("jobw") mod reuse"
     call Wrindx
    /*————————————————————————————————————————————*/
    /*–      IPOUPDTE job for .PLAN library    ———*/
    /*————————————————————————————————————————————*/
     jobw = fiupd
     "alloc da('"outdsupd"(ipoupppm)') f("jobw") mod reuse"
     call hdripo
sk.19='//IPOUPPPM EXEC PGM=IPOUPDTE,                             '
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
sk.20='//*       PARM='UPDATE'              /* Run with  UPDATE       '
sk.21='//*       PARM='CHECK'               /* Run without UPDATE     '
sk.22='//STEPLIB  DD  DSN=IPO1.LINKLIB,DISP=SHR                      '
sk.23='//SYSPRINT DD  SYSOUT=*                                       '
sk.24='//@INST    DD  DSN='outdsppm',DISP=SHR                        '
sk.25='//SYSIN    DD  DATA,DLM=@@                                    '
sk.26='</*                                                           '
sk.27=''subsys'<+                                                    '
sk.28=' ????<                           /* New subsystem DB2        '
sk.29=''cre_ppm'<+                                                   '
sk.30=' ????????<                       /* New Owner                '
sk.31=''qua_ppm'<+                                                   '
sk.32=' ????????<                       /* New Qualifier            '
sk.33=''acqu_ppm'<+                                                  '
sk.34=' ??????????<                     /* Acquire Use/Dealloacte   '
sk.35=''valid_ppm'<+                                                 '
sk.36=' ????<                           /* Validate RUN/BIND        '
sk.37=''isol_ppm'<+                                                  '
sk.38=' ??<                             /* Isolation RR/CS          '
sk.39=''rele_ppm'<+                                                  '
sk.40=' ??????????<                     /* Release Commit/Deallocate'
sk.41='REPLACE<+                                                     '
sk.42=' ???????<                        /* Action Replace/Add/Retain'
sk.43=''expl_ppm'<+                                                  '
sk.44=' ???<                            /* Explain YES/NO           '
sk.45=''pds_ppm'<+                                                   '
sk.46=' ????????????????<               /* New PDS for dbrms        '
sk.47='</*                                                           '
    sk.0=47
    Call Writeout
    return
   /*-------------------------------------------------------*/
   /*-      Header job IPOUPDTE                 ----*/
   /*-------------------------------------------------------*/
 Hdripo:
sk.1= '//'jna'I JOB ('account'),''IPOUPDTE JOB'',CLASS=S,MSGCLASS=X,  '
sk.2= '//      USER='user',REGION=3M,MSGLEVEL=(1,1),NOTIFY='notif'    '
sk.3= '/*JOBPARM BYTES=999999,LINES=9999                              '
sk.4= '//* ---------------------------- *                            '
sk.5= '//*   This job is used to change some value in members  *     '
sk.6= '//*   of bind :                                         *     '
sk.7= '//*                                                     *     '
sk.8= '//*    -  change the characters ''????????'' with the   *     '
sk.9= '//*       new value                                     *     '
sk.10='//*   -   do not modify the characters ''</*,<+,<''     *     '
sk.11='//*       because they are used from IPOUPDTE           *     '
sk.12='//*   -   Select the way to use the pgm IPOUPDTE        *     '
sk.13='//*       uncomment one of the following parameters     *     '
sk.14='//*                                                     *     '
sk.15='//*            //*       PARM=UPDATE                    *     '
sk.16='//*            //*       PARM=CHECK                     *     '
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
sk.17='//*                                                              *      '
sk.18='//* ------------------------------------------------------------  *      '
    return
   /*-------------------------------------------------------*/
   /*-      Header job BIND                                 --*/
   /*-------------------------------------------------------*/
 Hdrbnd:
 swsyspr= on
sb.#x='//'jna'B JOB ('account'),'' BIND JOB '',CLASS=S,MSGCLASS=X,    '
 #x = #x + 1
sb.#x='//      USER='user',REGION=3M,MSGLEVEL=(1,1),NOTIFY='notif'    '
 #x = #x + 1
sb.#x='/*JOBPARM BYTES=999999,LINES=9999                              '
 #x = #x + 1
sb.#x='//* ------------------------------------------------------------  *      '
 #x = #x + 1
sb.#x='//*               Full  job   B  I  N  D                       *      '
 #x = #x + 1
sb.#x='//* ------------------------------------------------------------  *      '
 #x = #x + 1
sb.#x='//JOBLIB    DD  DSN=SYS1.DSN310.SDSNLOAD,DISP=SHR              '
 #x = #x + 1
sb.#x='//* ------------------------------------------------------------  *      '
 #x = #x + 1
sb.#x='//BIND  EXEC  PGM=IKJEFT01,DYNAMNBR=100                        '
 #x = #x + 1
sb.#x='//SYSTSPRT  DD  SYSOUT=*                                       '
 #x = #x + 1
sb.#x='//REPORT    DD  SYSOUT=*                                       '
 #x = #x + 1
    return
   /*-------------------------------------------------------*/
   /*-      File ipoupdte allocation                       --*/
   /*-------------------------------------------------------*/
 Alldelt:
    outdsdlt= hiwork'.'subsys'.'creator'.DELTA'
    xx=outtrap(trp15.)
       address tso "delete '"outdsdlt"'"
       "alloc da('"outdsdlt"') dir(0) space(15,150) dsorg(ps)" ,
       "recfm(f,b,a) lrecl(121) blksize(1210)  tracks ",
       "unit("esounit") release new catalog f(outdd) "
    xx=outtrap(off)
    if rc > 0 then do
       do #a = 1 to trp15.0
          say trp15.#a
          end
       say '        '
       say '        '
       say '>>>>>>>>'
       say '>>>>>>>> "'outdsdlt'"  Allocation OK'
       say '>>>>>>>>  RC='rc'. Verify.          '
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
       say '>>>>>>>>'
       say '        '
       say '        '
       exit
       end
    else
       say '>>>>>>>>  'outdsdlt '           Allocation OK  '
       say '        '
    jobw = fippk
    "alloc da('"outdsppk"(Z9999999)') f("jobw") shr reuse"
sk.1='     ////////////////////////////////////////////////////// '
sk.2='     ****** Do not erase this member  !!!!    Thanks.  ****** '
sk.3='     ////////////////////////////////////////////////////// '
    sk.Ø = 3
    call Writeout
    jobw = fippm
    "alloc da('"outdsppm"(Z9999999)') f("jobw") shr reuse"
sk.1='     ////////////////////////////////////////////////////// '
sk.2='     ****** Do not erase this member  !!!!    Thanks.  ****** '
sk.3='     ////////////////////////////////////////////////////// '
    sk.Ø = 3
    call Writeout
    return
   /*———————————————————————————————————————————*/
   /*—      Analyse output compare            ———*/
   /*———————————————————————————————————————————*/
 Wrcomp:
    #r = Ø
    sw= off
    do #h = 1 to trpdl.Ø
       if sw = on then do
          sb.#r = word(trpdl.#h,1)
          #r = #r + 1
          end
       if trpdl.#h = '—MEMBERS—' then do
          sw = on
          #r = #r + 1
          end
       end
    sb.Ø=#r
    call Wrindx
    return
   /*———————————————————————————————————————————*/
   /*—      Write routine output record       ———*/
   /*———————————————————————————————————————————*/
 Writeout :
    "EXECIO * DISKW "jobw" (STEM sk. FINIS"
 Pulisci:
    DO #f = 1 to sk.Ø
       sk.#f = blk
       end
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
    return
   /*-------------------------------------------------------*/
   /*-      Write routine job Bind                         -*/
   /*-------------------------------------------------------*/
 Wrindx :
    "EXECIO * DISKW "jobw" (STEM sb. FINIS"
    DO #f = 1 to sb.Ø
       sb.#f = blk
       end
    return
   /*-------------------------------------------------------*/
   /*-      Free datasets                                  -*/
   /*-------------------------------------------------------*/
 Free    :
    "free fi(systsinp)"
     address tso "delete '"outds1"'"
     "free fi(fiupd)"
    if oper  = YYN then do
       "free fi(fippk)"
       "free fi(fipk)"
       end
    if oper  = YYY then do
       "free fi(fippk)"
       "free fi(fipk)"
       "free fi(fippm)"
       end
    if oper  = NNY then
       "free fi(fippm)"
    return
```

MDB2016 EDIT MACRO

```
isredit macro
isredit exclude all
isredit find '_|' all
isredit find ' Ø ROW'
isredit delete x all
isredit change P'======' '     ' 3 all
isredit change '|'  ' '  all
isredit save
isredit end
```

MDB2018 EDIT MACRO

```
isredit macro
isredit exclude all
isredit find '_|' all
isredit find ' Ø ROW'
isredit delete x all
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
isredit change P'======' '     ' 13 all
isredit change '|'  ' '  all
isredit save
isredit end
```

MDB2021 EDIT MACRO

```
/* REXX */
isredit macro
isredit exclude '''DSN SYSTEM'''
isredit exclude '''BIND PLAN'''
isredit exclude '''OWNER'''
isredit exclude '''QUALIFIER'''
isredit exclude '''ACT(REPLACE)'''
isredit exclude '''VALIDATE'''
isredit exclude '''ISOLATION'''
isredit exclude '''ACQUIRE'''
isredit exclude '''RELEASE'''
isredit exclude '''EXPLAIN'''
isredit delete x all
isredit save
isredit end
```

MDB2023 EDIT MACRO

```
isredit macro
isredit exclude all
isredit find '_|' all
isredit find ' Ø ROW'
isredit delete x all
isredit change P'======' '     ' 6 all
isredit change '|'  ' '  all
isredit save
isredit end
```

MDB2024 EDIT MACRO

```
/* REXX */
address ispexec 'VGET (movmemb)  PROFILE'
isredit macro
isredit move ''movmemb'' after 2
isredit save
isredit end
```

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

Giuseppe Rendano
DB2 System Programmer (Italy)

© Xephon 1998

 © 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

DB2 utility services

In a rapidly changing DB2 environment, the creation and maintenance of DB2 utility jobs can be both time-consuming and difficult to keep current. My DB2 services provide you with an easy way to execute any DB2 utility. Using the DBUT procedure, you simply enter the required parameters and submit the job for immediate or deferred execution.

The utilities supported by DBUT are:

- CHECK
- COPY
- MODIFY
- QUIESCE
- RECOVER
- REORG
- REPORT
- RUNSTATS
- STOSPACE.

The main menu is shown in Figure 1.

When any service is selected from the DB2 utilities main menu, the parameter entry panel is displayed – this is where you specify the parameters that control the operation of any service.

The CHECK service builds a CHECK analysis job from the parameters you supply, combined with the DB2 catalog information. Using the CHECK utility, you can check the status of data, indexes, or both. The CHECK DATA utility checks table spaces for violations of referential and table check constraints, and reports information about any violations detected. The CHECK INDEX utility tests whether indexes are consistent with the data they index, and issues warning messages when an inconsistency is found.

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

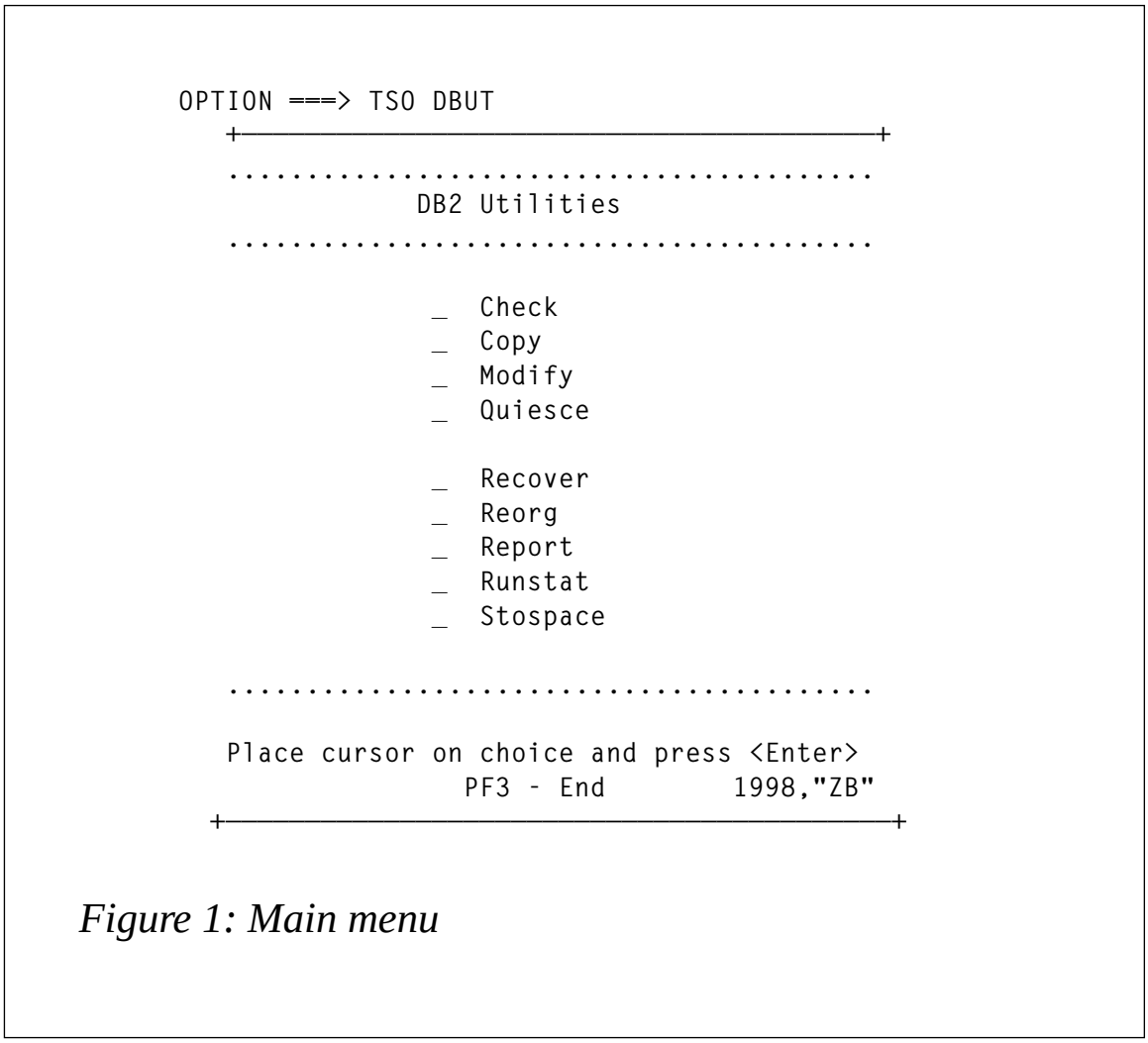

Figure 1: Main menu

The COPY service creates an image copy of a table space or a dataset within a table space. You can specify whether you want a full or incremental copy. A full copy copies all the data in a table space and an incremental copy copies only the data that has been modified since the last COPY was executed. The COPY service copies by default all partitions of a partitioned table space.

You can control which partitions are copied through an on-line dialogue.

Figure 2 shows the copy parameter entry panel. To the right of the panel is the prompt column, which should help in explaining the values to enter in the parameter value column.

The MODIFY service deletes records from the SYSIBM.SYSCOPY

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

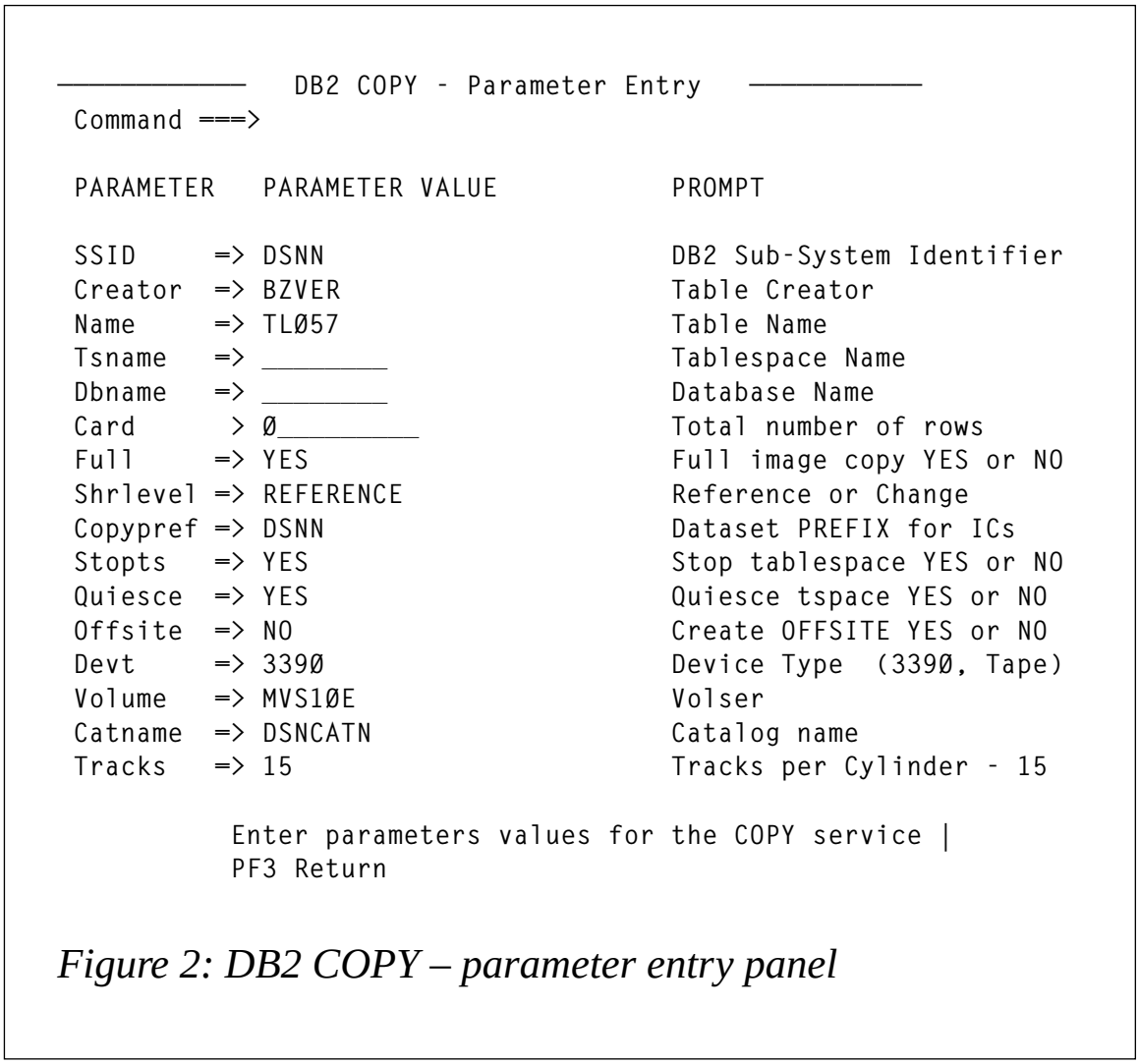

Figure 2: DB2 COPY – parameter entry panel

catalog table and related log records from the SYSIBM.SYSLGRNX directory table. You can remove records that were written before a specific date or you can remove records of a specific age. You can also delete datasets for image copies.

The QUIESCE service establishes a quiesce point for a table space or partition and records it in in the SYSIBM.SYSCOPY catalog table. A successful QUIESCE improves the probability of a successful RECOVER. You should run QUIESCE frequently between executions of COPY to establish recovery points for future recovery.

The RECOVER service recovers data to the current state or to a previous point in time. By default, the RECOVER service recovers all partitions of a partitioned table space, but you can control which

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

partitions are recovered through an on-line dialogue. The RECOVER service selects different table spaces to recover, based on the recovery type option (RBA,TOCOPY, or default) and also includes the RECOVER INDEX utility.

The REORG service reorganizes a table space to improve access performance and reclaim fragmented space. In addition, the service can reorganize all partitions of a partitioned table space, but you can control which partitions are recovered through an on-line dialogue. You can specify the entry parameters to determine the REORG service (image copy before/after reorg, quiesce, or runstat).

The REPORT service provides information about table spaces – information about recovery history from the SYSIBM.SYSCOPY table, log ranges from SYSIBM.SYSLGRNG, bootstrap dataset information, and names of all table spaces and tables in a table space set. It also lists all tables dependent upon those tables.

The RUNSTST service enables you to run statistics for a table space based on the parameters you enter. This information is recorded in the DB2 catalog and is used by DB2 to select access paths to data during the bind process. It is available to the database administrator for evaluating database design and determining when table spaces or indexes must be reorganized.

The STOSPACE service updates DB2 catalog columns that indicate how much space is allocated for storage groups and related table spaces and indexes.

COMPONENTS OF DBUT

The components of DBUT are as follows.

CLIST

CLIST comprises:

- DBUT – the driver procedure.
- DBUTIL1 – the procedure to build an image copy, check data/ index, RECOVER utility and REORG utility.

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

- DBUTIL2 – the procedure to build a QUIESCE, REPORT utility, and RUNSTAT utility.
- DBMOD – the procedure to build a MODIFY recovery job.
- DBSTO – the procedure to build a STOSPACE job.

Panels

The panels comprise:

- DBUTILM – the main menu.
- DBUTUM – the message display.
- DBCHEM – the CHECK panel.
- DBCOPM – the COPY panel.
- DBMODM – the MODIFY panel.
- DBQUIM – the QUIESCE panel.
- DBRECM – the RECOVER panel
- DBREPM – the REPORT panel.
- DBRUNM – the RUNSTAT panel.
- DBREOM – the REORG panel.
- DBSTOM – the STOSPACE panel.
- DBULIST – the selection result panel for CHECK, COPY, QUIESCE, REPORT, and RUNSTST.
- DBMLIST – the selection result panel for MODIFY.
- DBRLIST – the selection result panel for RECOVER.
- DBSLIST – the selection result panel for STOSPACE.
- DBPARTS – the selection partition tablespaces panel.

Message

The message component of DBUT compises:

- DBUT00 – DBUT message.

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

ISPLLIB

ISPLLIB comprises:

- PDBUTIL – PL/I program for CHECK, COPY, QUIESCE, REPORT, and RUNSTAT.
- PDBMODI – PL/I program for MODIFY utility.
- PDBRECO – PL/I program for RECOVER utility.
- PDBSTOS – PL/I program for STOSPACE job.

SYSLIB

SYSLIB comprises:

- DBCHECK – JCL CHECK skeleton.
- DBCOPY – JCL COPY skeleton.
- DBMODI – JCL MODIFY skeleton.
- DBQUIE – JCL QUIESCE skeleton.
- DBRECO – JCL RECOVER skeleton.
- DBREPO – JCL REPORT skeleton.
- DBRUNS – JCL RUMSTST skeleton.
- DBREORG – JCL REORG skeleton.
- DBSTOS – JCL STOSPACE skeleton.

DBUT

```
/* REXX *//* trace r */
 zpfctl = 'OFF'
 address ispexec 'vput (zpfctl) profile'
 address ispexec 'addpop row(1) column(10)'
 CUR='ch'
 address ispexec "display panel(dbutilm) cursor("CUR")"
 do while rc=0
    if kurs='CH' | kurs='CHE' then do
         address ispexec rempop all
         Call dbutil1 CHECK
         CUR='ch'
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
     address ispexec 'addpop row(1) column(10)'
end
if kurs='CO' | kurs='COP' then do
     address ispexec rempop all
     Call dbutil1 COPY
     CUR='co'
     address ispexec 'addpop row(1) column(10)'
end
if kurs='MO' | kurs='MOD' then do
     address ispexec rempop all
     Call dbmod
     CUR='mo'
     address ispexec 'addpop row(1) column(10)'
end
if kurs='QU' | kurs='QUI' then do
     address ispexec rempop all
     Call dbutil2 QUIESCE
     CUR='qu'
     address ispexec 'addpop row(1) column(10)'
end
if kurs='RE' | kurs='REC' then do
     address ispexec rempop all
     Call dbutil1 RECOVER
     CUR='re'
     address ispexec 'addpop row(1) column(10)'
end
if kurs='RO' | kurs='REO' then do
     address ispexec rempop all
     Call dbutil1 REORG
     CUR='ro'
     address ispexec 'addpop row(1) column(10)'
end
if kurs='RP' | kurs='REP' then do
     address ispexec rempop all
     Call dbutil2 REPORT
     CUR='rp'
     address ispexec 'addpop row(1) column(10)'
end
if kurs='RU' | kurs='RUN' then do
     address ispexec rempop all
     Call dbutil2 RUNSTATS
     CUR='ru'
     address ispexec 'addpop row(1) column(10)'
end
if kurs='ST' | kurs='STO' then do
     address ispexec rempop all
     Call dbsto
     CUR='st'
     address ispexec 'addpop row(1) column(10)'
end
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
   address ispexec "display panel(dbutilm) cursor("CUR")"
 end
 exit
```

DBUTIL1

```
/* REXX *//* Build an image copy, check data/index,                  */
/*           recover utility and reorg utility                      */
/* trace r */
 ARG util
 zpfctl = 'OFF'
 Y=MSG("OFF")
 /*********************************************************/
 /* Change to your convention standards                   */
 program = 'PDBUTIL'
 if util='RECOVER' then program='PDBRECO'
 plan    = 'PDBUTIL'
 if util='RECOVER' then plan   ='PDBRECO'
 llib    = 'SKUPNI.BATCH.LOADLIB'
 /*********************************************************/
 address ispexec 'vput (zpfctl) profile'
 Call Aloc
 head='DB2 '||util||' - Selection Result'
 cur='crec'
 Call Create_messg
 TOP:
 address ispexec "display panel(DB"substr(util,1,3)"M) cursor("CUR")"
 if rc=8 then do
    Call Free_proc
    address ispexec "tbclose "messdb""
    exit
 end
 /* Check for CHECK utility                              */
 if util = 'CHECK' then do
    if typ='INDEX' | typ='DATA' | typ='BOTH' then nop
    else do
       message=,
       'Invalid TYPE value. Valid values are: INDEX, DATA or BOTH.'
       Call Error 'typ'
    end
    if scop='PENDING' | scop='ALL' then nop
    else do
       message=,
       'Invalid SCOPE value. Valid values are: PENDING or ALL.'
       Call Error 'scop'
    end
 end
 /* Check input parameters                               */
 if crec=' ' & tabc=' ' & tsnc=' ' & dbnc=' ' then do
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
   message='At least one Catalog search field must be entered.'
   Call Error 'crec'
end
/* Check for COPY utility                              */
if util = 'COPY' then do
   car = verify(card,' Ø123456789')
   IF car > Ø then do
      message='Enter blank or Card value '||,
              'greater than number of rows in the table'
      Call Error 'card'
   end
   if ful='YES' | ful='NO' then nop
   else do
      message='Valid Full values for Full or Incremental '||,
              'image copy are: YES,NO.'
      Call Error 'ful'
   end
   if ref='REFERENCE' | ref='CHANGE' then nop
   else do
      message='Invalid SHRLEVEL value. '||,
              'Valid values are: REFERENCE,CHANGE.'
      Call Error 'ref'
   end
   if pref=' '
   then do
      message='Invalid Copypref value. '||,
              'Dataset PREFIX for image copies required.'
      Call Error 'pref'
   end
   Call Tspace_stop
   if off='YES' | off='NO' then nop
   else do
      message='Invalid Indicator to create '||,
              'OFFSITE IC'||"'"||'s. Valid values are: YES,NO.'
      Call Error 'off'
   end
   if dev='339Ø' | dev='TAPE' then nop
   else do
      message='Devt is a required field. '||,
              'Valid values are: 339Ø,TAPE.'
      Call Error 'dev'
   end
end
/* Check for RECOVER utility                           */
if util = 'RECOVER' then do
   if rtyp='RBA' | rtyp='TOCOPY' then no_rec=Ø
   if rtyp='CUR' | rtyp='RBA' | rtyp='TOCOPY' then nop
   else do
      message=,
      'Invalid Rectype value. Valid values are: CUR, RBA or TOCOPY.'
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
      Call Error 'rtyp'
   end
   if (rtyp='RBA' | rtyp='TOCOPY') & (dfr=' ' | dto=' ')
   then do
      message='RECOVER to '||rtyp||' was specified, '||,
              'but Datefrom and Dateto not supplied.'
      Call Error 'dfr'
   end
   if (rtyp='RBA' | rtyp='TOCOPY') &,
      (verify(dfr,'Ø123456789')>Ø | length(dfr) ¬=8 )
   then do
      message='Enter a valid DATE in a YYYYMMDD format.'
      Call Error 'dfr'
   end
   if (rtyp='RBA' | rtyp='TOCOPY') &,
      (verify(dto,'Ø123456789')>Ø | length(dto) ¬=8 )
   then do
      message='Enter a valid DATE in a YYYYMMDD format.'
      Call Error 'dto'
   end
end
/* Check for REORG utility                              */
if util = 'REORG' then do
   if ico='BEFORE' | ico='AFTER' | ico='BOTH' | ico='NONE' then nop
   else do
      message='Invalid Image Copy value. '||,
              'Valid values are: Before, After, Both or None.'
      Call Error 'ico'
   end
   if pref=' ' & ico ¬='NONE'
   then do
      message='Invalid Copypref value. '||,
              'Dataset PREFIX for image copies required.'
      Call Error 'pref'
   end
   if log='YES' | log='NO' then nop
   else do
      message='Invalid LOG parameter. Valid values are: YES, NO.'
      Call Error 'log'
   end
   if sor='YES' | sor='NO' then nop
   else do
      message='Invalid SORTDATA parameter. Valid values are: YES, NO.'
      Call Error 'sor'
   end
   if dic='YES' | dic='NO' then nop
   else do
      message='Invalid KEEPDICTIONARY parameter. '||,
              'Valid values are: YES, NO.'
      Call Error 'dic'
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
   end
   Call Tspace_stop
   if rru='YES' | rru='NO' then nop
   else do
      message='Invalid RUNSTATS parameter. Valid values are: YES, NO.'
      Call Error 'rru'
   end
end
if vol=' '
then do
   message='Invalid Volser value. '
   if dev='339Ø' then message=message||,
                 'Disk Volume value required (Devt=339Ø).'
   if dev='TAPE' then message=message||,
                 'Tape Volume value required (Devt=TAPE).'
   Call Error 'vol'
end
if catn=' '
then do
   message='Invalid Catname value. '||,
           'DB2 VSAM Catalog name required.'
   Call Error 'catn'
end
if trk=' '
then do
   message='Invalid Tracks value. '||,
           'Tracks per Cylinder required.'
   Call Error 'trk'
end
ind = verify(trk,'Ø123456789')
IF ind > Ø | trk < 1 then do
   message='Enter a valid NUMBER for Tracks parameter.'
   Call Error 'trk'
end
if util='COPY' then
parm=substr(crec,1,8)||substr(tabc,1,18)||substr(tsnc,1,8)||,
 substr(dbnc,1,8)||substr(sto,1,3)||substr(que,1,3)||substr(card,1,1Ø)
if util='CHECK' | util='REORG' then
parm=substr(crec,1,8)||substr(tabc,1,18)||substr(tsnc,1,8)||,
     substr(dbnc,1,8)||'   YES          '
if util='RECOVER' then
parm=substr(crec,1,8)||substr(tabc,1,18)||substr(tsnc,1,8)||,
     substr(dbnc,1,8)||substr(rtyp,1,6)||dfr||dto
messg = "Accessing  db2 system "db2""
messg = time() || " " || messg
Call Send_messg
messg = 'Select     systables      information'
messg = time() || " " || messg
Call Send_messg
ADDRESS TSO
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
QUEUE "RUN PROGRAM("program") PLAN("plan"),
       LIBRARY ('"llib"'),
       PARMS ('/"parm"')"
QUEUE "END "
"DSN SYSTEM("db2")"
if rc=12 then do
   "delstack"
   Call Free_proc
   Call Aloc
   address ispexec 'tbend messdb'
   Call Create_messg
   message = 'Error.   'db2||' ssid is not valid  |'
   Call Error 'db2'
END
"EXECIO * DISKR SYSPRINT (STEM ROW."
if substr(row.1,2) = 'NO CATALOG ENTRIES FOUND' then do
   Call Free_proc
   Call Aloc
   address ispexec 'tbend messdb'
   Call Create_messg
   message = 'No catalog entries found, check Search Fields.'
   Call Error 'crec'
end
else do
   address ispexec 'addpop row(1) column(5)'
   address ispexec 'tbcreate "blist" names(v1 v2 v3 v4 v5 v6)'
   count=0
   num=row.0
   do i=1 to row.0
      if substr(row.i,2,1)='A' then count=count+1
      if substr(row.i,2,1)='B' then do
         v6=' '
         v1= substr(row.i,3,8)
         v2= word(row.i,2)
         v3= word(row.i,3)
         v4= word(row.i,4)
         v5= right(word(row.i,5),13)
         if v5 = '-1' then v6='Runstat'
         address ispexec 'tbadd "blist"'
      end
   end
   address ispexec 'tbtop "blist"';
   address ispexec 'tbdispl "blist" panel(dbulist)'
   if rc=8 then do
      Call Free_proc
      address ispexec 'tbend "blist"'
      Call Aloc
      address ispexec rempop all
      address ispexec 'tbend messdb'
      Call Create_messg
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
      signal top
   end
   address ispexec rempop all
end
ctime=time('s')
messg = 'Calculating Tablespace Dataset Sizes'
messg = time() || " " || messg
Call Send_messg
Call Free_proc
rba=' '
icdsn=' '
address ispexec,
'tbcreate "alist" names(db ts pts pri sec detail scu rbadsn)'
asterisks= '************************************************'
procent=100/(2*count)
tot=0
cyl=0
scu=0
varx='NE'
do i=1 to row.0 while(substr(row.i,2,1)='A')
   db = substr(row.i,3,8)
   ts = word(row.i,2)
   if strip(db) = 'DSNDB01' & ts = 'SYSUTILX'
   then varx='JA'
   pr = word(row.i,3)
   pri=0
   sec=0
   alldata=''
   apart='NO'
   if pr>1 then do
      do j=1 to pr
         prow.j='{'||left(db,9)||left(ts,10)||right(j,4)||'   #I{ '
         alldata=alldata||prow.j
      end
      Call Parts_panel
   end
   if apart='NO' then do
      pts='S'
      do j=1 to pr
         part='.I0001.A'||right(j,3,'0')
         file=catn||'.DSNDBD.'||strip(db)||'.'||strip(ts)||part
         dsn = "('"file"')"
         X=OUTTRAP('var.')
         address tso "listc" entries dsn allocation
         X=OUTTRAP('OFF')
         Call Check_dsn
         if rc=0 then do
            if pr > 1 then do
               messg = 'Partition tablespace '||j||' of '||pr
               messg = time() || " " || messg
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
            Call Send_messg
          end
          hurba = word(translate(var.9,' ','-'),7)
          if hurba < trunc(737280/trk,0) then do
             prip=1
             secp=1
          end
          else do
             prip=trunc((hurba/(737280/trk)+1),0)
             secp=max(trunc(prip*0.05,0),1)
          end
       end
       pri=pri+prip
       sec=sec+secp
    end
    if util='RECOVER' & rtyp ¬='CUR' then Call Recover_part 0
    if rec_no=0 then nop
    else do
       db=space(db,0)
       ts=space(ts,0)
       tot=tot+pri
       scu=scu+1
       if pr=1
       then parti=right('NO',6)
       else parti=right('YES',6)
       detail=right(scu,4)||'  '||left(db,10)||,
              left(ts,12)||right(pri,7)||parti
       address ispexec 'tbadd "alist"'
    end
end
if apart='YES' then do
   do j=1 to pr
      if substr(word(in.j,4),2,1) ='I' then do
       part='.I0001.A'||right(j,3,'0')
       file=catn||'.DSNDBD.'||strip(db)||'.'||strip(ts)||part
       dsn = "('"file"')"
       X=OUTTRAP('var.')
       address tso "listc" entries dsn allocation
       X=OUTTRAP('OFF')
       Call Check_dsn
       if rc=0 then do
          if pr > 1 then do
             messg = 'Partition tablespace '||j||' of '||pr
             messg = time() || " " || messg
             Call Send_messg
          end
          hurba = word(translate(var.9,' ','-'),7)
          if hurba < trunc(737280/trk,0) then do
             pri=1
             sec=1
          end
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
            else do
               pri=trunc((hurba/(737280/trk)+1),0)
               sec=max(trunc(pri*0.05,0),1)
            end
         end
         if util='RECOVER' & rtyp ¬='CUR' then Call Recover_part j
         if rec_no=0 then nop
         else do
            db=space(db,0)
            ts=space(ts,0)
            tot=tot+pri
            scu=scu+1
            pts=j
            parti=right(j,6)
            detail=right(scu,4)||'  '||left(db,10)||,
                   left(ts,12)||right(pri,7)||parti
            address ispexec 'tbadd "alist"'
         end
        end
      end
   end
   messg = substr(asterisks,1,trunc(procent*i,0))
   Call Send_messg
end
if no_rec=0 then do
    message='NO IC'||"'"||'s found for RECOVER or '||,
            'you didn'||"'"||'t select IC.'
    Call Free_proc
    Call Aloc
    address ispexec 'tbend "alist"'
    address ispexec 'tbend "blist"'
    address ispexec  "tbclose "messdb""
    Call Create_messg
    Call Error 'crec'
end
tot=right(tot,7)
cyl=trunc(tot/trk,0)
if tot//trk > 0 then cyl=cyl+1
cyl=right(cyl,7)
address ispexec 'tbtop "alist"';
messg = 'Building a '||util||' job'
messg = time() || " " || messg
Call Send_messg
/* JCL Skeleton DB2 Utility    */
title = util||' UTILITY'
dsuf='D'||date('b')
tsuf='T'||time('s')
date=date()
time=time(c)
user=userid()
tempfile=userid()||'.UTIL.'||util
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
address tso
"delete '"tempfile"'"
"free dsname('"tempfile"')"
"free ddname(ispfile)"
"free attrlist(formfile)"
"attrib formfile blksize(800) lrecl(80) recfm(f b) dsorg(ps)"
"alloc ddname(ispfile) dsname('"tempfile"')",
       "new using (formfile) unit(3390) space(1 1) cylinders"
ctime=(time('s')-ctime)%60 min (time('s')-ctime)//60
if util='RECOVER' | util='REORG' | util='COPY' then do
   tsufA=time('M')+1
   dsufB='D'||right(date('D'),3,'0')||right(time('M'),4,'0')
   dsufA='D'||right(date('D'),3,'0')||right(tsufA,4,'0')
   if vola=' ' then vola=vol
   if volb=' ' then volb=vol
end
address ispexec
"ftopen"
if util='CHECK'   then "ftincl DBCHECK"
if util='COPY'    then "ftincl DBCOPY"
if util='RECOVER' then "ftincl DBRECO"
if util='REORG'   then "ftincl DBREORG"
"ftclose"
zedsmsg = "JCL shown"
zedlmsg = "JCL DB2 '||util||' shown"
"setmsg msg(isrz001)"
"edit dataset('"tempfile"')"
address ispexec 'tbend "alist"'
address ispexec 'tbend "blist"'
address ispexec  "tbclose "messdb""
Exit
Aloc:
  ADDRESS TSO "DELETE '"SYSVAR(SYSUID)".UTIL.DBUT'"
  "ALLOC DD(SYSPRINT) DSN('"SYSVAR(SYSUID)".UTIL.DBUT') SPACE(24 8),
  TRACK MOD UNIT(3390) RECFM(F,B) LRECL(130) BLKSIZE(1300) ,
  F(SYSPRINT) CATALOG REUSE "
Return
Error:
  ARG cur_par
  cur=cur_par
  address ispexec "setmsg msg(dbut001)"
  signal top
Return
Free_proc:
  "execio 0 diskr sysprint (finis"
  address tso "free f(sysprint)"
Return
Check_dsn:
 if rc>0 then do
    message=file||' not found.'
    cur='catn'
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
    address ispexec "setmsg msg(dbut001)"
    Call Free_proc
    Call Aloc
    address ispexec 'tbend "alist"'
    address ispexec 'tbend "blist"'
    address ispexec  "tbclose "messdb""
    Call Create_messg
    signal top
 end
Return
Recover_part:
 ARG j_par
 address ispexec 'tbcreate "rlist" names(icd ict ity idsn rbadsn)'
 rec_no=1
 ic_ind=0
 do r=1 to row.0
    dbts=strip(db)||'.'||strip(ts)
    if j_par¬=0 then dbts=strip(db)||'.'||strip(ts)||' PART '||j_par
    if substr(row.r,2,1)='C' & db=substr(row.r,3,8) &,
       ts=word(row.r,2) & (word(row.r,3)=0 | word(row.r,3)=j_par),
    then do
       icd  = word(row.r,5)
       ict  = word(row.r,6)
       ity  = word(row.r,4)
       idsn = word(row.r,3)
       rbadsn= word(row.r,7)
       ic_ind=1
       address ispexec 'tbadd "rlist"'
    end
 end
 if ic_ind=0 then do
    message='NO IC'||"'"||'s found for RECOVER. '||,
            'Press ENTER to skip this step.'
    address ispexec "setmsg msg(dbut001)"
 end
 address ispexec 'tbtop "rlist"';
 address ispexec 'tbdispl "rlist" panel(dbrlist)'
 if rc=8 then do
    Call Free_proc
    Call Aloc
    address ispexec 'tbend "alist"'
    address ispexec 'tbend "blist"'
    address ispexec  "tbclose "messdb""
    address ispexec 'tbend "rlist"'
    Call Create_messg
    signal top
 end
 if sel ¬='S' then rec_no=0
 else no_rec=1
 sel=' '
 address ispexec 'tbend "rlist"'
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
Return
Tspace_stop:
 if sto='YES' | sto='NO' then nop
 else do
    message='Invalid STOP Tablespace value. '||,
            'Valid values are: YES,NO.'
    Call Error 'sto'
 end
 if que='YES' | que='NO' then nop
 else do
    message='Invalid QUIESCE value. '||,
            'Valid values are: YES,NO.'
    Call Error 'que'
 end
Return
Parts_panel:
 recl=31
 maxrow= length(alldata)/recl
 do ip=1 to maxrow
    in.ip=substr(alldata,1+(ip-1)*recl,recl)
 end
 vrsta=Ø
 n1=1
 n2=maxrow
 dataline=alldata
 parmsg=''
 Tops:
 address ispexec "display panel(dbparts)"
 Call Back
 Call Update
 do while rc=Ø & zcmd¬='G'
    address ispexec 'vget (zverb zscrolla)'
    if zverb='down' then do
       if datatype(zscrolla,'n')=1 then lvline=zscrolla
       if zscrolla='max' | zsclla='m' then lvline=maxrow - lvline
       vrsta=vrsta+lvline
       if vrsta >= maxrow then vrsta=vrsta-lvline
       dataline=substr(alldata,1+recl*vrsta)
    end
    if zverb='up' then do
       if datatype(zscrolla,'n')=1 then lvline=zscrolla
       if zscrolla='min' | zsclla='m' then lvline=256
       vrsta=vrsta-lvline
       if vrsta < 1 then vrsta = Ø
       dataline=substr(alldata,1+recl*vrsta)
    end
    n1=word(dataline,3)
    address ispexec "display panel(dbparts)"
    Call Back
    Call Update
 end
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
 zcmd=''
 do ir=1 to maxrow while (substr(word(in.ir,4),2,1) ¬='I')
 end
 if ir=maxrow+1 then do
    parmsg='All partitions are Excluded.'
    signal tops
 end
 apart='NO'
 do ir=1 to maxrow
    if substr(word(in.ir,4),2,1) ¬='I' then apart='YES'
 end
Return apart
Back:
 if rc=8 then do
    Call Free_proc
    address ispexec 'tbend "blist"'
    Call Aloc
    address ispexec 'tbend messdb'
    Call Create_messg
    signal top
 end
Return
Update:
   ppos=word(dataline,3)
   jj=0
   do jp=ppos to ppos+lvline-1 while jp <= maxrow
      parmsg = ''
      jj=jj+1
      detrow=substr(dataline,1+(jj-1)*recl,recl)
      if substr(word(detrow,4),2,1)='I' |,
         substr(word(detrow,4),2,1)='E' then nop
      else do
         parmsg='Incl/Excl error for partition '||word(detrow,3)||,
                '. Valid is I or E. '
         signal tops
      end
   end
   jj=0
   do jp=ppos to ppos+lvline-1 while jp <= maxrow
      jj=jj+1
      in.jp=substr(dataline,1+(jj-1)*recl,recl)
   end
   alldata=''
   do ia=1 to maxrow
      alldata=alldata||in.ia
   end
Return
Create_messg:
 messg = "S"||userid()
 address ispexec "tbcreate "messdb" names(messg) write replace"
Return
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
 Send_messg:
  address ispexec "tbadd " messdb
  address ispexec "control display lock "
  address ispexec "addpop row(13) column(6)"
  address ispexec "tbdispl "messdb" panel(dbutum)"
  address ispexec rempop
 Return
```

DBUTIL2

```
/* REXX *//* Build a quiesce, report utility  *//* trace r */ ARG util
zpfctl = 'OFF' Y=MSG("OFF")
 /*****************************************************/
 /* Change to your convention standards               */
 program = 'PDBUTIL'
 plan    = 'PDBUTIL'
 llib    = 'SKUPNI.BATCH.LOADLIB'
 /*****************************************************/
 address ispexec 'vput (zpfctl) profile'
 Call Aloc
 head='DB2 '||util||' - Selection Result'
 cur='crec'
 Call Create_messg
 TOP:
 address ispexec "display panel(DB"substr(util,1,3)"M) cursor("CUR")"
 if rc=8 then do
    Call Free_proc
    address ispexec "tbclose "messdb""
    exit
 end
 /* Check input parameters                             */
 if crec=' ' & tabc=' ' & tsnc=' ' & dbnc=' ' then do
    message='At least one Catalog search field must be entered.'
    Call Error 'crec'
 end
 if util = 'REPORT' then do
    if rtype='RECOVERY' | rtype='TABLESPACESET' | rtype='BOTH' then nop
    else do
       message='Invalid TYPE value. '||,
               'Valid values are: RECOVERY, TABLESPACESET or BOTH.'
       Call Error 'rtype'
    end
    if rcu='YES' | rcu='NO' then nop
    else do
       message='Invalid CURRENT value. Valid values are: YES or NO.'
       Call Error 'rcu'
    end
    if rsu='YES' | rsu='NO' then nop
    else do
       message='Invalid SUMMARY value. Valid values are: YES or NO.'
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
      Call Error 'rsu'
   end
end
if util = 'RUNSTATS' then do
   if ref='REFERENCE' | ref='CHANGE' then nop
   else do
      message='Invalid SHRLEVEL value. '||,
              'Valid values are: REFERENCE or CHANGE.'
      Call Error 'ref'
   end
   if rre='YES' | rre='NO' then nop
   else do
      message='Invalid REPORT value. Valid values are: YES or NO.'
      Call Error 'rre'
   end
   if upd='ALL' | upd='ACCESSPATH' | upd='SPACE' | upd='NONE' then nop
   else do
      message='Invalid UPDATE value. '||,
              'Valid values are: ALL, ACCESSPATH, SPACE or NONE.'
      Call Error 'upd'
   end
end
parm=substr(crec,1,8)||substr(tabc,1,18)||substr(tsnc,1,8)||,
 substr(dbnc,1,8)||'   YES          '
messg = "Accessing  db2 system "db2""
messg = time() || " " || messg
Call Send_messg
messg = 'Select     systables      information'
messg = time() || " " || messg
Call Send_messg
ADDRESS TSO
QUEUE "RUN PROGRAM("program") PLAN("plan"),
       LIBRARY ('"llib"'),
       PARMS ('/"parm"')"
QUEUE "END "
"DSN SYSTEM("db2")"
if rc=12 then do
   "delstack"
   Call Free_proc
   Call Aloc
   address ispexec 'tbend messdb'
   Call Create_messg
   message = 'Error.   'db2||' ssid is not valid  |'
   Call Error 'db2'
end
"EXECIO * DISKR SYSPRINT (STEM ROW."
if substr(row.1,2) = 'NO CATALOG ENTRIES FOUND' then do
   Call Free_proc
   Call Aloc
   address ispexec 'tbend messdb'
   Call Create_messg
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
   message = 'No catalog entries found, check Search Fields.'
   Call Error 'crec'
end
else do
   address ispexec 'addpop row(1) column(5)'
   address ispexec 'tbcreate "blist" names(v1 v2 v3 v4 v5 v6)'
   count=Ø
   num=row.Ø
   do i=1 to row.Ø
      if substr(row.i,2,1)='A' then count=count+1
      if substr(row.i,2,1)='B' then do
         v6=' '
         v1= substr(row.i,3,8)
         v2= word(row.i,2)
         v3= word(row.i,3)
         v4= word(row.i,4)
         v5= right(word(row.i,5),13)
         if v5 = '-1' then v6='Runstat'
         address ispexec 'tbadd "blist"'
      end
   end
   address ispexec 'tbtop "blist"';
   address ispexec 'tbdispl "blist" panel(dbulist)'
   if rc=8 then do
      Call Free_proc
      address ispexec 'tbend "blist"'
      Call Aloc
      address ispexec rempop all
      address ispexec 'tbend messdb'
      Call Create_messg
      signal top
   end
end
Call Free_proc
address ispexec 'tbcreate "alist" names(db ts detail scu)'
scu=Ø
do i=1 to row.Ø while(substr(row.i,2,1)='A')
   db = space(substr(row.i,3,8),Ø)
   ts=  space(word(row.i,2),Ø)
   scu=scu+1
   detail=right(scu,4)||'  '||left(db,1Ø)||left(ts,12)
   address ispexec 'tbadd "alist"'
end
address ispexec 'tbtop "alist"';
messg = 'Building a '||util||' job'
messg = time() || " " || messg
Call Send_messg
/* JCL Skeleton DB2 Utility    */
title = util||' UTILITY'
address ispexec rempop all
dsuf='D'||date('b')
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
tsuf='T'||time('s')
date=date()
time=time(c)
user=userid()
tempfile=userid()||'.UTIL.'||util
address tso
"delete '"tempfile"'"
"free dsname('"tempfile"')"
"free ddname(ispfile)"
"free attrlist(formfile)"
"attrib formfile blksize(800) lrecl(80) recfm(f b) dsorg(ps)"
"alloc ddname(ispfile) dsname('"tempfile"')",
       "new using (formfile) unit(3390) space(1 1) cylinders"
address ispexec
"ftopen"
if util='QUIESCE'  then "ftincl DBQUIE"
if util='REPORT'   then "ftincl DBREPO"
if util='RUNSTATS' then "ftincl DBRUNS"
"ftclose"
zedsmsg = "JCL shown"
zedlmsg = "JCL DB2 '||util||' shown"
"setmsg msg(isrz001)"
"edit dataset('"tempfile"')"
address ispexec 'tbend "alist"'
address ispexec 'tbend "blist"'
address ispexec  "tbclose "messdb""
exit
Aloc:
  ADDRESS TSO "DELETE '"SYSVAR(SYSUID)".UTIL.DBUT'"
  "ALLOC DD(SYSPRINT) DSN('"SYSVAR(SYSUID)".UTIL.DBUT') SPACE(24 8),
  TRACK MOD UNIT(3390) RECFM(F,B) LRECL(80) BLKSIZE(800) ,
  F(SYSPRINT) CATALOG REUSE "
Return
Error:
  ARG cur_par
  cur=cur_par
  address ispexec "setmsg msg(dbut001)"
  signal top
Return
Free_proc:
  "execio 0 diskr sysprint (finis"
  address tso "free f(sysprint)"
Return
Create_messg:
 messg = "s"||userid()
 address ispexec "tbcreate "messdb" names(messg) write replace"
Return
Send_messg:
 address ispexec "tbadd " messdb
 address ispexec "control display lock "
 address ispexec "addpop row(13) column(6)"
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
  address ispexec "tbdispl "messdb" panel(dbutum)"
  address ispexec rempop
 Return
```

DBMOD

```
/* REXX *//* MODIFY: Build a modify recovery job         *//* trace r */
zpfctl = 'OFF' Y=MSG("OFF") /
**************************************************/ /* Change to your
convention standards               */
 program = 'PDBMODI'
 plan    = 'PDBMODI'
 llib    = 'SKUPNI.BATCH.LOADLIB'
 /************************************************/
 address ispexec 'vput (zpfctl) profile'
 Call Aloc
 cur='dbnc'
 Call Create_messg
 TOP:
 address ispexec "display panel(dbmodm) cursor("CUR")"
 if rc=8 then do
    Call Free_proc
    address ispexec "tbclose "messdb""
    exit
 end
 /* Check input parameters                              */
 if tsnc=' ' & dbnc=' ' then do
    message='At least one Catalog search field must be entered.'
    Call Error 'dbnc'
 end
 if (dage=' ' & ddate=' ') | (dage ¬=' ' & ddate ¬=' ') then do
    message='You must enter either the age or date field. '||,
             'You may not enter both.'
    Call Error 'ddate'
 end
 if (ddate=' ' & verify(dage,'Ø123456789') >Ø ) |,
    (ddate=' ' & dage > 32767) then do
    message='The AGE parameter must be an integer, '||,
            'a number between Ø and 32767'
    Call Error 'dage'
 end
 if (dage=' ' & verify(ddate,'Ø123456789') >Ø ) |,
    (dage=' ' & length(ddate) ¬= 8) then do
    message='The DATE parameter must be a number in the form YYYYMMDD.'
```

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

Bernard Zver
Database Administrator
Informatika Maribor (Slovenia)

© Xephon 1998

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

Automated QMF user-id change

BACKGROUND

Occasionally, the need to change QMF user-ids arises. This can be for various reasons, for example:

- Many organizations often need to shuffle employees between departments. Such transfers require the changing of user-ids (usually part or all of the department code/identifier is embedded in the user-id).
- New naming standards can affect existing user-ids.

The QMF user's objects (ie FORM, PROC, QUERY, etc) are linked to the user-id as OWNER or CREATOR. To retain this link, the following simple tool is used at our site to respond to the situations mentioned above.

THE SOLUTION

QMF can be used to change the QMF user-id (while retaining the objects' inter-relationship).

A QMF procedure (P_CHG_QMF_USERID) is run to execute four QMF queries that would update all occurrences of the existing user-id in the appropriate QMF system tables with the new user-id .

The required card to run this procedure under TSO (using IKJEFT01) is also provided.

Note: you should ensure that a back-up of the tablespaces to be updated (DSQTSCT1, DSQTSCT2, DSQTSCT3, and DSQTSPRO) is available .

OWNERID.P_CHG_QMF_USERID

```
–
– THIS PROC CAN BE USED TO AUTOMATE THE CHANGE OF AN EXISTING –
– QMF USER-ID (INPUT AS &OLD) TO A DIFFERENT ONE (INPUT AS &NEW
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
- AND FOR VERIFICATION DISPLAYS THE COUNT OF THE OCCURRENCES BEFORE-
- AND AFTER THE CHANGE (THEY SHOULD BE EQUAL!). THE CHANGE WOULD   -
- AFFECT 4 QMF TABLES. (IF SEVERAL USERS ARE TO BE CHANGED THE
- SAME PROC CAN BE SUBMITTED FROM BATCH.
-
SET GLOBAL (OLD_USR=&OLD NEW_USR=&NEW
-----------------------------OBJECT DATA
RUN QUERY Q_CNT_OBJ_DATA (&&OWNER=&OLD
PRINT REPORT
RUN QUERY Q_U_OBJ_DATA(&&OLD=&OLD &&NEW=&NEW
RUN QUERY Q_CNT_OBJ_DATA (&&OWNER=&NEW
PRINT REPORT
-----------------------------OBJECT DIRECTORY
RUN QUERY Q_CNT_OBJ_DIR(&&OWNER=&OLD
PRINT REPORT
RUN QUERY Q_U_OBJ_DIRECTORY (&&OLD=&OLD &&NEW=&NEW
RUN QUERY Q_CNT_OBJ_DIR(&&OWNER=&NEW
PRINT REPORT
-----------------------------OBJECT REMARKS
RUN QUERY Q_CNT_OBJ_REMARKS(&&OWNER=&OLD
PRINT REPORT
RUN QUERY Q_U_OBJ_REMARKS (&&OLD=&OLD &&NEW=&NEW
RUN QUERY Q_CNT_OBJ_REMARKS(&&OWNER=&NEW
PRINT REPORT
-----------------------------OBJECT PROFILES
RUN QUERY Q_CNT_PROFILES(&&OWNER=&OLD
PRINT REPORT
RUN QUERY Q_U_PROFILES (&&OLD=&OLD &&NEW=&NEW
RUN QUERY Q_CNT_PROFILES(&&OWNER=&NEW
PRINT REPORT
-----------------------------
```

OWNERID.Q_CNT_OBJ_DATA

```
SELECT COUNT(*) FROM Q.OBJECT_DATA
WHERE OWNER = &OWNER
```

OWNERID.Q_CNT_OBJ_DIR

```
SELECT COUNT(*) FROM Q.OBJECT_DIRECTORY
WHERE OWNER = &OWNER
```

OWNERID.Q_CNT_OBJ_REMARKS

```
SELECT COUNT(*) FROM Q.OBJECT_REMARKS
WHERE OWNER = &OWNER
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

OWNERID.Q_CNT_PROFILES

```
SELECT COUNT(*) FROM Q.PROFILES
WHERE CREATOR = &OWNER
```

OWNERID.Q_U_OBJ_DATA

```
UPDATE Q.OBJECT_DATA
SET OWNER = &NEW
WHERE OWNER = &OLD
```

OWNERID.Q_U_OBJ_DIRECTORY

```
UPDATE Q.OBJECT_DIRECTORY
SET OWNER = &NEW
WHERE OWNER = &OLD
```

OWNERID.Q_U_OBJ_REMARKS

```
UPDATE Q.OBJECT_REMARKS
SET OWNER = &NEW
WHERE OWNER = &OLD
```

OWNERID.Q_U_PROFILES

```
UPDATE Q.PROFILES
SET CREATOR = &NEW
WHERE CREATOR = &OLD
```

JCL

```
//SYSTSIN  DD  *
ISPSTART PGM(DSQQMFE) PARM(M=B,S=DSN,I=P_CHG_QMF_USERID( -
&&OLD='USR001', &&NEW='USR002'))
```

Othman M Kidwai
DBA
RSADF/Orlecon (Saudi Arabia)

© Xephon 1998

DB2 Update is looking for REXX EXECs, macros, program code, etc, that experienced DB2 users have written to make their life easier. Articles can be sent or e-mailed to Robert Burgess at any of the addresses shown on page 2.

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

DB2 news

IBM has announced Version 5.2 of its DB2 Universal Database, which is targeted at more complex data warehousing, data mining, and OLAP. New features and functions included in Version 5.2 enable the administrator to create bigger tables with more columns, speed up DB2 data loading, create summary tables, and replicate them across multiple nodes. Administators can also remotely manage DB2 databases via a Web control centre, use SQLJ to embed static SQL in Java programs, use SCO UnixWare 7 and HP-UX 11 with DB2 servers and clients, and manage external files using DB2 Data Links Manager.

For further information contact your local IBM representative.

* * *

Starquest Software has announced StarSQL 2.6, for connecting Windows-based applications with DB2 databases. StarSQL provides transfer of data on mainframe, mid-range, or Unix systems into Open Database Connectivity-enabled PC applications through either a TCP/IP or SNA network. The software also has the ability to change mainframe or AS/400 host passwords from a PC.

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

Princeton Softech has announced Version 3.0 of The Relational Tools, a suite of DB2 extraction, comparison, and verification tools. It features object-level semantic ageing, legacy date interpretation, and the new Semantic SafeGuard, which, respectively, overcome single table ageing restrictions, prevent legacy date omission errors, and eliminate date warping errors.

Object-level semantic ageing ages dates in interrelated tables via a single pass of the source database, maintaining the referential integrity of the data. Support for built-in semantic date ageing using the user's own business rules eliminates the need to code exits or create holiday tables by hand.

The Legacy Date Interpretation feature recognizes legacy dates that weren't converted into DB2 date data types, but were instead defined to DB2 as general alphanumeric fields. As a result, every date can be aged.

Version 3.0 shares business rules and date semantics defined to Princeton's Ager 2000, providing a single store for date-oriented semantics for Y2K test data generation.

For further information contact:
Princeton Softech, 1060 State Road, Princeton, NJ 08540-1423, USA.
Tel:(609) 497 0205.
URL: http://www.princetonsoftech.com.

* * *