90

DB2 update

April 2000

In this issue

DB2 Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38342
From USA: 01144 1635 38342
E-mail: trevore@xephon.com

North American office

Xephon
PO Box 350100
Westminster, CO 80035-0100
USA
Telephone: 303 410 9344

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

Trevor Eddolls

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 1997 issue, are available separately to subscribers for £22.50 ($33.50) each including postage.

Printed in England.

Viewing DB2 dataset information for a database using the LISTCAT command

The TSO LISTCAT command can be used to view information pertaining to DB2 datasets. Unfortunately, the information is not comprehensible, can be seen for only one dataset at a time, and does need some interpretation to be useful.

There have been articles in *DB2 Update* that translate the information, one dataset at a time. This utility writes the output for a complete database into a flat file and also summarizes space usage. It can also handle multi-volume datasets.

Information may be captured at the database level for any given database or for all the databases with a particular VCAT name. DB2 datasets are generally named as VCATNAME.DSNDBD. DBNAMEXX.TSNAMEXX.I001.A00n. At our installation, the VCATNAME is split into a 4-character high-level qualifier (HLQ) and a 4-character DB2 sub-system identifier. (You can modify the utility to accept an 8-character VCAT name or split it into two 4-character strings and provide the VCAT HLQ and DB2 SSID inputs.)

Values input to the utility are VCAT name (the 4-character HLQ), SSID, database name, output dataset name (default generated and newly allocated), and a percentage increase over existing values. The database name is an optional input value. If provided, the output is generated only for that database – otherwise all databases with that VCAT HLQ and SSID combination are represented. The percentage increase will be used to calculate new PRIMARY quantities, which will be used by another utility for generating ALTER DDLs.

Output from the utility are two sequential datasets:

1 OUTPUT dataset containing information – see Figure 1.

2 SUMMARY dataset containing volume usage summary – see Figure 2.

The output dataset name has the format PREFIX.USERID.OUTPUT.*, or the user can give it a name.

```
  DBNAME   OBJECT PART VOLSER  NUPGS   PQTY   SQTY  EXTS SPCALC  SPCUSE  %USE   NPQTY    NSQTY  N%use PART   OBNAME
-----------------------------------------------------------------------------------------------------------------------
DBLSTCAT TSAQRFPD  ØØ1 VOLØØ1    18Ø    72Ø    72Ø     1    72Ø     72Ø 1ØØ.ØØ   144Ø  DEFAULT  5Ø.ØØ  ØØ1  TSAQRFPD TN
DBLSTCAT IXAQRFA1  ØØ1 VOLØØ2    18Ø    72Ø    72Ø     1    72Ø     72Ø 1ØØ.ØØ   144Ø  DEFAULT  5Ø.ØØ  ØØ1  IXAQRFA1 IN
DBLSTCAT TSAREPPD  ØØ1 VOLØØ1    18Ø    72Ø    72Ø     1    72Ø     72Ø 1ØØ.ØØ   144Ø  DEFAULT  5Ø.ØØ  ØØ1  TSAREPPD TN
DBLSTCAT IXAREPA1  ØØ1 VOLØØ2    18Ø    72Ø    72Ø     1    72Ø     72Ø 1ØØ.ØØ   144Ø  DEFAULT  5Ø.ØØ  ØØ1  IXAREPA1 IN
DBLSTCAT TSASTSPD  ØØ1 VOLØØ1    36Ø  1296Ø   144Ø     1  1296Ø    144Ø  11.11   216Ø  DEFAULT  66.66  ØØ1  TSASTSPD TN
DBLSTCAT IXASTSA1  ØØ1 VOLØØ2    36Ø  3Ø24Ø   36ØØ     1  3Ø24Ø    144Ø   4.76   216Ø  DEFAULT  66.66  ØØ1  IXASTSA1 IN
DBLSTCAT IXASTSA2  ØØ1 VOLØØ2    36Ø  3Ø24Ø   36ØØ     1  3Ø24Ø    144Ø   4.76   216Ø  DEFAULT  66.66  ØØ1  IXASTSA2 IN
DBLSTCAT TSAUIDPD  ØØ1 VOLØØ1    36Ø  1296Ø   144Ø     1  1296Ø    144Ø  11.11   216Ø  DEFAULT  66.66  ØØ1  TSAUIDPD TN
DBLSTCAT IXAUIDA1  ØØ1 VOLØØ2    36Ø  2Ø16Ø   216Ø     1  2Ø16Ø    144Ø   7.14   216Ø  DEFAULT  66.66  ØØ1
```

Figure 1: Example output dataset

```
VOLUME NAME   SPC ALLOC    SPC-U CYL
--------------------------------------------------------
VOLØØ1           17856Ø   28.3666667
VOLØØ2           22968Ø           38
--------------------------------------------------------
Total space used = 47784
Total space allc = 4Ø824Ø
Total cyls. used = 66.3666667
Total cyls. allc = 567
Total rounded cyls. used = 67
Total rounded cyls. allc = 567
```

Figure 2: Dataset containing volume usage summary

The summary dataset name has the format PREFIX.USERID.SMRY.*

The headings in Figure 1 denote the following:

- DBNAME – database name.
- OBJECT – tablespace name.
- PART – partition number.
- VOLSER – volume name (the utility can handle datasets residing on multiple volumes).
- NUPGS – number of used pages.
- PQTY – primary quantity allocated in KB (this information is stored in DB2 catalog tables in terms of pages).
- SQTY – secondary quantity allocation in KB.
- EXTS – number of extents.
- SPCALC – total space allocated.
- SPCUSE – space used.
- %USE – percentage of space used (100 * SPCUSE / SPCALC).

- NPQTY – new primary quantity (rounded to the next higher cylinder boundary).
- NSQTY – new secondary quantity (always written as DEFAULT).
- N%use – new percentage used (100 * SPCUSE / NPQTY).

The last two characters on each line indicate whether the dataset is a tablespace (T) or index space (I), and whether it is partitioned (P) or non-partitioned (N). This information is used internally for ease of coding and for generating the JCL for other utilities, and is derived from a knowledge of the naming scheme used in the application. The code should be modified appropriately as indicated for application-specific information for identifying tablespaces or indexspaces. If no standards exist, then this will have to be skipped altogether. For example, all indexspace names could begin with the letters I or X, and the tablespace names could begin with the letter T. Alternatively, the seventh letter in the name could be an X or I to indicate index spaces, and tablespaces are indicated by the absence of those letters.

The NPQTY is calculated based on an input received at the time of executing the utility. The NSQTY is coded as DEFAULT always. We have used this report as the input for another utility that generates ALTER statements and also verifies space on the volume.

The utility also produces a summary dataset, which lists the volumes used and the spaces allocated and utilized.

The summary dataset also provides precise information on the total amount of space allocated and used by a particular database or all the databases.

This utility is tremendously useful where there is a consistent and organized naming scheme as in our installation. The output of this utility has been used as the basis for generating ALTERs and associated JCL for image copies, REORGs, and RUNSTATS. At our installation there are several hundred tables and we have periodic bursts of data across several tables in a database. We use this utility to generate and analyse the space information. We also edit the values under the NPQTY and SPQTY columns to appropriate values as we desire. We then use this file as an input to another utility which performs two activities:

- Generates the ALTER commands for the rows where the NPQTY is different from the PQTY.
- Generates a file summarizing total space requested for each volume and total space available in each volume. Such a utility may be presented in a future article.

Error checking logic and user interface have been kept to a minimum. Typically, arithmetic errors might occur if incorrect VCAT names are provided.

LSTCAT

```
/* rexx */
/*trace r     on if errors */
/*trace i     on always    */
/*trace o    off always    */
/*                                                                 */
/*                                                                 */
trace o
clear
gralc = 0
gruse = 0
usecyl = 0
alccyl = 0
rusecyl = 0
ralccyl = 0
PREFX = SYSVAR(SYSPREF)
/*                                                                 */
/*   The 4 char VCAT HLQ and the subsystem ID together must form   */
/*   the full first-level node of the DB2 datasets.               */
/*                                                                 */
say
say 'Please input the 4-char VCAT HLQ  ...'
parse upper pull I_HLQ
say 'Enter the 4-char subsystem ID ...: '
parse upper pull I_sid
say 'Enter a Database name if you want to limit to one '
say ' Or Press Enter for all Databases .....'
parse upper pull I_dbname

I_HLQ = strip(I_HLQ)
I_sid = strip(I_sid)
```

```
I_dbname = strip(I_dbname)
P_CATHLQ = I_HLQ||I_sid

cd = date(U)
us_date = substr(cd,7,2)||substr(cd,1,2)||substr(cd,4,2)
ods_name = PREFX||"."||userid()||".OUTPUT."||P_CATHLQ
dbnode = substr(I_dbname,1,6)
if I_dbname = '' then
   nop
else
   ods_name = PREFX||"."||userid()||".OUTPUT."||P_CATHLQ||"."||dbnode

ods_name = ods_name||".D"||us_date

smry_ds = PREFX||'.'||userid()||'.SMRY.'||P_CATHLQ
if I_dbname = '' then
   nop
else
   smry_ds = PREFX||"."||userid()||".SMRY."||P_CATHLQ||"."||dbnode

smry_ds = smry_ds||".D"||us_date

call GETDSN
call GETDEF
xx=outtrap("zap.","*")
address tso "delete '"ods_name"'"
address tso "delete '"smry_ds"'"
xx=outtrap("OFF")
address tso "alloc f(opds) new unit(hsm) space(1,2)",
            "cyl reuse dsname('"ods_name"')",
            "dsorg(ps) blksize(13300) lrecl(133) recfm(f b)"
address tso "alloc f(smds) new unit(hsm) space(1,2)",
          "cyl release dsname('"smry_ds"')",
          "dsorg(ps) blksize(13300) lrecl(133) recfm(f b)"

If I_dbname = '' then
   P_CATNAM = P_CATHLQ||'.DSNDBD'
else
   P_CATNAM = P_CATHLQ||'.DSNDBD.'||I_dbname

P_CATNAM = strip(P_CATNAM)
x = outtrap("lsout.","*")
"Listcat level('"P_CATNAM"')"
x = outtrap("OFF")
i2=0
do  j=1 to lsout.0 by 2
  strng = strip(lsout.j)
  i2=i2+1
  parse VAR strng w1 w2 P_ddn.i2  w4.
```

```
end

fnd = 0
step1:
k=0
do z = 1 to i2

   if  (z//50) = 0 then
      say 'Processed 'z' members so far ...'

   DDN = strip(P_ddn.z)
   ADDRESS TSO
   parse var DDN a1 '.' a2 '.' O_dbname '.' O_obname '.' a3 '.' pno
   O_obname = strip(O_obname)
   O_dbname = strip(O_dbname)
   pno      = strip(pno)
   pno = substr(pno,2)

   x = outtrap("lcout.","*")
   "Listcat entries('"DDN"') all"
   x = outtrap("OFF")

 /*  get extents information from line 17  */

   strng = strip(lcout.17)
   parse VAR strng w1 w2 w3
   w3 = strip(w3)
   parse  VAR w3 dumy 8 exts
   N_exts = strip(exts,Leading,'-')

 /*  get space type and HI-ARBA  from line 22  */

   strng = strip(lcout.22)
   parse VAR strng w1 w2
   w1=strip(w1)
   w2=strip(w2)
   parse var w1  dumy 11 spctyp
   parse var w2  dumy 10 hi_arba
   spctyp = strip(spctyp,Leading,'-')
   hi_arba = strip(hi_arba,Leading,'-')

 /*  get PRIQTY and HI-URBA  from line 23  */

   strng = strip(lcout.23)
   parse VAR strng w1 w2
   w1=strip(w1)
   w2=strip(w2)
   parse var w1  dumy 10 priqty
   parse var w2  dumy 9 hi_urba
   priqty = strip(priqty,Leading,'-')
   hi_urba = strip(hi_urba,Leading,'-')

 /*  get SECQTY              from line 24  */

   strng = strip(lcout.24)
   parse VAR strng w1
   w1=strip(w1)
   parse var w1 dumy 10 secqty
   secqty = strip(secqty,Leading,'-')

 /*  get          RECSIZE    from line 26  */
 /*                                        */

   strng = strip(lcout.26)
   parse VAR strng w1 w2 w3
   w1=strip(w1)
   w2=strip(w2)
   parse var w2 dumy 12 recsize
   recsize = strip(recsize,Leading,'-')

 /*  get vol names and tracks in each vol   */
   h = 0
   do v = 26 to lcout.0
      strng = strip(lcout.v)
      parse VAR strng w1 rest
      if substr(w1,1,6) = 'VOLSER' then
      do
         parse var w1 dumy 7 mvols
         mvols = strip(mvols,Leading,'-')
         h = h+1
         mulvols.h = mvols
         trkrow = v+4
         trkstr = strip(lcout.trkrow)
         parse var trkstr w1 w2 w3
         w3=strip(w3)
         parse var w3 dumy 7 mtrks
         mtrks = strip(mtrks,Leading,'-')
         multrks.h = strip(mtrks)
      end
   end

/*   This line is left here for debugging  purposes */
/*say 'exts, spctyp, hi_arba, priqty' N_exts spctyp hi_arba priqty */
/*say 'secqty hi_urba volser recsize' secqty hi_urba volser recsize*/

   napgs = hi_arba / recsize
   nupgs = hi_urba / recsize
   spcuse = hi_urba/1024
   spcalc = hi_arba/1024
   cylalc = trunc(((spcuse*definc)+719)/720) * 720
   peruse = trunc((spcuse/spcalc*100),2)
   if cylalc <> 0 then
      newuse = trunc((spcuse/cylalc*100),2)
   else
      newuse = 99

   if spctyp = 'CYLINDER' then
   do
      priqty = priqty*15
      secqty = secqty*15
      spctyp = 'CYL'
   end

   if spctyp = 'TRACKS' then
   do
      spctyp = 'TRK'
   end
/*   this number 49152 is based on 3390 mod-3 packs      */
/*   and it is the number of bytes per cylinder          */
 priqty = trunc(priqty*49152/1024)
 secqty = trunc(secqty*49152/1024)

   do while length(O_obname) < 8
      O_obname = ' '||O_obname
   end

   do while length(O_dbname) < 8
      O_dbname = ' '||O_dbname
   end

   do while length(priqty) < 10
      priqty = ' '||priqty
   end

   do while length(secqty) < 8
      secqty = ' '||secqty
   end

   do while length(N_exts) < 3
      N_exts = ' '||N_exts
   end

   do while length(nupgs) < 8
      nupgs = ' '||>upgs
   end

   do while length(spcuse) < 8
      spcuse = ' '||spcuse
   end
```

```
do while length(spcalc) < 8
   spcalc = ' '||spcalc
end

do while length(cylalc) < 8
   cylalc = ' '||cylalc
end

do while length(peruse) < 6
   peruse = ' '||peruse
end

do while length(newuse) < 6
   newuse = ' '||newuse
end

if  h > 1 then
   volser = '******'

if  h = 1 then
do
    volser = mulvols.1
    volname = volser
    Call SUMVOLS
end

if substr(O_obname,7,1) = 'X' then
   ixmrkr = 'I'
else
   ixmrkr = 'T'

if pno > 1 then
do
   oldstr = out.k
   out.k = overlay('P',oldstr,133)
   prtind = 'P'
end
else
  prtind = 'N'

k = k+1
out.k = O_dbname||' '||O_obname||'  '||pno||' '||volser
out.k = out.k||' '||>upgs||' '||priqty||' '||secqty||'   '||N_exts
out.k = out.k||' '||spcalc|| ' '||spcuse ||' 'peruse
out.k = out.k||' '||cylalc||'  DEFAULT  '||>ewuse
out.k = out.k||'  '||pno||'  'O_obname' '||ixmrkr||prtind

if h>1 then
do
```

```
     do j = 1 to h
        k=k+1
        multrks.j = trunc(multrks.j * 49152/1024)
        volname = mulvols.j
        spcalc = multrks.j
        cylalc = 0
        Call SUMVOLS
        do  while length(multrks.j) < 10
            multrks.j = ' '||multrks.j
        end
        out.k = '                       '||mulvols.j
        out.k = out.k||'           '||multrks.j
     end
   end
end

hdr.1 = '  DBNAME   OBJECT PART VOLSER    NUPGS       PQTY     SQTY'
hdr.1 = hdr.1||'  EXTS   SPCALC   SPCUSE   %USE    NPQTY    NSQTY'
hdr.1 = hdr.1||'   N%use PART    OBNAME'
hdr.2 = '-------------------------------------------------------'
hdr.2 = hdr.2||'--------------------------------------------------'
hdr.2 = hdr.2||'-----------------------------'

"execio * diskw opds (stem hdr. "

"execio * diskw opds (FINIS  stem out. "
say 'Output written to 'ods_name

address tso "free ddname(opds)"
Call PRINT_SMRY
say 'Summary written to 'smry_ds
address tso "free ddname(smds)"

exit(0)
/*    End of main routine       */

SUMVOLS:
gruse = gruse + spcuse
gralc = gralc + spcalc
usecyl = usecyl + (spcuse/720)
rusecyl = rusecyl + (trunc((spcuse+719)/720))
alccyl = alccyl + (spcalc/720)
ralccyl = ralccyl + (trunc((spcalc+719)/720))
if fnd = 0 then
do
   fnd=fnd + 1
   vollst.fnd = volname
   voltot.fnd = spcalc
   newtot.fnd = cylalc
   cyltot.fnd = spcuse/720
   return
end
else
do
   fndflg = Ø
   do p = 1 to fnd
      if  vollst.p = volname then
      do
          voltot.p = voltot.p + spcalc
          newtot.p = newtot.p + cylalc
          cyltot.p = cyltot.p + (spcuse/72Ø)
          fndflg = 1
      end
   end
   if  fndflg = Ø then
   do
      fnd=fnd+1
      vollst.fnd = volname
      voltot.fnd = spcalc
      newtot.fnd = cylalc
      cyltot.fnd = (spcuse/72Ø)
      fndflg = 1
   end
end
return

PRINT_SMRY:

do g = 1 to fnd
   voltot.g = strip(voltot.g)
   newtot.g = strip(newtot.g)
   cyltot.g = strip(cyltot.g)
   voluse.g =  trunc(((newtot.g/voltot.g)*1ØØ),2)
   do while length(newtot.g) < 12
      newtot.g = ' '||>ewtot.g
   end
   do while length(voltot.g) < 12
      voltot.g = ' '||voltot.g
   end
   do while length(cyltot.g) < 12
      cyltot.g = ' '||cyltot.g
   end

   smry.g = vollst.g||'     '||voltot.g
   smry.g = smry.g||' '||cyltot.g
end
shdr.1 = 'VOLUME NAME   SPC ALLOC    SPC-U CYL'
shdr.2 = '--------------------------------------------------------'
"execio * diskw smds (stem shdr. "
ftr.1  = '--------------------------------------------------------'
ftr.2  = 'Total space used = 'gruse
ftr.3  = 'Total space allc = 'gralc
ftr.4  = 'Total cyls. used = 'usecyl
ftr.5  = 'Total cyls. allc = 'alccyl
ftr.6  = 'Total rounded cyls. used = 'rusecyl
ftr.7  = 'Total rounded cyls. allc = 'ralccyl
"execio * diskw smds (stem smry. "
"execio * diskw smds (stem ftr. FINIS"
return

GETDSN:
say
say 'Please enter output dataset name or Press Enter to ...'
say '   Use default dataset 'ods_name
say ' *** Note that the output dataset will be deleted if it exists ***'
pull I_dsname
upper I_dsname
I_dsname = strip(I_dsname,Both,"'")
I_dsname= strip(I_dsname)
if I_dsname = '' then
   nop
else
   ods_name = I_dsname
return

GETDEF:
definc = 30
Say  'Give the default percentage increase over the used quantity  '
Say  '  Or Press Enter for default (30) ...'
pull I_definc
upper I_definc
I_definc = strip(I_definc)

if strip(I_definc) = ''  then
   I_definc = definc
else
   definc = I_definc

if strip(I_definc) < 1 | strip(I_definc) > 100 then
do
   say ' ***  Error ***  Percentage must be between 1 and 100 '
   say
   SIGNAL GETDEF
end

definc = 1 + (definc/100)
return
```

Jaiwant K Jonathan
DB2 DBA (USA)

The importance of the MODIFY utility

DB2 for OS/390 comes packaged with its own utilities. One of these is the MODIFY utility, whose main function is to delete outdated information from the DB2 catalog and the DB2 directory. Specifically, tables SYSIBM.SYSCOPY in the DB2 catalog and table SYSIBM.SYSLGRNX in the DB2 directory are purged of obsolete information by the MODIFY utility, based on the parameters specified.

The purpose of this article is to emphasize the importance of the MODIFY utility, as well as to provide two SQL queries to check the status of the DB2 catalog as it relates to the MODIFY utility. Two additional SQL queries are also provided, which generate MODIFY utility statements.

Syntax and additional information on how to use the MODIFY utility can be found in the *DB2 Utility Guide and Reference* manual for the specific version of DB2 at your installation. The syntax for the MODIFY utility for DB2 OS/390 Version 5.1 is as follows:

```
    MODIFY RECOVERY TABLESPACE DBNAME.TSNAME DSNUM PARM1 DELETE PARM2
```

Where:

- DBNAME is an optional name of the database containing the tablespace, the default being database DSNDB04. Note that the database name is followed by a period (full stop).
- TSNAME is the required tablespace name.
- PARM1 is the parameter for the optional keyword DSNUM, which identifies a single partition or a dataset of the tablespace for which records are to be deleted. PARM1 can be either an integer in the range of 1 to 254 or ALL. The default is ALL.
- PARM2 is the parameter for the required keyword DELETE, which indicates the manner in which records will be cleared from SYSCOPY and SYSLGRNX. PARM2 can be AGE + integer, or DATE + integer. AGE deletes all records older than a specified number of days. DATE deletes all records written before a specified date.

In most shops, the database administrators are responsible for implementing the required DB2 utility jobs necessary for the operational support of DB2 databases. Unfortunately, the MODIFY utility is not as glamorous as the IMAGE COPY utility, nor very critical at the time of the database initial implementation. In many cases, the MODIFY utility does not get scheduled to execute at all.

There are two main reasons why the MODIFY utility should be executed on a normal basis: firstly, because of the performance impact, and, secondly, to ensure the DB2 catalog and directory contain valid recovery data.

PERFORMANCE IMPACT

The SYSCOPY table contains information needed by DB2 for recovery purposes. DB2 automatically inserts a row in SYSCOPY every time one of the following utilities is executed: IMAGE COPY (full or incremental), RECOVER (TOCOPY or TORBA), QUIESCE, LOAD (with REPLACE option, LOG YES or LOG NO), and REORG. DB2 will also record whenever the TERM utility command is executed.

This information is stored for every tablespace defined in the DB2 sub-system. In Version 6, DB2 will be able to back-up and recover index spaces as well, so additional information related to index recovery will be stored in the SYSIBM.SYSCOPY table, increasing the importance of the MODIFY utility.

Similarly, the SYSLGRNX (SYSLGNRG for DB2 Version 3.1) table stores recovery log ranges that record the time a tablespace was open for updates. This information provides an efficient way for DB2 to access the appropriate log records necessary for the recovery of any specific tablespace, rather than having to scan every record in the known DB2 recovery logs.

These two tables will accumulate many records for every tablespace in your DB2 system, unless this information is purged on a regular basis. In a medium to large DB2 system, these tables and their corresponding tablespaces can easily grow to several hundred megabytes.

As these two tablespaces grow, any process against them will consume more resources. For example, taking an image copy of these tablespaces will take longer, scanning the tables will consume more CPU and I/O operations, and of course, reorganization of these tables will take longer (availability implications).

There are other less visible implications for large catalog and directory tables, such as the negative impact they may have on the BP0 bufferpool, which is the only bufferpool assigned to all catalog and directory tablespaces. The more pages on the tablespaces and index spaces of the DB2 catalog and directory, the higher the number of pages that would be brought into the BP0 bufferpool, thus reducing its efficiency.

VALID RECOVERY DATA

The second reason for running the MODIFY utility against all tablespaces is to ensure that the DB2 catalog and directory contain valid recovery data. In order to recover any tablespace to the current point in time, DB2 needs the latest full image copy and the DB2 recovery logs that were created since the last full image copy. If incremental image copies are available, DB2 will use a full image copy, the incremental image copies, and the DB2 recovery logs from the time of the last incremental image copy.

Depending on your installation standards, the image copies of your tablespaces will be created on DASD devices and/or tape devices. Most likely, the image copies are set to be stored using generation dataset files (GDGs). Using a GDG allows the same JCL to be re-executed without having to change the name of the image copy dataset. Third-party products that auto-build utility JCL may use either a GDG or a dataset with the date and time embedded in the dataset name.

Because the retention of DB2 image copies and DB2 archive logs is managed by the MVS (OS/390) operating system, it is very possible that the recovery information known to DB2 is no longer in sync with the MVS external environment. GDGs may have rolled-off, datasets may have expired, tapes containing image copies may no longer be available because of damage, DASD datasets may have been deleted because they were not in use during the last 30 days, etc. These and many other external ‘environmental’ reasons will cause the DB2 recovery data to be invalid.

HOW CAN THE DB2 MODIFY UTILITY HELP WITH PERFORMANCE AND RECOVERY DATA PROBLEMS?

The DB2 MODIFY utility cleans old entries from the SYSIBM.SYSCOPY and SYSIBM.SYSLGRN*x* tables. In addition to the clean-up process, the MODIFY utility performs the following actions:

- Recovery information that is maintained in the Data Base Descriptor (DBD) table is updated to reflect the removal of the old SYSLGRNX entries.
- The copy of the Data Base Descriptor (DBD) information that is kept in the Environmental Descriptor Manager (EDM) pool is also updated. Because of this action, heavy-duty MODIFY processes are best performed at times when there is very low system activity.
- If, as a result of the MODIFY utility, all records of DB2 full image copies are removed for a specific tablespace, the status of that tablespace will immediately change to COPY PENDING and no further updates against it will be allowed. Programs with update intention or ambiguous cursors will fail to execute.

SQL QUERIES TO CHECK YOUR ENVIRONMENT

The first SQL query to investigate the state of your SYSIBM.SYSCOPY table is as follows:

```
SELECT DBNAME, TSNAME, COUNT(*)
  FROM SYSIBM.SYSCOPY ONE
 WHERE ONE.ICTYPE = 'F'
   AND DATE(ONE.TIMESTAMP) < (CURRENT DATE - 30 DAYS )
 GROUP BY DBNAME, TSNAME
 ORDER BY DBNAME, TSNAME
 FOR FETCH ONLY;
```

The query above will count the number of full image copies for all tablespaces in your DB2 sub-system that are older than thirty days, and will list the results ordered by database and tablespace in ascending order. This query is very useful in order to get a quick overview of the state of the SYSCOPY table.

A second query, specifically targeted against database XYZ, is as follows:

```
SELECT ONE.DBNAME, ONE.TSNAME, MAX(DATE(ONE.TIMESTAMP))
  FROM SYSIBM.SYSCOPY ONE
 WHERE ONE.DBNAME = 'XYZ'
      AND ONE.ICTYPE = 'F'
GROUP BY ONE.DBNAME, ONE.TSNAME
HAVING 0 =
       ( SELECT COUNT(*)
           FROM SYSIBM.SYSCOPY TWO
          WHERE TWO.ICTYPE = 'F'
            AND TWO.DBNAME = ONE.DBNAME
            AND TWO.TSNAME = ONE.TSNAME
            AND DATE(TWO.TIMESTAMP) >= ( CURRENT DATE - 30 DAYS )
        )
 ORDER BY DBNAME, TSNAME
 FOR FETCH ONLY;
```

The query above will return a list of tablespaces and dates for database XYZ for those tablespaces that have not had a full image copy taken during the last thirty days. This query could expose a potential tablespace recovery problem.

This query could be modified to match your installation's 'environmental' standards, but make sure that you always use a DBNAME = 'XYZ' clause in the select statement.

SQL QUERIES TO GENERATE MODIFY UTILITY STATEMENTS

A simple SQL query to generate MODIFY utility statements is as follows:

```
SELECT DISTINCT 'MODIFY RECOVERY TABLESPACE '
       || STRIP(DBNAME)
       || '.'
       || STRIP(TSNAME)
       || ' DELETE AGE(30)'
 FROM SYSIBM.SYSCOPY
 WHERE  DBNAME = 'XYZ'
```

```
   AND ICTYPE = 'F'
   AND DATE(TIMESTAMP) < ( CURRENT DATE - 30)
GROUP BY DBNAME, TSNAME
 ORDER BY 1
 FOR FETCH ONLY;
```

The SQL statement above will generate a MODIFY utility statement to delete recovery entries older than thirty days for each tablespace of database XYZ that has an image copy older than thirty days. The drawback with this statement is that it does not check for the possibility that there may be tablespaces which do not have a full image copy created in the last thirty days. If these MODIFY utility statements were to be processed, those tablespaces would go into COPY PENDING status.

A more complex SQL statement to generate MODIFY utility statements is as follows:

```
SELECT DISTINCT 'MODIFY RECOVERY TABLESPACE '
       || STRIP(ONE.DBNAME)
       || '.'
       || STRIP(ONE.TSNAME)
       || ' DELETE DATE('
       || MAX(ONE.ICDATE)
       || ')'
  FROM SYSIBM.SYSCOPY ONE
 WHERE ONE.DBNAME = 'XYZ'
   AND ONE.ICTYPE = 'F'
   AND DATE(ONE.TIMESTAMP) < ( CURRENT DATE - 30 DAYS)
   AND ONE.START_RBA <
   ( SELECT MAX(TWO.START_RBA)
       FROM SYSIBM.SYSCOPY TWO
      WHERE TWO.DBNAME = 'XYZ'
        AND TWO.ICTYPE = 'F'
        AND TWO.DBNAME = ONE.DBNAME
        AND TWO.TSNAME  = ONE.TSNAME
        AND DATE(TWO.TIMESTAMP) < ( CURRENT DATE - 30 DAYS)
   )
 GROUP BY ONE.DBNAME, ONE.TSNAME
 ORDER BY 1
 FOR FETCH ONLY;
```

The SQL query above generates the MODIFY statements taking into consideration the contents of the SYSIBM.SYSCOPY table. The sub-query allows the bypassing of those tablespaces that do not have any full image copies created during the last thirty days, thus avoiding the possible COPY PENDING situation.

CLOSING

The MODIFY utility exists for important reasons well described in the DB2 utilities reference guide. This article presents additional opinions complementing the reasons expressed by IBM in its manuals. I hope that after reading this article, you will spend some time reviewing the status of your SYSCOPY table, and if need be, clean it up and reorganize it with the assistance of your DB2 system programmer. Good luck!

Antonio Luis Salcedo Freidel
Lead DB2 System Programmer (USA)

Checking SYSIBM.SYSCOPY for a second dataset with the same VOLSER/FILESEQNO

We use:

```
COPY TABLESPACE FULL YES SHRLEVEL REFERENCE COPYDDN(DØØØØØ1A)
```

to back-up our DB2 databases.

COPYDDN(D000001A) is not catalogued and can extend onto two VOLSERs from the JCL:

```
//DØØØØØ1A DD DISP=(NEW,keep),LABEL=(3454,SL)
```

It is therefore extremely important that VOLSER and FILESEQNO are consistent in SYSIBM.SYSCOPY.

Below is a job which checks that no two COPYDDNs share the same VOLSER and FILESEQNO.

HVJDBCHK

```
//TSHVRB JOB (),'HVJDBCHK',CLASS=A,MSGCLASS=X,NOTIFY=TSHVR
//*CHECK DB2 SYSIBM.SYSCOPY FOR NO 2 DS WITH SAME VOLSER/FILESEQNO
//HVJDBCHK EXEC PGM=IKJEFTØ1
//*STEP1    EXEC SQLDNBT
//*SYSIN    DD  *
//SYSTSPRT DD  SYSOUT=*
//*HVNTEP2=DSNTEP2
//SYSTSIN  DD  *
```

```
 DSN SYSTEM(DSNT)
 RUN  PROGRAM(HVNTEP2) PLAN(HVNTEP2) -
 LIB('PRJSGP.DB2.LOAD')
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 SELECT MAX(X.TSNAME),MIN(X.TSNAME),X.FILESEQNO,X.VOLSER FROM
 (
  SELECT DBNAME,TSNAME,FILESEQNO,
         SUBSTR(DSVOLSER,1,13) AS VOLSER
  FROM SYSIBM.SYSCOPY
  WHERE DSNUM=Ø AND ICTYPE='F' AND ICUNIT='T'
 ) AS X
 GROUP BY X.FILESEQNO,X.VOLSER
 HAVING COUNT(*)>1;
--
 SELECT DBNAME,TSNAME,TIMESTAMP,FILESEQNO,
        SUBSTR(DSVOLSER,1,13) AS VOLSER
 FROM SYSIBM.SYSCOPY
 WHERE FILESEQNO IN
 (
  SELECT FILESEQNO
   FROM SYSIBM.SYSCOPY
    WHERE DSNUM=Ø AND ICTYPE='F' AND ICUNIT='T'
  GROUP BY FILESEQNO
  HAVING COUNT(*)>1
 )
 ;
/*
//
```

Herman Vierendeels
Systems Programmer (Belgium)

Spring-cleaning your DB2 catalog

As your DB2 installation grows older and older, your DB2 catalog becomes more and more dusty in the ‘corners’ and unused objects start to pile up in them!

Access path and index usage change as DB2 evolves and new program releases are installed in production.

In this article we will explore a few simple queries and produce a report to pinpoint the indexes (and views) that are candidates for removal or ‘dusting’ (ie investigate which ones are used or unused).

IDENTIFYING POTENTIAL UNUSED INDEXES

Unused indexes are only candidates for removal or modification – each index still needs to be evaluated, for example by asking why it was created.

Now, although an index is unused, it might still be valid because:

- It might support referential integrity
- It might be necessary for uniqueness
- It might support an access path used in *ad hoc* queries.

The query below filters out indexes covered by items 1 and 2 above.

DB2 favours unique indexes, when choosing an access path, so it is worth investigating why these unique indexes are left ‘unused’.

You have no chance of spotting indexes used by *ad hoc* queries via the catalog. You might be able to do some clever correlation with your statistic data if you have tools and traces for this. If you do have a subsystem with a mixture of dynamic and static SQL, you could filter out indexes defined on tables used for *ad hoc* queries in the queries below.

And now the queries.

IX-QUERY 1

The first query will extract all unused non-unique indexes. These indexes are those most easily dealt with. Unless you have *ad hoc* SQL in your DB2 environment, they will all be candidates for removal!

```
--
-- QUERY TO EXTRACT UNUSED INDEXES
--
SELECT CREATOR,NAME
FROM SYSIBM.SYSINDEXES T1
     WHERE
--
-- YOUR CREATOR GOES HERE
--
             CREATOR IN ('PROD','PILOT')
```

```
--
-- APPLY ANY NAMING STANDARD HERE
-- (TO AVOID INCLUDING 3-PART VENDOR VIEWS )
--
     AND NAME LIKE '_______I'
--
-- DON'T INCLUDE INDEXES DEFINED AS UNIQUE (WE WILL LOOK AT THEM LATER)
--
     AND UNIQUERULE <> 'D'

-- EXTRACT REMAINING NON-USED INDEXES THAT ARE NOT USED BY
-- ANY PLAN OR PACKAGE
--
    AND NOT EXISTS
        (
          SELECT BNAME
             FROM SYSIBM.SYSPLANDEP
             WHERE BNAME = T1.NAME
             AND   BCREATOR = T1.CREATOR
             AND BTYPE = 'I'
        )

     AND NOT EXISTS
        (
          SELECT BNAME
             FROM SYSIBM.SYSPACKDEP
             WHERE BNAME = T1.NAME
             AND   BQUALIFIER = T1.CREATOR
             AND BTYPE = 'I'
        )
--
-- DO NOT INCLUDE RI-INDEXES
--
     AND NOT EXISTS
(SELECT 'DUMMY' FROM SYSIBM.SYSINDEXES T2,SYSIBM.SYSFOREIGNKEYS T3,
SYSIBM.SYSKEYS T4
WHERE
      T1.CREATOR = T4.IXCREATOR
AND   T1.NAME  = T4.IXNAME
AND   T4.IXCREATOR = T2.CREATOR
AND   T4.IXNAME    = T2.NAME
AND   T2.TBCREATOR = T3.CREATOR
AND   T2.TBNAME    = T3.TBNAME
AND   T4.COLSEQ  = T3.COLSEQ
AND   T4.COLNAME = T3.COLNAME
)
     ORDER BY 1,2
     ;
```

IX-QUERY 2

This query will extract all unused unique indexes. These are a bit harder to deal with because you will have to investigate each table and corresponding indexes and evaluate whether the listed indexes are eligible for removal or not.

```
-- UNUSED UNIQUE INDEXES
-- THESE INDEXES MIGHT BE OK, THOUGH SUSPICIOUS AS THEY ARE UNUSED!
--
SELECT CREATOR, NAME
FROM SYSIBM.SYSINDEXES T1
     WHERE
--
-- YOUR CREATOR GOES HERE
--
             CREATOR IN ('PROD','PILOT')
--
-- APPLY ANY NAMING STANDARD HERE
-- (TO AVOID INCLUDING 3-PART VENDOR VIEWS )
--
     AND NAME LIKE '_______I'
     AND UNIQUERULE <> 'D'
     AND NOT EXISTS
        (
          SELECT BNAME
             FROM SYSIBM.SYSPLANDEP
             WHERE BNAME = T1.NAME
             AND   BCREATOR = T1.CREATOR
             AND BTYPE = 'I'
        )
     AND NOT EXISTS
        (
          SELECT BNAME
             FROM SYSIBM.SYSPACKDEP
             WHERE BNAME = T1.NAME
             AND   BQUALIFIER = T1.CREATOR
             AND BTYPE = 'I'
        )
     ORDER BY 1,2
     ;
```

RE-EVALUATING INDEX DESIGN

When you are spring-cleaning your indexes, it will be handy to have a crosstab report for each table like the one listed below.

The X-axis lists the column names for the table, the Y-axis contains primary-key, constraints, and indexes defined on the table.

In the grid, the column order for each column participating in primary-key, constraints, and indexes is listed. Thus it is possible with one glance to see any missing or redundant indexes:

```
--------- TABLENAME ----------

               C   C   C   C   C   C

               O   O   O   O   O   O

               L   L   L   L   L   L

               U   U   U   U   U   U

               M   M   M   M   M   M

               N   N   N   N   N   N

               -   -   -   -   -   -
               1   2   3   4   5   6
----------------------------------------------------------
PRIMARY-KEY  ! 1 ! 2 ! 3 ! 4 ! 5 ! 6 !   !   !   !   !   !
----------------------------------------------------------
CONSTRAINT1  ! 1 !   !   !   ! 2 !   !   !   !   !   !   !
----------------------------------------------------------
INDEX1       ! 1 ! 2 ! 3 ! 4 ! 5 ! 6 !   !   !   !   !   !
----------------------------------------------------------
INDEX2       ! 1 ! 4 ! 3 ! 6 ! 2 ! 5 !   !   !   !   !   !
----------------------------------------------------------
INDEX3       ! 1 ! 5 ! 3 ! 2 ! 4 ! 6 !   !   !   !   !   !
----------------------------------------------------------
```

The report above was custom made for the repository used in this example (naming changed for clarity) and you will probably need to produce a matching report for yourself at your site.

Anyway, with this report (or something close to it) you look for redundant indexes where all the columns in one index are included – with the same column order – in another index. (Yes, you will find some, if you are a ‘mature’ DB2 installation.)

Typically you will be able to find indexes matching a constraint and another index matching an (old and now unused) access path.

For multi-column indexes you might consider eliminating one index if only one or two of the last columns differ in sequence.

The same report can be used to look for missing indexes! After using the report you will wonder how you managed to get along without it.

IDENTIFYING POTENTIALLY UNUSED VIEWS

Finding unused views is a trivial task with SQL.

Once again you need to evaluate whether listed views are really unused or might be referenced by dynamic SQL:

```
--
-- UNUSED VIEWS
--
SELECT DISTINCT CREATOR,NAME
FROM SYSIBM.SYSVIEWS T1
     WHERE
--
-- YOUR CREATOR GOES HERE
--
             CREATOR IN ('PROD','PILOT')
--
-- APPLY ANY NAMING STANDARD HERE
-- (TO AVOID INCLUDING 3-PART VENDOR VIEWS )
--
     AND NAME LIKE '_______V'
     AND NOT EXISTS
        (
          SELECT BNAME
             FROM SYSIBM.SYSPLANDEP
             WHERE BNAME = T1.NAME
             AND   BCREATOR = T1.CREATOR
             AND BTYPE = 'V'
        )
     AND NOT EXISTS
        (
          SELECT BNAME
             FROM SYSIBM.SYSPACKDEP
             WHERE BNAME = T1.NAME
             AND   BQUALIFIER = T1.CREATOR
             AND BTYPE = 'V'
        )
     ORDER BY 1,2
     ;
```

Nothing like a good spring-clean... happy ‘dusting’!

Kim Hjortholm
Kommunedata (Denmark)

Analysing the DSNZPARM load module – revisited

DB2 Update Issues 80, 81, 82, in June, July, and August 1999 contained an article entitled *Analysing the DSNZPARM load module*. This contained ZPARMREE ASSEMBLY, a program developed for DB2 for OS/390, which analyses the DSNZPARM load module and recreates the originating Assembler macro parameters. It is designed to analyse the content of DSNZPARM and can be used as input for an assembly.

The author has updated the code, adding all the new parameters for DB2 for OS/390 Versions 5 and 6.

To work properly, the source must be assembled using the same version of DB2 as the DSNZPARM to be analysed.

```
//SYF9ZPR5 JOB (ØØ94Ø,TEST,,,,776Ø),'ZDV 6.11',COND=(Ø,NE),
//         NOTIFY=SYF9,MSGCLASS=V
/*ROUTE  PRINT N2 R1.N99
/*ROUTE XEQ   N2
/*JOBPARM S=ENTW
//J    OUTPUT  CLASS=J,FORMS=2344,FORMDEF=DUPBIN,DEST=N1R99,
//             PAGEDEF=PHØ88,CHARS=GT15,COPIES=1
//* OUTPUT   DUPKAS    DUP DOPPELSEITIGER DRUCK    BIN GELBES PAPIER V6
//*                    SIM EINSEITIGER DRUCK       KAS WEISSES PAPIERV6
//* PAGEDEF       PH11Ø       PHØ72       PHØ88        PHZWEI   -> HOCH
//*               PGE56                                PQZWEI   -> QUER
//ALDLXLX PROC TLIB=TE1,DLIB=D51Ø SYSOUT='*' (,),OUTPUT=*.J'
//*
//*      STEP  SEQUENCE:
//*      A     COMPILE ZPARMREE
//*      L     LINK    ZPARMREE TO &&LOAD
//*      AD    COMPILE DSNZPARM EXAMPLE
//*      LD    LINK    DSNZPARM TO &&LOAD
//*      AX    COMPILE DSNHDECM EXAMPLE
//*      LX    LINK    DSNHDECM TO &&DSNHDECP
//*      X     EXECUTE ZPARMREE
//*
//A      EXEC  PGM=ASMA9Ø,PARM='OBJECT,NODECK,ESD,NORLD,FLAG(SUBSTR)'
//*YSLIB   DD  DISP=SHR,DSN=SYF9.DB2.CNTL
//SYSLIB   DD  DISP=SHR,DSN=DB2S.TLIB.&TLIB..SDSNMACS
//         DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSLIN   DD  DISP=(MOD,PASS),DSN=&&LOADSET,UNIT=SYSDA,
//             SPACE=(8ØØ,(5ØØ,5ØØ)),DCB=(BLKSIZE=8ØØ)
//SYSPRINT DD  SYSOUT=* &SYSOUT
```

```
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//SYSUT2   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//SYSUT3   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//*
//L      EXEC  PGM=IEWL,PARM='LIST,XREF,RENT',COND=(4,LT,A)
//SYSLIN   DD  DISP=(OLD,PASS),DSN=&&LOADSET
//SYSLIB   DD  DISP=SHR,DSN=DB2S.DLIB.&DLIB..ADSNLOAD DIST LIB
//SYSLMOD  DD  DISP=(,PASS),DSN=&&LOAD(ZPARMREE),
//             SPACE=(TRK,(5Ø,5Ø,2)),UNIT=SYSDA
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(1Ø24,(5Ø,5Ø))
//AD     EXEC  PGM=ASMA9Ø,PARM='OBJECT,NODECK,ESD,NORLD,FLAG(SUBSTR)'
//SYSLIB   DD  DISP=SHR,DSN=DB2S.TLIB.&TLIB..SDSNMACS
//         DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSLIN   DD  DISP=(OLD,PASS),DSN=&&LOADSET
//SYSPRINT DD  SYSOUT=* &SYSOUT
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//SYSUT2   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//SYSUT3   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//*
//LD     EXEC  PGM=IEWL,PARM='LIST,XREF,LET,RENT',COND=(4,LT,A)
//SYSLIN   DD  DDNAME=SYSIN
//DSNHDECM DD  DISP=(OLD,PASS),DSN=&&LOADSET
//SYSLMOD  DD  DISP=(,PASS),DSN=&&DSNHDECP,
//             SPACE=(TRK,(5Ø,5Ø,2)),UNIT=SYSDA
//ADSNLOAD DD  DSN=DB2S.TLIB.&TLIB..SDSNEXIT,DISP=SHR
//         DD  DSN=DB2S.DLIB.&DLIB..ADSNLOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(1Ø24,(5Ø,5Ø))
//AX     EXEC  PGM=ASMA9Ø,PARM='OBJECT,NODECK,ESD,NORLD,FLAG(SUBSTR)'
//SYSLIB   DD  DISP=SHR,DSN=DB2S.TLIB.&TLIB..SDSNMACS
//         DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSLIN   DD  DISP=(OLD,PASS),DSN=&&LOADSET
//SYSPRINT DD  SYSOUT=* &SYSOUT
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//SYSUT2   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//SYSUT3   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//*
//LX     EXEC  PGM=IEWL,PARM='LIST,XREF,NCAL,RENT',COND=(4,LT,A)
//SYSLIN   DD  DDNAME=SYSIN
//LOADSET  DD  DISP=(OLD,PASS),DSN=&&LOADSET
//SYSLMOD  DD  DISP=(MOD,PASS),DSN=&&LOAD
//ADSNLOAD DD  DISP=SHR,DSN=DB2S.DLIB.&DLIB..ADSNLOAD DIST LIB
//SDSNLOAD DD  DISP=SHR,DSN=DB2S.TLIB.&TLIB..SDSNLOAD DSNZPARM
//SYSPRINT DD  SYSOUT=*
```

```
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(1Ø24,(5Ø,5Ø))
//X      EXEC  PGM=*.L.SYSLMOD,COND=(4,LT),PARM=DSNZPARM
//DSNZPARM DD  DISP=(OLD,PASS),DSN=&&LOAD
//         DD  DISP=SHR,DSN=DB2S.TLIB.&TLIB..SDSNLOAD DSNZPARM
//DSNHDECP DD  DISP=(OLD,PASS),DSN=&&DSNHDECP
//         DD  DISP=SHR,DSN=DB2S.TLIB.&TLIB..SDSNEXIT DSNHDECP
//ABNLIGNR DD  DUMMY
//SYSPUNCH DD  DISP=SHR,DSN=SYF9.SYSPUNCH SYSOUT=&SYSOUT
//SNAPDUMP DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//AY     EXEC  PGM=ASMA9Ø,PARM='OBJECT,NODECK,ESD,NORLD,FLAG(SUBSTR)'
//SYSLIB   DD  DISP=SHR,DSN=DB2S.TLIB.&TLIB..SDSNMACS
//         DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSLIN   DD  DISP=(OLD,PASS),DSN=&&LOADSET
//SYSPRINT DD  SYSOUT=* &SYSOUT
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//SYSUT2   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//SYSUT3   DD  UNIT=SYSDA,SPACE=(8ØØ,(5ØØ,5ØØ),,,ROUND)
//SYSIN    DD  DISP=SHR,DSN=SYF9.SYSPUNCH                            V6
//*
//LY     EXEC  PGM=IEWL,PARM='LIST,XREF,NCAL,RENT',COND=(4,LT,AY)
//SYSLIN   DD  DDNAME=SYSIN
//LOADSET  DD  DISP=(OLD,DELETE),DSN=&&LOADSET
//SYSLMOD  DD  DISP=(MOD,PASS),DSN=&&LOAD
//ADSNLOAD DD  DISP=SHR,DSN=DB2S.DLIB.&DLIB..ADSNLOAD DIST LIB
//SDSNLOAD DD  DISP=SHR,DSN=DB2S.TLIB.&TLIB..SDSNLOAD DSNZPARM
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(1Ø24,(5Ø,5Ø))
//Y      EXEC  PGM=*.L.SYSLMOD,COND=(4,LT),PARM=DSNZPARN
//DSNZPARM DD  DISP=(OLD,PASS),DSN=&&LOAD
//         DD  DISP=SHR,DSN=DB2S.TLIB.&TLIB..SDSNLOAD DSNZPARM
//DSNHDECP DD  DISP=(OLD,PASS),DSN=&&DSNHDECP
//         DD  DISP=SHR,DSN=DB2S.TLIB.&TLIB..SDSNEXIT DSNHDECP
//ABNLIGNR DD  DUMMY
//SYSPUNCH DD  DISP=SHR,DSN=SYF9.SYSPUNC2 SYSOUT=&SYSOUT
//SNAPDUMP DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//ALDLXLX PEND
//ZPARMREE EXEC ALDLXLX TLIB=TE1,DLIB=D51Ø,SYSOUT='(,),OUTPUT=*.J'
//A.SYSIN  DD  *
ZPARMV6  TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               RECONSTRUCT TSO INSTALL INPUT'
* ZPARMV6 :  LIST DSNZPARM MACRO VALUES
* FUNCTION : THIS PROGRAM READS THE DSNZPARM AND DSNHDECP LOAD MODULE
*            AND GENERATES SOURCE DSNZPARM AND DSNHDECP MACRO SIMILAR
*            TO SAMPLE JOB DSNTIJUZ WHICH ASSEMBLES/LINKS THE ZPARM.
*            THE PROGRMA MUST BE COMPILED WITH THE SAME MACRO LIBRARY
*            VERSION AS THE DSNZPARM TO BE ANALYZED.
*            BEFORE EXECUTING, COMPILE WITH THE VERSION OF THE DSNZPARM
*            USING THE SDSNMAC OF THAT VERSION AS SYSLIB.            V6
* VERSION  : DB2 VERSION 5,6
* JCL      : SAMPLE JCL TO X THIS PROGRAM IS SHOWN BELOW
*    //X       EXEC PGM=ZPARMV6,COND=(4,LT),
*    //        PARM='DSNZPARM'             <- NAME OF ZPARM
*    //STEPLIB  DD DSN=MY.PROGRAM.LOAD,    <- SYSLMOD OF THIS PGM
*    //         DD DISP=SHR
*    //DSNHDECP DD DSN=MY.TSO.DSNLOAD,DISP=SHR <- YOUR SHOP'S DSNHDECP
*    //DSNZPARM DD DSN=MY.TSO.DSNEXIT,DISP=SHR <- YOUR SHOP'S DSNZPARM
*    //SYSPUNCH DD SYSOUT=*
*    //SNAPDUMP DD SYSOUT=*
* PSEUDOCODE:
*    INITIALIZATION
*      - GET ZPARM NAME FROM PARMLIST
*      - OPEN FILES
*      - PRINT HEADER LINES
*    MAINLINE
*      - LOAD DSNZPARM LOAD MODULE
*      - FORMAT DSN6SPRM CONTROL BLOCK
*      - FORMAT DSN6ARVP CONTROL BLOCK
*      - FORMAT DSN6LOGP CONTROL BLOCK
*      - FORMAT DSN6SYSP CONTROL BLOCK
*      - FORMAT DSN6FAC  CONTROL BLOCK
*      - LOAD DSNHDECP LOAD MODULE
*      - FORMAT DSNHDECM CONTROL BLOCK
*    FINALIZATION
*      - CLOSE FILES
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               ANALYSE STARTUP              '
         LCLA  &ZPRMLNE
         LCLB  &REEV6                    V6+ INDICATOR              V6
ZPARMV6 START , X'6A1Ø'
&ZPRMLNE SETA  8Ø                        SYSPUNCH LRECL=8Ø
         USING ZPARMV6,R15
ZPARMV6 AMODE 31
ZPARMV6 RMODE 24
RØ       EQU   Ø
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5                         BASE FOR CONSTANTS
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
R11      EQU   11
```

```
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         STM   R14,R12,12(R13)    STANDARD LINKAGE CONVENTION
         LR    R1Ø,R15
         LR    R11,R15
         A     R11,=A(4Ø96)
         LR    R12,R11
         A     R12,=A(4Ø96)
         DROP  R15
         USING ZPARMV6,R1Ø,R11,R12
         LR    R9,R13
         L     R13,=A(SAVEAREA)
         LM    R4,R5,=A(SAVEAREA+4Ø96,SAVEAREA+2*4Ø96)
         USING SAVEAREA,R13,R4,R5
         ST    R13,8(,R9)
         ST    R9,4(,R13)
         B     INITIALZ           GO AROUND EYECATCHER/SAVEAREA
         DS    ØD
EYENAME  DC    CL9'ZPARMV6'
EYEDATE  DC    CL9'&SYSDATE.'
EYETIME  DC    CL9'&SYSTIME.'
         DC    CL6'LOEBEN'
         LTORG
DSNZPARM DSECT
         PRINT GEN
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               DSNZPARM EXAMPLE COMPILE '
    DSN6ENV    MVS=XA
    DSN6SPRM   RESTART,ALL,ABEXP=YES,ABIND=YES,AUTH=YES,AUTHCACH=1Ø24,
X3
               BINDNV=BINDADD,BMPTOUT=4,CACHEDYN=NO,CACHEPAC=32768,    X
               CATALOG=DSNCssØ,CDSSRDEF=ANY, ANY 1                     X
               CHGDC=NO,CONTSTOR=NO,DECDIV3=NO,DEFLTID=IBMUSER,        X
               DESCSTAT=NO,DLITOUT=6,DSMAX=2ØØØ,EDMPOOL=7312,          X
               EDPROP=NO, HOPAUTH=BOTH,   V6: BOTH/RUNNER V5:YES/NO    X
               IRLMAUT=YES,IRLMPRC=IRLMPROC,IRLMSID=IRLM,IRLMRWT=6Ø,   X
               IRLMSWT=3ØØ,MAXRBLK=4384,MAXKEEPD=5ØØØ,NUMLKTS=1ØØØ,    X
               NUMLKUS=1ØØØØ,RECALL=YES,RECALLD=12Ø,                   X
               RELCURHL=YES, RETLWAIT=7,  V6: Ø-  V5:  YES/NO          X
               RETVLCFK=NO,                                            X
               RGFCOLID=DSNRGCOL,RGFDBNAM=DSNRGFDB,RGFDEDPL=NO,        X
               RGFDEFLT=ACCEPT,RGFESCP=,RGFFULLQ=YES,RGFINSTL=NO,      X
               RGFNMORT=DSN_REGISTER_OBJT,RGFNMPRT=DSN_REGISTER_APPL,  X
               RRULOCK=NO,SEQCACH=BYPASS,SEQPRES=NO,SITETYP=LOCALSITE,
X3
               SRTPOOL=876,                                            X
               SYSADM=SYSADM,SYSADM2=SYSADM,SYSOPR1=SYSOPR,            X
               SYSOPR2=SYSOPR,                                         X
               TRKRSITE=NO,UTIMOUT=6,XLKUPDLT=NO
    DSN6ARVP   ALCUNIT=BLK,ARCWRTC=(1,3,4),ARCWTOR=YES,                X
               ARCPFX1=DSNCss0.ARCHLOG1,ARCPFX2=DSNCss0.ARCHLOG2,      X
               ARCRETN=9999,BLKSIZE=28672,CATALOG=NO,COMPACT=NO,       X
               PRIQTY=4320,PROTECT=NO,QUIESCE=5,SECQTY=540,TSTAMP=NO,  X
               UNIT=TAPE,UNIT2=
    DSN6LOGP   DEALLCT=(0000),MAXARCH=1000,MAXRTU=2,OUTBUFF=4000,      X
               TWOACTV=YES,TWOARCH=YES,WRTHRSH=20  ARC2FRST=NO    V6
    DSN6SYSP   AUDITST=NO, BACKODUR=5     V6                           X
               CONDBAT=64, TBSBPOOL=BP2, V6                            X
               CTHREAD=70, DBPROTCL=DRDA,            V6                X
               DLDFREQ=5, DSSTIME=5,EXTRAREQ=100,EXTRASRV=100, V6      X
               IDBACK=20,                                              X
               IDFORE=40, IDXBPOOL=BP0, LBACKOUT=AUTO, LOBVALA=2048,   X
               LOGLOAD=50000, LOBVALS=2048, LOGAPSTG=0, V6             X
               MAXDBAT=64,MON=NO,MONSIZE=8192,PCLOSEN=5,PCLOSET=10,    X
               RLF=NO,RLFTBL=01,RLFERR=1000,RLFAUTH=SYSIBM,            X
               ROUTCDE=(1),EXTSEC=NO,SMFACCT=1,SMFSTAT=YES,            X
               STATIME=30,STORMXAB=0,STORPROC=,STORTIME=180,           X
               TRACSTR=YES,TRACTBL=16,URCHKTH=0
    DSN6FAC    DDF=NO,CMTSTAT=ACTIVE,IDTHTOIN=0,RESYNC=2,              X
               RLFERRD=NOLIMIT  TCPALVER=NO,MAXTYPE1=0,TCPKPALV=ENABLE
    DSN6GRP    DSHARE=NO,GRPNAME=,MEMBNAME=,COORDNTR=NO,ASSIST=NO
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - 03.08.99  *
               DSNHDECM EXAMPLE COMPILE '
DSNHDECM DSECT
         PRINT GEN
     DSNHDECM  CHARSET=ALPHANUM,ASCCSID=850,AMCCSID=65534,             X
               AGCCSID=65534,SCCSID=273,MCCSID=65534,GCCSID=65534,     X
               ENSCHEME=EBCDIC,DATE=EUR,DATELEN=0,DECARTH=DEC15,       X
               DECIMAL=PERIOD,DEFLANG=COB2,DELIM=APOST,MIXED=NO,       X
               SQLDELI=APOST,DSQLDELI=APOST,SSID=DB2T,DYNRULS=YES,     X
               STDSQL=YES,TIME=JIS,                                    X
               TIMELEN=0
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - 03.08.99  *
               ANALYSE DSNZPARM             '
ZPARMV6  CSECT
******** I N I T I A L I Z E
INITIALZ DS    0H
         BAS   R14,GETPRTN                GET PARMLIST
         BAS   R14,INITRTN                DO INITIALIZE VALUES
         BAS   R14,HDRLRTN             DO PRINT HEADER LINES
******** M A I N L I N E
MAINLINE DS    0H
** LOAD DSNZPARM IN VIRTUAL STORAGE
         LOAD  EPLOC=LOADNAME,LOADPT=LOADMPTR,DCB=DSNZPRM
         LTR   R15,15
         BNZ   ABEND100
         LR    R15,R0
         LA    R15,0(,R15)
         ST    R15,LOADMPTR
         LR    R7,R15
         LA    R1,Ø(,R1)               REMOVE HIGH ORDER BYTE
         ST    R1,LOADMPTR+4           SAVE LENGTH
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               ANALYSE DSN6SPRM              '
         L     R7,LOADMPTR
         USING DSN6SPRM,R7                ESTABLISH ADDRESSABILITY
         LA    RØ,4
         LA    R1,255(,R7)
         CLC   =CLØ8'DSN6SPRM',4(R7)
         BE    *+12
         BXLE  R7,RØ,*-1Ø
         B     ABEND19Ø
         L     R7,Ø(,R7)
         L     R2,=A(DSN6SPRM)         SECTION TO BE ANALYSED
         CLC   SPRMID,SPRMID-DSN6SPRM(R2)
         BNE   ABEND1Ø1
         CLC   SPRMEYE,SPRMEYE-DSN6SPRM(R2)
         BNE   ABEND1Ø2                SECTION DSN6SPRM NOT FOUND
** DISPLAY ZPARM NAME AND ASSEMBLY DATE
&A       SETC  '+ZPRMLINE-ZPRMTITL'                                  V6
         MVC   ZPRMLINE,ZPRMTITL                                     V6
         BAS   R14,WRITRTN                                           V6
         L     R2,=A(DSN6SPRM)         SECTION TO BE ANALYSED
         MVC   ZPRMLINE(Ø2),=CLØ2'* '                                V6
         MVC   ZTITNAME&A.,EYENAME                                   V6
         MVC   ZTITLVL&A.,SPRMLVL-DSN6SPRM(R2)                       V6
         MVC   ZTITDATE&A.,SPRMDATE-DSN6SPRM(R2)
         L     R2,=A(DSN6SYSP)                                       V6
         MVC   ZTITLVLC&A.,SYSPLVLC-DSN6SYSP(R2)
         MVC   REELINE,ZPRMLINE           SAVE FOR DSNHDECM          V6
         BAS   R14,WRITRTN                                           V6
         L     R2,=A(DSN6SPRM)         SECTION TO BE ANALYSED
         MVC   ZPRMLINE(Ø2),=CLØ2'* '                                V6
         MVC   ZTITNAME&A.,LOADNAME       ZPARM LOAD MODULE NAME
         MVC   ZTITLVL&A.,SPRMLVL         LEVEL COMPILED FOR
         MVC   ZTITDATE&A.,SPRMDATE       DATE COMPILED
         BAS   R14,LVLC                   DEFINE LEVEL COMPILED FOR  V6
         BAS   R14,WRITRTN             DO PRINT LINE
         MVC   ZPRMLINE,=(&ZPRMLNE)C'-'
         MVI   ZPRMCL72,C' '
         MVC   ZPRMCL73,=CLØ8' '
         MVC   ZPRMLINE(Ø2),=CLØ2'* '
         BAS   R14,WRITRTN
&REEV6   SETB  (D'SPRMWAIT)               SET FOR V6 FF TO 1, ELSE Ø V6
*> FORMAT DSN6ENV ****************************************************
         MVC   ZPRMCLØ5(Ø8),=CLØ8'DSN6ENV '
*> MVS       - MVS 37Ø OR XA
         MVC   ZPRMCL16(Ø4),=CLØ4'MVS='   FIELD LITERAL
         MVC   ZPRMCL16+Ø4(Ø3),SPRMMVS    GET ZPARM VALUE
         MVC   ZPRMCL4Ø,=CL32'DEFAULT VALUE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> FORMAT DSN6SPRM ****************************************************
3
         MVC   ZPRMCLØ5(Ø8),=CLØ8'DSN6SPRM'
*> RESTART   - AUTOSTARTED DATABASE/TABLESPACE
         MVC   ZPRMCL16(Ø8),=CLØ8'RESTART,' FIELD LITERAL
         CLC   SPRMVCAT,=CLØ8'SPRMVCAT'
         BNE   ABEND133
         CLC   SPRMDB,=CLØ8'SPRMDB'
         BNE   ABEND134
         TM    SPRMSTRT,X'8Ø'                                      V6
         BO    *+1Ø                                                V6
         MVC   ZPRMCL16(Ø8),=CLØ8'DEFER,  ' FIELD LITERAL
         MVC   ZPRMCL4Ø,=CL32'RESTART TYPE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ALL       - DATABASE LIST
         MVC   ZPRMCL16(Ø4),=CLØ4'ALL,'   FIELD LITERAL
         ICM   RØ,3,SPRMDB#
         LTR   Ø,Ø
         BZ    *+1Ø                                                V6
         MVC   ZPRMCL16(21),=CL21'(..LIST SPECIFIED..),'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ABEXP     - ALLOW/DISALLOW EXPLAIN DURING AUTOBIND
         MVC   ZPRMCL16(Ø9),=CLØ9'ABEXP=NO,'
         TM    SPRMMISZ,B'1ØØØØØØØ'
         BNO   *+1Ø
         MVC   ZPRMCL16+Ø6(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'EXPLAIN DURING AUTOBIND'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ABIND     - AUTOBIND ACTIVATED/DEACTIVATED
         MVC   ZPRMCL16(Ø8),=CLØ8'ABIND=?,'
         CLI   SPRMABN,C'D'               DISABLED=YES
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø6(Ø3),=CLØ3'NO,'
         CLI   SPRMABN,C'E'               ENABLED=YES
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø6(Ø4),=CLØ4'YES,'
         CLI   SPRMABN,C'C'               ENABLED=YES
         BNE   *+1Ø                       COEXIST
         MVC   ZPRMCL16+Ø6(Ø8),=CLØ8'COEXIST,'
         MVC   ZPRMCL4Ø,=CL32'AUTOBIND ENABLED'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ALPOOLX   - ALLOCATION POOL EXTENSION SIZE  - NOT FOR V6 FF
         AIF   (&REEV6).ALPOOLX           NOT IN V6                  V6
         MVC   ZPRMCL16(Ø8),=CLØ8'ALPOOLX=' FIELD LITERAL
         ICM   R9,B'1111',SPRMTXS
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D
         OI    ZPRMCL16+22,X'FØ'
         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
         MVC   ZPRMCL4Ø,=CL32'ALLOCATION POOL EXTENSION'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.ALPOOLX ANOP                                                       V6
*> AUTH      - AUTHORIZATION ENABLED/DISABLED
         MVC   ZPRMCL16(Ø8),=CLØ8'AUTH=NO,'
         CLI   SPRMAUTH,C'E'
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø5(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'AUTHORIZATION ENABLED'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> AUTHCACH  - AUTHORIZATION CACHE
         MVC   ZPRMCL16(Ø9),=CLØ9'AUTHCACH=' FIELD LITERAL
         SR    R9,R9
         ICM   R9,B'ØØ11',SPRMAUCA
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(15),D
         OI    ZPRMCL16+23,X'FØ'
         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
         MVC   ZPRMCL4Ø,=CL32'AUTHORIZATION CACHE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> BINDNV    - BIND OR BINDADD AUTHORITY  FOR DIFFERENT VERSION
         MVC   ZPRMCL16(Ø7),=CLØ7'BINDNV='
         MVC   ZPRMCL16+Ø7(Ø8),SPRMBNVA
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'BIND OR BINDADD AUTHORITY'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> BMPTOUT   - IMS BMP TIMEOUT FACTOR
         MVC   ZPRMCL16(Ø8),=CLØ8'BMPTOUT=' FIELD LITERAL
         SR    R9,R9
         ICM   R9,B'ØØ11',SPRMBMP
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D
         OI    ZPRMCL16+22,X'FØ'
         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
         MVC   ZPRMCL4Ø,=CL32'IMS BMP TIMEOUT FAKTOR'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> CACHEDYN  - CACHE DYNAMIC SQL IN EDM POOL                          V5
         MVC   ZPRMCL16(12),=CL12'CACHEDYN=YES'                      V5
         TM    SPRMMIS2,B'ØØØ1ØØØØ'       BIT 3                      V5
         BO    *+1Ø                                                  V5
         MVC   ZPRMCL16+Ø9(Ø3),=CLØ3'NO '                            V5
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK           V5
         MVI   Ø(1),C','                  PLUG COMMA HERE            V5
         MVC   ZPRMCL4Ø,=CL32'CACHE DYNAMIC SQL IN EDM POOL'         V5
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V5
*> CACHEPAC  - CACHE FOR PACKAGE AUTHORIZATION                        V5
         MVC   ZPRMCL16(Ø9),=CLØ9'CACHEPAC=' FIELD LITERAL           V5
         ICM   R9,15,SPRMPAC              GET ZPARM VALUE            V5
         CVD   R9,D                       CONVERT DECIMAL            V5
         UNPK  ZPRMCL16+Ø9(15),D          PACK TO ZONE               V5
         OI    ZPRMCL16+23,X'FØ'          FIX LAST DIGIT             V5
         MVC   ZEROHOLD,ZPRMCL16+Ø9       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS         V5
         MVC   ZPRMCL16+Ø9(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK           V5
         MVI   Ø(1),C','                  PLUG COMMA HERE            V5
         MVC   ZPRMCL4Ø,=CL32'CACHE FOR PACKAGE AUTHORIZATION'       V5
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V5
*> CACHERAC  - AUTH CACHE FOR ROUTINES                                V6
         AIF   (NOT &REEV6).CACHRAC               IF V6 THEN DO ELSE V6
         MVC   ZPRMCL16(Ø9),=CLØ9'CACHERAC='                         V6
         SR    R9,R9                      ZERO REGISTER              V6
         L     R9,SPRMRAC                 GET ZPARM VALUE            V6
         CVD   R9,D                       CONVERT DECIMAL            V6
         UNPK  ZPRMCL16+Ø9(15),D          PACK TO ZONE NUMERIC       V6
         OI    ZPRMCL16+23,X'FØ'          FIX LAST DIGIT             V6
         MVC   ZEROHOLD,ZPRMCL16+Ø9       MOVE NUMBER IN HOLD AREA   V6
         BAS   R14,DZERORTN               DROP LEADING ZEROS         V6
         MVC   ZPRMCL16+Ø9(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK V6
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK           V6
         MVI   Ø(1),C','                  PLUG COMMA HERE            V6
         MVC   ZPRMCL4Ø,=CL32'AUTHORIZATION CACHE FOR ROUTINES'      V6
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V6
.CACHRAC ANOP                                                        V6
*> CHGDC     - CHANGE DATA CAPTURE ACT/DEACT
         MVC   ZPRMCL16(Ø9),=CLØ9'CHGDC=NO,'
         TM    SPRMMISC,B'ØØØØ1ØØØ'
         BNO   *+1Ø
         MVC   ZPRMCL16+Ø6(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'ACTIVATE CHANGED DATA CAPTURE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> CATAGALOG - VSAM CATALOG NAME
         MVC   ZPRMCL16(Ø8),=CLØ8'CATALOG='
         LA    R8,SPRM
         A     R8,SPRMVCOF
         MVC   ZPRMCL16+Ø8(Ø8),12(R8)
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'VSAM CATALOG NAME'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> CDSSRDEF  - CURRENT DEGREE SPECIAL REGISTER
         MVC   ZPRMCL16(Ø9),=CLØ9'CDSSRDEF='
         MVC   ZPRMCL16+Ø9(L'SPRMCDEG),SPRMCDEG
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'CURRENT DEGREE SPECIAL REGISTER'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> CONTSTOR  - CONTRACT DBM1 CT STORAGE
         MVC   ZPRMCL16(12),=CL12'CONTSTOR=NO,'
         TM    SPRMMSC2,B'ØØØØØØ1Ø'
         BZ    *+1Ø
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'CONTRACT DBM1 CT STORAGE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DBCHK     - CHECK DB FOR CONSISTENCY
         MVC   ZPRMCL16(Ø9),=CLØ9'DBCHK=NO,'
         TM    SPRMDBCK,X'8Ø'
         BZ    *+1Ø
         MVC   ZPRMCL16+Ø6(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'SERVICE AID - CHECK DB CONSISTENCY'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DECDIV    - DECIMAL DIVIDE OPTION
         MVC   ZPRMCL16(11),=CL11'DECDIV3=NO,'
         TM    SPRMMISC,B'Ø1ØØØØØØ'
         BNO   *+1Ø
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'DECIMAL DIVIDE OPTION'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DEFIXTP   - DEFAULT INDEX TYPE         NOT V6
         AIF   (&REEV6).DEFIXTP                                      V6
         MVC   ZPRMCL16(1Ø),=CL1Ø'DEFIXTP=2,'
         CLI   SPRMDXTP,2
         BE    *+8
         MVI   ZPRMCL16+Ø8,C'1'
         MVC   ZPRMCL4Ø,=CL32'DEFAULT INDEX TYPE (V5)'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.DEFIXTP ANOP                                                        V6
*> DEFLTID   - SYSTEM DEFAULT USER ID
         MVC   ZPRMCL16(Ø8),=CLØ8'DEFLTID='
         MVC   ZPRMCL16+Ø8(Ø8),SPRMDFID
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'SYSTEM DEFAULT USERID'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DESCSTAT  - DESCRIBE STATIC SQL - YES/NO
```

```
         MVC   ZPRMCL16(12),=CL12'DESCSTAT=NO,'
         TM    SPRMMIS2,X'8Ø'
         BNO   *+1Ø
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'REMOTE DESCRIBE AS STATIC SQL'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DLITOUT   - DLI TIMEOUT FACTOR
         MVC   ZPRMCL16(Ø8),=CLØ8'DLITOUT=' FIELD LITERAL
         SR    R9,R9
         ICM   R9,B'ØØ11',SPRMDLI
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D
         OI    ZPRMCL16+22,X'FØ'
         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
         MVC   ZPRMCL4Ø,=CL32'IMS/DLI WAIT TIMEOUT FACTOR'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSMAX     - MAX NUMBER OF DATASETS CONCURRENLY IN USE
         MVC   ZPRMCL16(Ø6),=CLØ6'DSMAX=' FIELD LITERAL
         L     R1,=A(B'ØØ11')
         CLI   =AL1(L'SPRMDSMX),2
         BE    *+8
         L     R1,=A(B'1111')
         SR    R9,R9                      ZERO REGISTER
         ICM   R9,Ø,SPRMDSMX              GET ZPARM VALUE
         EX    R1,*-4
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø6(Ø7),D          PACK TO ZONE
         OI    ZPRMCL16+12,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø6       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø6(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'MAXIMUM CONCURRENT DATASETS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> EDMPOOL   - EDMPOOL SIZE
         MVC   ZPRMCL16(Ø8),=CLØ8'EDMPOOL=' FIELD LITERAL
         SR    R8,R8                      ZERO REGISTER
         L     R9,SPRMEDPL                GET ZPARM VALUE
         D     R8,=F'1Ø24'                DIVIDE BY 1Ø24
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE
         OI    ZPRMCL16+22,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
         MVC   ZPRMCL4Ø,=CL32'EDMPOOL SIZE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> EDMDSPAC  - EDMPOOL DATA SPACE SIZE                               V6
         AIF   (NOT D'SPRMEDDS).EDMDSPC                              V6
         MVC   ZPRMCL16(Ø9),=CLØ9'EDMDSPAC=' FIELD LITERAL           V6
         SR    R8,R8                      ZERO REGISTER              V6
         L     R9,SPRMEDDS                GET ZPARM VALUE            V6
         D     R8,=F'1Ø24'                DIVIDE BY 1Ø24             V6
         CVD   R9,D                       CONVERT DECIMAL            V6
         UNPK  ZPRMCL16+Ø9(15),D          PACK TO ZONE               V6
         OI    ZPRMCL16+23,X'FØ'          FIX LAST DIGIT             V6
         MVC   ZEROHOLD,ZPRMCL16+Ø9       MOVE NUMBER IN HOLD AREA   V6
         BAS   R14,DZERORTN               DROP LEADING ZEROS         V6
         MVC   ZPRMCL16+Ø9(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK V6
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK           V6
         MVI   Ø(1),C','                  PLUG COMMA HERE            V6
         MVC   ZPRMCL4Ø,=CL32'EDMPOOL DATA SPACE SIZE'               V6
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V6
.EDMDSPC ANOP                                                        V6
*> EDPROP    -
         MVC   ZPRMCL16(1Ø),=CL1Ø'EDPROP=NO '
         TM    SPRMMISC,B'ØØØØØ1ØØ'       BIT 6
         BNO   *+1Ø
         MVC   ZPRMCL16+Ø7(Ø3),=CLØ4'YES'
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'ALLOW CHANGES TO CAPTURED TABLES'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> HOPAUTH   - PKG/RUNNER AUTH TO HOP SITE
         MVC   ZPRMCL16(11),=CL11'HOPAUTH=NO,'
         TM    SPRMMISZ,B'Ø1ØØØØØØ'       BIT 2
         BNO   *+1Ø
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,'
         AIF   (NOT D'SPRMWAIT).HOPAUTH   IF NOT DB2 V6 THEN         V6
         MVC   ZPRMCL16+Ø8(Ø5),=CLØ5'BOTH,'
         TM    SPRMMISZ,B'Ø1ØØØØØØ'       BIT 2
         BO    *+1Ø
         MVC   ZPRMCL16+Ø8(Ø7),=CLØ7'RUNNER,'
.HOPAUTH MVC   ZPRMCL4Ø,=CL32'3RD SITE HOP REQUESTER AUTHORITY'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> IRLAUT    - IRLM AUTO START
         MVC   ZPRMCL16(11),=CL11'IRLMAUT=YES' FIELD LITERAL
         TM    SPRMIAUT,X'8Ø'
         BO    *+1Ø
         MVC   ZPRMCL16+Ø8(Ø3),=CLØ3'NO ' IRLMAUT=NO
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'IRLM AUTOSTART'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> IRLMPRC   - IRLM STARTED PROC
         MVC   ZPRMCL16(Ø8),=CLØ8'IRLMPRC=' FIELD LITERAL
         MVC   ZPRMCL16+Ø8(Ø8),SPRMIPRC   GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'IRLM PROZEDURE NAME'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> IRLMSID   - IRLM SUBSYSTEM ID
         MVC   ZPRMCL16(Ø8),=CLØ8'IRLMSID=' FIELD LITERAL
         MVC   ZPRMCL16+Ø8(Ø4),SPRMISID   GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'IRLM SUBSYSTEM ID'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> IRLMRWT   - IRLM TIMEOUT VALUE
         MVC   ZPRMCL16(Ø8),=CLØ8'IRLMRWT=' FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         L     R9,SPRMTOUT
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D
         OI    ZPRMCL16+22,X'FØ'
         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
         MVC   ZPRMCL4Ø,=CL32'DB2 MAXIMUM SECONDS WAIT FOR LOCK'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> IRLMSWT   - IRLM TIME TO WAIT FOR START
         MVC   ZPRMCL16(Ø8),=CLØ8'IRLMSWT=' FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         L     R9,SPRMISWT                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE
         OI    ZPRMCL16+22,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
         MVC   ZPRMCL4Ø,=CL32'IRLM START COMPLETION DELAY'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> LEMAX     - LE ELEMENTS                                            V6
         AIF   (NOT D'SPRMLEM).LEMAXX                                 V6
         MVC   ZPRMCL16(Ø6),=CLØ6'LEMAX='                             V6
         LH    R9,SPRMLEM                 LEMAX PARM                  V6
         CVD   R9,D                       CONVERT DECIMAL             V6
         UNPK  ZEROHOLD(Ø7),D+L'D-3(Ø3)                               V6
         OI    ZEROHOLD+Ø6,C'Ø'                                       V6
         BAS   R14,DZERORTN               DROP LEADING ZEROS         V6
         MVC   ZPRMCL16+Ø6(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK V6
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK           V6
         MVI   Ø(1),C','                  PLUG COMMA HERE            V6
         MVC   ZPRMCL4Ø,=CL32'LE ELEMENTS'                           V6
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V6
.LEMAXX  ANOP                                                        V6
*> MAXKEEPD  - DYNAMIC SQL KEPT AFTER COMMIT                         V5
         MVC   ZPRMCL16(Ø9),=CLØ9'MAXKEEPD=' FIELD LITERAL           V5
         ICM   R9,15,SPRMMXKD             GET ZPARM VALUE            V5
         CVD   R9,D                       CONVERT DECIMAL            V5
         UNPK  ZPRMCL16+Ø9(15),D          PACK TO ZONE               V5
         OI    ZPRMCL16+23,X'FØ'          FIX LAST DIGIT             V5
         MVC   ZEROHOLD,ZPRMCL16+Ø9       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS         V5
         MVC   ZPRMCL16+Ø9(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK           V5
         MVI   Ø(1),C','                  PLUG COMMA HERE            V5
         MVC   ZPRMCL4Ø,=CL32'SYSTEM KEEPDYNAMIC SQL ALLOWED'        V5
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V5
*> MAXRBLK   - MAX RID BLOCK
         MVC   ZPRMCL16(Ø8),=CLØ8'MAXRBLK=' FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         L     R9,SPRMRMAX                GET ZPARM VALUE
         M     R8,=F'16'                  MULTIPLY BY 16
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE
         OI    ZPRMCL16+22,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
         MVC   ZPRMCL4Ø,=CL32'RID SIZE IN KBYTES'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> MINRBLK   - LEAST # OF RIDLISTS FOR EACH RIDMAP
         MVC   ZPRMCL16(Ø8),=CLØ8'MINRBLK=' FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         ICM   R9,B'1111',SPRMRMIN        GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE
         OI    ZPRMCL16+22,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
         MVC   ZPRMCL4Ø,=CL32'MIN RIDLISTS IN EACH RIDMAP'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> NUMLKTS   - MAX PAGE LOCKS PER TABLESPACE
```

```
         MVC   ZPRMCL16(Ø8),=CLØ8'NUMLKTS=' FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         L     R9,SPRMLKTS
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D
         OI    ZPRMCL16+22,X'FØ'
         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
         MVC   ZPRMCL4Ø,=CL32'LOCKS PER TABLESPACE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> NUMLKUS   - MAX PAGE LOCKS PER APPLICATION
         MVC   ZPRMCL16(Ø8),=CLØ8'NUMLKUS=' FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         L     R9,SPRMLKUS                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE
         OI    ZPRMCL16+22,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
         MVC   ZPRMCL4Ø,=CL32'LOCKS PER USER'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> OJPERFEH  - OUTER JOIN PERFORMANCE ENHANCEMENTS                   V5
         AIF   (NOT D'SPRMTYP).OJPERFE                V5 ELSE        V5
         MVC   ZPRMCL16(13),=CL13'OJPERFEH=YES,'                     V5
         TM    SPRMMSC2,B'ØØØØØ1ØØ'                                  V5
         BO    *+1Ø                                                  V5
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'NO, '                           V5
         MVC   ZPRMCL4Ø,=CL32'OUTER JOIN PERFORMANCE ENHANCEMENTS'   V5
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V5
.OJPERFE ANOP                                                        V6
*> OPTHINTS  - ALLOW OPTIMIZER HINTS                                 V6
         AIF   (NOT &REEV6).OPHIX                                    V6
         MVC   ZPRMCL16(13),=CL13'OPTHINTS=YES,'                     V6
         TM    SPRMMSC2,B'ØØØØ1ØØØ'                                  V6
         BO    *+1Ø                                                  V6
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'NO, '                           V6
         MVC   ZPRMCL4Ø,=CL32'ALLOW OPTIMIZER HINTS'                 V6
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V6
.OPHIX   ANOP                                                        V6
*> PARAMDEG  - PARALLEL GROUP DEGREE LIMIT - PQ28414                 V5
         AIF   (NOT D'SPRMMDEG).PARAMDG                              V5
         L     R9,SPRMMDEG                IF STILL DEFAULT (Ø)       V6
         LTR   R9,R9                      THEN                       V6
         BZ    XPARAMDG                      IGNORE PRINTOUT         V6
         MVC   ZPRMCL16(Ø9),=CLØ9'PARAMDEG=' LITERAL                 V5
         L     R9,SPRMMDEG                                           V5
         CVD   R9,D                       CONVERT DECIMAL            V5
         UNPK  ZPRMCL16+Ø9(Ø7),D                                     V5
         OI    ZPRMCL16+15,X'FØ'                                     V5
         MVC   ZEROHOLD,ZPRMCL16+Ø9       MOVE NUMBER IN HOLD AREA   V5
         BAS   R14,DZERORTN               DROP LEADING ZEROS         V5
         MVC   ZPRMCL16+Ø9(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK V5
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK           V5
         MVI   Ø(1),C','                  PLUG COMMA HERE            V5
         MVC   ZPRMCL4Ø,=CL32'PARALLEL GROUP DEGREE LIMIT'           V5
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V5
.PARAMDG ANOP                                                        V5
XPARAMDG DS    ØH                                                    V6
*> SPRMPKYU  - ALLOW UPDATE OF PARTITIONING KEY                      V6
         AIF   (NOT D'SPRMPKYU).PKYU                                 V6
         MVC   ZPRMCL16(14),=CL14'PARTKEYU=YES, ' PARTKEYU=YES       V6
         CLI   SPRMPKYU,C'N'              PARTKEYU=NO
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø9(5),=C'NO,  '
         CLI   SPRMPKYU,C'S'              PARTKEYU=SAME
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø9(5),=C'SAME,'
         MVC   ZPRMCL4Ø,=CL32'ALLOW UPDATE OF PARTITIONING KEY'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.PKYU    ANOP                                                        V6
XPKYU    DS    ØH                                                    V6
*> RECALL    - HSM AUTO RECALL
         MVC   ZPRMCL16(1Ø),=CL1Ø'RECALL=NO,'
         TM    SPRMHRCL,X'8Ø'
         BNO   *+1Ø
         MVC   ZPRMCL16+Ø7(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'HSM AUTO RECALL'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RECALLD   - HSM AUTO DELAY TIME
         MVC   ZPRMCL16(Ø8),=CLØ8'RECALLD=' LITERAL
         SR    R9,R9                      ZERO REGISTER
         LH    R9,SPRMHRCD
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),D
         OI    ZPRMCL16+14,X'FØ'
         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
         MVC   ZPRMCL4Ø,=CL32'SECONDS WAIT HSM AUTO RECALL COMPLETE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RELCURHL  - RELEASE LOCKS FOR CURSOR WITH HOLD AT COMMIT          V5
         AIF   (NOT D'SPRMPAC).RELCUHL             IF V5 THEN        V5
         MVC   ZPRMCL16(12),=CL12'RELCURHL=NO '                      V5
         TM    SPRMMIS2,B'00001000'       BIT 4                      V5
         BZ    *+10                                                  V5
         MVC   ZPRMCL16(12),=CL12'RELCURHL=YES'                      V5
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK           V5
         MVI   0(1),C','                  PLUG COMMA HERE            V5
         MVC   ZPRMCL40,=CL32'RELEASE LOCKS FOR HELD CURSOR AT COMMIT'
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V5
.RELCUHL ANOP                                         V5 ELSE        V5
*> RETLWAIT  - IRLM WAIT FOR INCOMPATIBLE RETAINED LOCKS
         AIF   (D'SPRMWAIT).RETLV6        IF NOT DB2 V6 THEN         V6
         MVC   ZPRMCL16(12),=CL12'RETLWAIT=NO,'
         TM    SPRMMSC2,B'00100000'
         BZ    *+10
         MVC   ZPRMCL16+09(04),=CL04'YES,'
         MVC   ZPRMCL40,=CL32'IRLMWAIT FOR INCOMPATIBLE RETAINED LOCKS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
         AGO   .RETLV5E
.RETLV6  ANOP                                                        V6
         MVC   ZPRMCL16(09),=CL09'RETLWAIT='                         V6
         SR    R9,R9                      ZERO REGISTER              V6
         ICM   R9,B'0011',SPRMWAIT                                   V6
         CVD   R9,D                       CONVERT DECIMAL            V6
         UNPK  ZPRMCL16+09(15),D                                     V6
         OI    ZPRMCL16+23,X'F0'                                     V6
         MVC   ZEROHOLD,ZPRMCL16+09       MOVE NUMBER IN HOLD AREA   V6
         BAS   R14,DZERORTN               DROP LEADING ZEROS         V6
         MVC   ZPRMCL16+09(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK V6
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK           V6
         MVI   0(1),C','                  PLUG COMMA HERE            V6
         MVC   ZPRMCL40,=CL32'IRLMWAIT FOR INCOMPATIBLE RETAINED LOCKS'
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V6
.RETLV5E ANOP
*> RETVLCFK  - RETRIEVE VARYING LENGTH CHAR FROM KEY                 V6
         AIF   (NOT &REEV6).RETVLCF               IF V6 THEN DO ELSE V5
         MVC   ZPRMCL16(12),=CL12'RETVLCFK=NO,'                      V6
         TM    SPRMMIS2,1                                            V6
         BZ    *+10                                                  V6
         MVC   ZPRMCL16+09(04),=CL04'YES,'                           V6
         MVC   ZPRMCL40,=CL32'ALLOW KEY WITH VARCHAR'                V6
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V6
.RETVLCF ANOP                                                        V6
*> RGFCOLID  - DDL REGISTRATION TABLE OWNER LID
         MVC   ZPRMCL16(09),=CL09'RGFCOLID='
         MVC   ZPRMCL16+09(08),SPRMREGC
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   0(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL40,=CL32'DDL REGISTRATION OWNER ID'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFDBNAM  - DDL REGISTRATION DATABASE NAME
```

```
         MVC   ZPRMCL16(Ø9),=CLØ9'RGFDBNAM='
         MVC   ZPRMCL16+Ø9(Ø8),SPRMREGN
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'DDL REGISTRATION DATABASE NAME'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFDEDPL  - DDL REGISTRATION DEDICATED APPLS
         MVC   ZPRMCL16(12),=CL12'RGFDEDPL=NO,'
         TM    SPRMREGF,B'Ø1ØØØØØØ'       BIT 6
         BNO   *+1Ø
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'DDL ONLY BY REGISTERED APPLICATIONS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFDEFLT  - DDL REGISTRATION DEFAULT
         MVC   ZPRMCL16(16),=CL16'RGFDEFLT=REJECT,'
         TM    SPRMREGF,B'ØØØ1ØØØØ'       IF ACCEPT
         BZ    *+1Ø                       THEN
         MVC   ZPRMCL16+Ø9(Ø7),=CLØ7'ACCEPT,' DO ACCEPT
         TM    SPRMREGF,B'ØØØØ1ØØØ'       IF APPL
         BZ    *+1Ø                       THEN
         MVC   ZPRMCL16+Ø9(Ø7),=CLØ7'APPL,  ' DO APPL
         MVC   ZPRMCL4Ø,=CL32'NOT REGISTERED DDL REACTION DEFAULT'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFESCP   - DDL REGISTRATION ESCAPE CHAR
         MVC   ZPRMCL16(Ø8),=CLØ8'RGFESCP=' FIELD LITERAL
         MVC   ZPRMCL16+Ø8(Ø1),SPRMREGE   GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'DDCS DEFAULT ESCAPE IN ART/ORT SEARCHES'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFFULLQ  - DDL REGISTRATION FULLY QUALIFIED NAME
         MVC   ZPRMCL16(12),=CL12'RGFFULLQ=NO,'
         TM    SPRMREGF,B'ØØ1ØØØØØ'       BIT 3
         BNO   *+1Ø
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'OBJECT LOOKUP WITH FULL LOCAL NAME'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFINSTL  - DDL REGISTRATION INSTALLED
         MVC   ZPRMCL16(12),=CL12'RGFINSTL=NO,'
         TM    SPRMREGF,B'1ØØØØØØØ'       BIT 1
         BNO   *+1Ø
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'VALIDATE DDL STATEMENTS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFNMORT  - DDL REGISTRATION ORT NAME
         MVC   ZPRMCL16(Ø9),=CLØ9'RGFNMORT='
         MVC   ZPRMCL16+Ø9(L'SPRMREGO),SPRMREGO
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø+Ø4(L'ZPRMCL4Ø-4),=CL32'NAME OF OBJECT RGN TBL'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RGFNMPRT  - DDL REGISTRATION ART NAME
         MVC   ZPRMCL16(Ø9),=CLØ9'RGFNMPRT='
         MVC   ZPRMCL16+Ø9(L'SPRMREGA),SPRMREGA
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø+Ø4(L'ZPRMCL4Ø-4),=CL32'NAME OF APPL. RGN TBL'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RRULOCK   - RR U-LOCK FOR CURRENT PAGE
         MVC   ZPRMCL16(11),=CL11'RRULOCK=NO,'
         TM    SPRMMISZ,B'ØØØØØ1ØØ'       BIT 5
         BZ    *+1Ø
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'RR U-LOCK FOR CURRENT PAGE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SEQCACH   - SEQ MODE/BYPASS 339Ø CACHE
         MVC   ZPRMCL16(12),=CL12'SEQCACH=SEQ,'
         TM    SPRMMISZ,B'ØØØ1ØØØØ'       BIT 4
         BO    *+1Ø
         MVC   ZPRMCL16+Ø8(Ø7),=CLØ7'BYPASS,'
         MVC   ZPRMCL4Ø,=CL32'SET SEQ MODE BYPASS IN I/O COMMAND'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SEQPRES   - SEQU. UTILITY DATA IN 399Ø CACHE
         MVC   ZPRMCL16(11),=CL11'SEQPRES=NO,'
         TM    SPRMMIS2,B'Ø1ØØØØØØ'       IF SEQPRES=YES
         BZ    *+1Ø                          THEN
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,'       SET YES
         MVC   ZPRMCL4Ø,=CL32'UTILITIES CAN CACHE SEQUENTIAL DATA'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SITETYP   - SITE TYPE                                             V5
         AIF   (NOT D'SPRMTYP).SITETYP                V5 ELSE        V5
         MVC   ZPRMCL16(18),=CL18'SITETYP=LOCALSITE,'                V5
         TM    SPRMTYP,B'1ØØØØØØØ'        BIT 1                      V5
         BO    *+1Ø                                                  V5
         MVC   ZPRMCL16+Ø8(13),=CL13'RECOVERYSITE,'                  V5
         MVC   ZPRMCL4Ø,=CL32'TYPE OF RESTART'                       V5
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V5
.SITETYP ANOP                                                        V6
*> TRKRSITE  - TRACKER SITE                                          V5
         AIF   (NOT D'SPRMTYP).TRKSITE                V5 ELSE        V5
         MVC   ZPRMCL16(12),=CL12'TRKRSITE=NO,'                      V5
         TM    SPRMTYP,B'Ø1ØØØØØØ'        BIT 1                      V5
         BZ    *+1Ø                                                  V5
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'YES,'                           V5
         MVC   ZPRMCL4Ø,=CL32'SITE IS USED FOR TRACKER'              V5
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V5
.TRKSITE ANOP                                                        V6
*> XLKUPDLT  - X LOCK FOR SEARCHED UPDATE/DELETE                     V5
         AIF   (NOT D'SPRMTYP).XLKUPDL                V5 ELSE        V5
         MVC   ZPRMCL16(12),=CL12'XLKUPDLT=NO,'                      V5
         TM    SPRMTYP,B'ØØ1ØØØØØ'                                   V5
         BZ    *+1Ø                                                  V5
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'YES,'                           V5
         MVC   ZPRMCL4Ø,=CL32'X LOCK FOR SEARCHED UPDATE/DELETE'     V5
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V5
.XLKUPDL ANOP                                                        V5
*> SRTPOOL   - SORT POOL
         MVC   ZPRMCL16(Ø8),=CLØ8'SRTPOOL=' LITERAL
         SR    R8,R8                      ZERO REGISTER
         L     R9,SPRMSORP
         D     R8,=F'1Ø24'                DIVIDE BY 1Ø24
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D
         OI    ZPRMCL16+22,X'FØ'
         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
         MVC   ZPRMCL4Ø,=CL32'SIZE OF SORT POOL'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SYSADM    - SYSTEM ADMINISTRATOR 1
         MVC   ZPRMCL16(Ø7),=CLØ7'SYSADM='
         MVC   ZPRMCL16+Ø7(Ø8),SPRMSADM
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'SYSADM1'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SYSADM2   - SYSTEM ADMINISTRATOR 2
         MVC   ZPRMCL16(Ø8),=CLØ8'SYSADM2='
         MVC   ZPRMCL16+Ø8(Ø8),SPRMADM2
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'SYSADM2'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SYSOPR1   - SYSTEM OPERATOR 1
         MVC   ZPRMCL16(Ø8),=CLØ8'SYSOPR1='
         MVC   ZPRMCL16+Ø8(Ø8),SPRMOPR1
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'SYSOPR1'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SYSOPR2   - SYSTEM OPERATOR 2
         MVC   ZPRMCL16(Ø8),=CLØ8'SYSOPR2='
         MVC   ZPRMCL16+Ø8(Ø8),SPRMOPR2
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'SYSOPR2'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> UTIMOUT   - UTILITY TIMEOUT FACTOR
```

```
         MVC   ZPRMCL16(Ø8),=CLØ8'UTIMOUT=' FIELD LITERAL
         SR    R9,R9                      ZERO REGISTER
         LH    R9,SPRMUTO                 GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),D          PACK TO ZONE NUMERIC
         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 COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'UTILITY TIME OUT FACTOR'
         BAS   R14,ZWRTRTN             DO PRINT LINE
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               DSN6ARVP                 '
*> FORMAT DSN6ARVP ****************************************************
3
         USING DSN6ARVP,R7
         L     R7,LOADMPTR
         LA    RØ,4
         LA    R1,255(,R7)
         CLC   =CLØ8'DSN6ARVP',4(R7)
         BE    *+12
         BXLE  R7,RØ,*-1Ø
         B     ABEND192
         L     R7,Ø(,R7)
         L     R2,=A(DSN6ARVP)         SECTION TO BE ANALYSED
         CLC   ARVPID,ARVPID-DSN6ARVP(R2)
         BNE   ABEND1Ø3
         CLC   ARVPEID,ARVPEID-DSN6ARVP(R2)
         BNE   ABEND1Ø3                SECTION DSN6ARVP NOT FOUND
         MVC   ZPRMCLØ5(Ø8),=CLØ8'DSN6ARVP'
*> ALCUNIT   - ARCHIVE ALLOCATION UNIT
         MVC   ZPRMCL16(11),=CL11'ALCUNIT=CYL'
         MVC   WRKPFLG1,ARVPFLG1       SAVE
         TM    ARVPFLG1,B'Ø1ØØØØØØ'    CYL ?
         BO    FLG145                  Y
         MVC   ZPRMCL16+Ø8(Ø3),=CLØ3'TRK'
         TM    ARVPFLG1,B'ØØ1ØØØØØ'    TRK ?
         BO    FLG145                  Y
         MVC   ZPRMCL16+Ø8(Ø3),=CLØ3'BLK' DEFAULT IS BLK
FLG145   TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE ALLOCATION UNIT'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ARCWRTC   - ARCHIVE WRITE ROUTE CODE
         MVC   ZPRMCL16(Ø8),=CLØ8'ARCWRTC=' FIELD LITERAL
         MVC   WORKB16,ARVPWTØ1+X'88'
         BAS   R14,BIT16RTN
         CLI   WORKCHR1,C')'              IF ) MEANS ALL BITS ARE Ø
         BNE   *+1Ø                       N. GO ON
         MVC   WORKCHAR(Ø2),=CLØ2'NO'     Y. SAY NO HERE
         MVC   ZPRMCL16+Ø8(48),WORKCHAR
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         LA    RØ,2(,1)                   IF FLAGS EXTENT INTO TEXT
         C     RØ,=A(ZPRMCL4Ø)            THEN
         BH    *+1Ø                          DO NOT SET EXPLAINING TEXT
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE MSG ROUTE CODE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ARCWTOR   - ARCHIVE WRITE TO OPERATOR REPLY
         MVC   ZPRMCL16(11),=CL11'ARCWTOR=NO,'
         TM    ARVPFLG1,B'ØØØØ1ØØØ'       BIT 5 ON
         BZ    *+1Ø                       Y.
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE WTOR REQUIRED'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ARCPFX1   - ARCHIVE PREFIX NAME 1
         MVC   ZPRMCL16(Ø8),=CLØ8'ARCPFX1=' FIELD LITERAL
         MVC   ZPRMCL16+Ø8(35),ARVPRE1N   GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ARCPFX2   - ARCHIVE PREFIX NAME 2
         MVC   ZPRMCL16(Ø8),=CLØ8'ARCPFX2=' FIELD LITERAL
         MVC   ZPRMCL16+Ø8(35),ARVPRE2N   GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ARCRETN   - ARCHIVE RETENTION PERIOD
         MVC   ZPRMCL16(Ø8),=CLØ8'ARCRETN='
         SR    R9,R9                      ZERO REGISTER
         LH    R9,ARVPRETN                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),D          PACK TO ZONE NUMERIC
         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
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE RETENTION PERIOD'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> BLKSIZE   - ARCHIVE BLOCKSIZE
         MVC   ZPRMCL16(Ø8),=CLØ8'BLKSIZE='
         SR    R9,R9                      ZERO REGISTER
         L     R9,ARVPBKSZ                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE NUMERIC
         OI    ZPRMCL16+22,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
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE BLOCKSIZE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> CATALOG   - CATALOG ARCHIVE DATASET NAME
         MVC   ZPRMCL16(11),=CL11'CATALOG=NO,'
         TM    ARVPFLG1,B'1ØØØØØØØ'       IF CATALOG=YES
         BZ    *+1Ø                       THEN
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,'    SET.
         MVC   ZPRMCL4Ø,=CL32'CATALOG ARCHIVE DATASET'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> COMPACT   - COMPACT ENABLED/DISABLED
         MVC   ZPRMCL16(11),=CL11'COMPACT=NO,'
         TM    ARVPFLG1,B'ØØØØØ1ØØ'       BIT 6 ON
         BNO   *+1Ø                       Y.
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,' N.
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE TAPE COMPACT WITH IDRC'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> MSVGP     - NAME OF A GROUP OF MSS VOLUMES FOR ARC LOG DS 1
         MVC   ZPRMCL16(Ø6),=CLØ6'MSVGP=' FIELD LITERAL
         MVC   ZPRMCL16+Ø6(Ø8),ARVPMSV1   GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE MSS GROUP'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> MSVGP2    - NAME OF A GROUP OF MSS VOLUMES FOR ARC LOG DS
         MVC   ZPRMCL16(Ø7),=CLØ7'MSVGP2=' FIELD LITERAL
         MVC   ZPRMCL16+Ø7(Ø8),ARVPMSV2   GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE MSS GROUP'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> PRIQTY    - PRIMARY SPACE ALLOCATION
         MVC   ZPRMCL16(Ø7),=CLØ7'PRIQTY='
         SR    R9,R9                      ZERO REGISTER
         L     R9,ARVPRISP                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø7(15),D          PACK TO ZONE NUMERIC
         OI    ZPRMCL16+21,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø7       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø7(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE ALLOCATION PRIMARY SPACE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> PROTECT   - RACF PROTECTION OF ARCHIVE LOG DATA SET
         MVC   ZPRMCL16(11),=CL11'PROTECT=NO,'
         TM    ARVPFLG1,B'ØØØ1ØØØØ'       BIT 4 ON
         BNO   *+1Ø                       Y.
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,' N.
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE RACF PROTECTION'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> QUIESCE     - MAX QUIESCE PERIOD
         MVC   ZPRMCL16(Ø8),=CLØ8'QUIESCE='
         SR    R9,R9                      ZERO REGISTER
         LH    R9,ARVPMQP                 GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),D          PACK TO ZONE NUMERIC
         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
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE LOG MODE(QUIESCE) MAX PERIOD'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SECQTY      - SECONDARY SPACE ALLOCATION
         MVC   ZPRMCL16(Ø7),=CLØ7'SECQTY='
         SR    R9,R9                      ZERO REGISTER
         L     R9,ARVPSECS                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø7(15),D          PACK TO ZONE NUMERIC
         OI    ZPRMCL16+21,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø7       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø7(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE LOG SECONDARY SPACE ALLOCATION'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> TSTAMP      - TIME STAMP IN ARCHIVE LOG DATA SET
         MVC   ZPRMCL16(1Ø),=CL1Ø'TSTAMP=NO,'
         TM    ARVPFLG1,B'ØØØØØØ1Ø'       BIT 7 ON
         BNO   *+1Ø                       Y.
         MVC   ZPRMCL16+Ø7(Ø4),=CLØ4'YES,' N.
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE LOG MIDDLE-FIX IS TIMESTAMP'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> UNIT        - TAPE DEVICE TYPE
         MVC   ZPRMCL16(Ø5),=CLØ5'UNIT='  FIELD LITERAL
         MVC   ZPRMCL16+Ø5(Ø8),ARVPUNT1   GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE TAPE UNIT TYPE'
         BAS   R14,WRITRTN             DO PRINT LINE
         MVC   ZPRMCL16(Ø6),=CLØ6'UNIT2=' FIELD LITERAL
         MVC   ZPRMCL16+Ø6(Ø8),ARVPUNT2   GET ZPARM VALUE
         MVC   ZPRMCL4Ø,=CL32'ARCHIVE TAPE UNIT TYPE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               DSN6LOGP                '
*> FORMAT DSN6LOGP    ***********************************************
         USING DSN6LOGP,R7
         L     R7,LOADMPTR
         LA    RØ,4
         LA    R1,255(,R7)
         CLC   =CLØ8'DSN6LOGP',4(R7)
         BE    *+12
         BXLE  R7,RØ,*-1Ø
         B     ABEND194
         L     R7,Ø(,R7)
         L     R2,=A(DSN6LOGP)         SECTION TO BE ANALYSED
         CLC   LOGPID,LOGPID-DSN6LOGP(R2)
         BNE   ABEND1Ø4
         CLC   LOGPEID,LOGPEID-DSN6LOGP(R2)
         BNE   ABEND1Ø4                SECTION DSN6LOGP NOT FOUND
         MVC   ZPRMCLØ5(Ø8),=CLØ8'DSN6LOGP'
*> DEALLCT   - DEALLOCATION TIME IN MINUTES
         MVC   ZPRMCL16(Ø12),=CL12'DEALLCT=(ØØ) '
         LH    R9,LOGPDMIN                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(Ø7),D          PACK TO ZONE NUMERIC
         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(Ø7),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         LR    R8,R1                                                 V6
         BCTR  R1,Ø                                                  V6
         CLI   Ø(R1),C'('                 IF BLANK MINUTES           V6
         BNE   *+12                          THEN                    V6
         MVI   Ø(R8),C'Ø'                        SET MINUNTES=Ø      V6
         LA    R8,1(,R8)                                             V6
         MVI   Ø(R8),C','                 PLUG COMMA HERE
         LA    R8,1(,R8)                  NEW START OF STRING FOR SEC
         LH    R9,LOGPDSEC                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  Ø(Ø7,R8),D                 PACK TO ZONE NUMERIC
         OI    Ø6(R8),X'FØ'               FIX LAST DIGIT
         MVC   ZEROHOLD,Ø(R8)             MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   Ø(Ø7,R8),ZEROHOLD          MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         LR    R8,R1                                                 V6
         BCTR  R1,Ø                                                  V6
         CLI   Ø(R1),C','                 IF Ø MINUTES               V6
         BNE   *+12                          THEN                    V6
         MVI   Ø(R8),C'Ø'                        SET MINUNTES=Ø      V6
         LA    R8,1(,R8)                                             V6
         MVC   Ø(2,R8),=C'),'              PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'ARCH TAPE DEALLOCATION TIME (MINUTES)'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> INBUFF    - INPUT BUFFER POOL SIZE             NOT ANYMORE IN V6
         AIF   (NOT D'LOGPIBPS).MAXARCH           IF V6 THEN DO ELSE V5
         MVC   ZPRMCL16(Ø7),=CLØ7'INBUFF='
         SR    R9,R9                       ZERO REGISTER
         L     R9,LOGPIBPS                 GET ZPARM VALUE
         CVD   R9,D                        CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø7(15),D           PACK TO ZONE NUMERIC
         OI    ZPRMCL16+21,X'FØ'           FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø7        MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN                DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø7(16),ZEROHOLD    MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE            FIND FIRST BLANK
         MVI   Ø(1),C','                   PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'ARCH TAPE DEALLOCATION TIME (MINUTES)'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.MAXARCH ANOP
*> MAXARCH   - MAX ARCHIVE ENTRIES IS BSDS
         MVC   ZPRMCL16(Ø8),=CLØ8'MAXARCH='
         SR    R9,R9                       ZERO REGISTER
         L     R9,LOGPARCL                 GET ZPARM VALUE
         CVD   R9,D                        CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),D           PACK TO ZONE NUMERIC
         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
         MVC   ZPRMCL4Ø,=CL32'MAX ARCHIVE ENTRIES IS BSDS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> MAXRTU    - MAXIMUM ARCHIVE READ TAPE UNITS
         MVC   ZPRMCL16(Ø7),=CLØ7'MAXRTU='
         SR    R9,R9                       ZERO REGISTER
         LH    R9,LOGPMRTU                 GET ZPARM VALUE
         CVD   R9,D                        CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø7(Ø7),D           PACK TO ZONE NUMERIC
         OI    ZPRMCL16+13,X'FØ'           FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø7        MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN                DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø7(16),ZEROHOLD    MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE            FIND FIRST BLANK
         MVI   Ø(1),C','                   PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'MAXIMUM ARCHIVE READ TAPE UNITS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> OFFLOAD   - ONLINE INITIATION OF THE OFFLOAD PROCESS
         TM    LOGOPT1,64
         BO    OUTBUFF
         MVC   ZPRMCL16(12),=CL12'OFFLOAD=NO, '
         MVC   ZPRMCL16+2Ø(3Ø),=CL3Ø'==> NOT FIT FOR PRODUCTION <==' V6
         TM    LOGOPT1,64
         BZ    *+1Ø
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> OUTBUFF   - OUTPUT BUFFER POOL SIZE
OUTBUFF  MVC   ZPRMCL16(Ø8),=CLØ8'OUTBUFF='
         SR    R9,R9                      ZERO REGISTER
         L     R9,LOGPOBPS                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE NUMERIC
         OI    ZPRMCL16+22,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
         MVC   ZPRMCL4Ø,=CL32'OUTPUT BUFFER FOR ACTIVE LOG'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> TWOACTV   -
         MVC   ZPRMCL16(12),=CL12'TWOACTV=NO, '
         TM    LOGOPT1,128
         BZ    *+1Ø
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'TWO ACTIVE LOG COPIES'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> TWOARCH   -
         MVC   ZPRMCL16(12),=CL12'TWOARCH=NO, '
         TM    LOGOPT2,128
         BZ    *+1Ø
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'TWO ARCHIVE COPIES'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> TWOBSDS   -
         MVC   ZPRMCL16(12),=CL12'TWOBSDS=NO, '
         TM    LOGOPT1,32
         BZ    *+1Ø
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'TWO BSDS DATASETS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ARC2FRST  - FOR RECOVERYSITE ALLOC SECOND ARCHIV FIRST            V6
         AIF   (NOT &REEV6).ARC21ST       IF HIGHER THAN V5 THEN     V6
         MVC   ZPRMCL16(12),=CL12'ARC2FRST=NO,'                      V6
         TM    LOGOPT2,64                                            V6
         BZ    *+1Ø                                                  V6
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'YES,'                           V6
         MVC   ZPRMCL4Ø,=CL32'ALLOC SECOND ARCHIVE AT RECOVERYSITE'  V6
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V6
```

```
.ARC21ST ANOP                                                        V6
*> WRTHRSH   -
         MVC   ZPRMCL16(Ø8),=CLØ8'WRTHRSH='
         SR    R9,R9                      ZERO REGISTER
         LH    R9,LOGPWRTH                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),D          PACK TO ZONE NUMERIC
         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 IN COL 72
         MVC   ZPRMCL4Ø,=CL32'OUTPUT BUFFER THRESHOLD VALUE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               DSN6SYSP                 '
*> FORMAT DSN6SYSP    ************************************************
         L     R7,LOADMPTR
         LA    RØ,4
         LA    R1,255(,R7)
         CLC   =CLØ8'DSN6SYSP',4(R7)
         BE    *+12
         BXLE  R7,RØ,*-1Ø
         B     ABEND195
         L     R7,Ø(,R7)
         USING DSN6SYSP,R7
         LR    R15,R7
         LA    RØ,1
         LA    R1,DSN6SYSP+L'SYSPLVLC-1
         DROP  R7
         USING DSN6SYSP,R15
         CLI   SYSPLVLC,C' '
         BL    ABEND1Ø5
         BXLE  R15,RØ,*-8
         DROP  R15
         USING DSN6SYSP,R7
*> FORMAT DSN6SYSP    ************************************************
         MVC   ZPRMCLØ5(Ø8),=CLØ8'DSN6SYSP'
*> AUDITST   - AUDIT TRACE START
         MVC   ZPRMCL16(Ø8),=CLØ8'AUDITST='
         MVC   WORKB32,SYSPAUDT           GET 32 BITS
         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   SYSPAUDB                   N. GO ON
         MVC   WORKCHAR(Ø2),=CLØ2'NO'     Y. SAY NO HERE
SYSPAUDB MVC   ZPRMCL16+Ø8(48),WORKCHAR
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         LA    RØ,2(,1)                   IF FLAGS EXTENT INTO TEXT
         C     RØ,=A(ZPRMCL4Ø)            THEN
         BH    *+1Ø                          DO NOT SET EXPLAINING TEXT
         MVC   ZPRMCL4Ø,=CL32'AUDIT TRACE START'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> BACKODUR  - NON DATA SHARING BACKOUT DURATION
         AIF   (NOT D'SYSPBDUR).BACKODR                              V6
         MVC   ZPRMCL16(Ø9),=CLØ9'BACKODUR='
         SR    R9,R9                                                 V6
         IC    R9,SYSPBDUR                NON DATA SHARING BACKOUT DURA
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'NON DATA SHARING BACKOUT DURATION'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.BACKODR ANOP                                                        V6
*> CONDBAT   - MAX NO. CONNECTED DBAT
         MVC   ZPRMCL16(Ø8),=CLØ8'CONDBAT='
         AIF   (&REEV6).CONDBTF                                      V6
         LH    R9,SYSPCDB                 GET NUMBER OF CONNECTED DBATS
         AGO   .CONDBAT                                              V6
.CONDBTF L     R9,SYSPCDB                                            V6
.CONDBAT CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'MAX NO. CONNECTED DBAT'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> CTHREAD   - MAX NO OF CONCURRENT THREADS
         MVC   ZPRMCL16(Ø8),=CLØ8'CTHREAD='
         LH    R9,SYSPCT                  GET CONCURRENT THD
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'MAX NO OF CONCURRENT THREADS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DBPROTCL  - DB2 CONNECTION DRDA
```

```
         AIF   (NOT D'SYSPDBPR).DBPROT                               V6
         MVC   ZPRMCL16(14),=CL14'DBPROTCL=DRDA,'
         CLI   SYSPDBPR,C'P'                                         V6
         BNE   *+1Ø                                                  V6
         MVC   ZPRMCL16+Ø9(Ø8),=CLØ8'PRIVATE,' DRDA PROT DEF PRIVATE
         MVC   ZPRMCL4Ø,=CL32'MAX NO OF CONCURRENT THREADS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.DBPROT  ANOP                                                        V6
*> DLDFREQ   - CHECKPOINTS PER LEVEL ID UPDATE
         MVC   ZPRMCL16(Ø8),=CLØ8'DLDFREQ='
         LH    R9,SYSPDFRQ                GET CONCURRENT THD
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'CHECKPOINTS PER LEVEL ID UPDATE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSSTIME   - TIME BETWEEN RESETTING OF DATASET STATS
         AIF   (NOT D'SYSPDTIM).DSSTIME                              V6
         MVC   ZPRMCL16(Ø8),=CLØ8'DSSTIME='
         LH    R9,SYSPDTIM                NUMBER OF 4K ELEMENTS
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'TIME BETWEEN RESET OF DATASET STATS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.DSSTIME ANOP                                                        V6
*> EXTSEC    - EXTENDED SECURITY                                     V5
         AIF   (NOT D'SYSPSCER).EXTSEC                               V5
         MVC   ZPRMCL16(1Ø),=CL1Ø'EXTSEC=NO,'                        V5
         CLI   SYSPSCER,C'Y'                                         V5
         BNE   *+1Ø                                                  V5
         MVC   ZPRMCL16+Ø7(Ø4),=CLØ4'YES,'                           V5
         MVC   ZPRMCL4Ø,=CL32'TIME BETWEEN RESET OF DATASET STATS'
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V5
.EXTSEC  ANOP                                                        V5
*> IDBACK    - MAX NO OF BACKGROUND IDS
         MVC   ZPRMCL16(Ø7),=CLØ7'IDBACK='
         LH    R9,SYSPIDB                 GET BACKGROUND IDS
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø7(Ø7),D          PACK TO ZONE
         OI    ZPRMCL16+13,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø7       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø7(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'MAX NO OF BACKGROUND IDS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> EXTRAREQ  - EXTRA DRDA QUERY BLOCKS REQUESTER
         AIF   (NOT D'SYSPEXBR).EXTRARQ                               V6
         MVC   ZPRMCL16(Ø9),=CLØ9'EXTRAREQ='
         LH    R9,SYSPEXBR                NUMBER OF EXTRA QUERZ BLOCKS
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'EXTRA DRDA QUERY BLOCKS REQUESTER'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.EXTRARQ ANOP                                                         V6
*> EXTRASRV  - EXTRA DRDA QUERY BLOCKS SERVER
         AIF   (NOT D'SYSPEXBS).EXTRASV                               V6
         MVC   ZPRMCL16(Ø9),=CLØ9'EXTRASRV='
         LH    R9,SYSPEXBS                EXTRA QUERY BLOCKS FOR SERVER
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'EXTRA DRDA QUERY BLOCKS SERVER'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.EXTRASV ANOP                                                         V6
*> IDFORE    - MAX NO OF FOREGROUND IDS
         MVC   ZPRMCL16(Ø7),=CLØ7'IDFORE='
         LH    R9,SYSPIDF                 GET FOREGROUND IDS
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø7(Ø7),D          PACK TO ZONE
         OI    ZPRMCL16+13,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø7       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø7(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'MAX NO OF FOREGROUND IDS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> LOGLOAD    - LOGLOAD VALUE CHECKPOINT FREQUENCY
         MVC   ZPRMCL16(Ø8),=CLØ8'LOGLOAD='
         SR    R9,R9                      ZERO REGISTER
         L     R9,SYSPLOGL                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE
         OI    ZPRMCL16+22,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
         MVC   ZPRMCL4Ø,=CL32'LOGLOAD VALUE CHECKPOINT FREQUENCY'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> MAXDBAT    - MAX NO OF ACTIVE REMOTE THREADS
         MVC   ZPRMCL16(Ø8),=CLØ8'MAXDBAT='
         LH    R9,SYSPRMT                 GET MAXIMUM ACTIVE REMOTE THD
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'MAX NO OF ACTIVE REMOTE THREADS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> MON        - MONITOR TRACING FLAG
         MVC   ZPRMCL16(Ø4),=CLØ4'MON='
         MVC   WORKB32,SYSPMON            GET MONITOR TRACING FLAGS
         BAS   R14,BIT16RTN               CONVERT FIRST 16 BITS TO NUM
         BAS   R14,BIT32RTN               CONVERT NEXT 16 BITS TO NUM
         CLI   WORKCHR1,C')'              IF ALL BITS ARE Ø
         BNE   *+1Ø                       N. GO ON
         MVC   WORKCHAR(Ø2),=CLØ2'NO'     Y. SAY NO
         MVC   ZPRMCL16+Ø4(48),WORKCHAR
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         LA    RØ,2(,1)                   IF FLAGS EXTENT INTO TEXT
         C     RØ,=A(ZPRMCL4Ø)            THEN
         BH    *+1Ø                          DO NOT SET EXPLAINING TEXT
         MVC   ZPRMCL4Ø,=CL32'MONITOR TRACING FLAGS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> MONSIZE    - MONITOR BUFFER SIZE
         MVC   ZPRMCL16(Ø8),=CLØ8'MONSIZE='
         SR    R9,R9                      ZERO REGISTER
         L     R9,SYSPMONS                GET MONITOR SIZE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(15),D          PACK TO ZONE
         OI    ZPRMCL16+22,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
         MVC   ZPRMCL4Ø,=CL32'MONITOR BUFFER SIZE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> PCLOSEN   - NUMBER OF CHECKPOINT FOR READ ONLY SWITCHING
         MVC   ZPRMCL16(Ø8),=CLØ8'PCLOSEN='
         LH    R9,SYSPFRQ                 GET CONCURRENT THD
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'CHECKPOINTS FOR READ ONLY SWITCHING'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> PCLOSET   - MINUTES TO PSEUDO-CLOSE READ ONLY SWITCHING
         MVC   ZPRMCL16(Ø8),=CLØ8'PCLOSET='
         LH    R9,SYSPTMR                 GET CONCURRENT THD
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'MINUTES TO PSEUDO-CLOSE READ ONLY'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> PTASKROL  - ROLL UP PARALLEL TASK ACCOUNTING                      V6
         AIF   (NOT D'SYSPFLAG).PTASKRL                              V6
         MVC   ZPRMCL16(12),=CL12'PTASKROL=NO, '                     V6
         TM    SYSPFLAG,B'ØØ1ØØØØØ'                                  V6
         BNO   *+1Ø                                                  V6
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'YES,'                           V6
         MVC   ZPRMCL4Ø,=CL32'ROLL UP PARALLEL TASK ACCOUNTING'      V5
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V6
.PTASKRL ANOP                                                        V6
*> RLF       - RESOURCE LIMIT FACILITY ENABLED
         MVC   ZPRMCL16(Ø8),=CLØ8'RLF=YES '
         AIF   (&REEV6).PRFLG1                                       V6
         TM    WRKPFLG1,B'ØØØØØØ1Ø'       BIT 7 ON
         AGO   .PRFLG1T                                              V6
.PRFLG1  TM    SYSPRLFR,128                                          V6
.PRFLG1T BO    *+1Ø                       Y.
```

```
         MVC   ZPRMCL16+Ø4(Ø3),=CLØ3'NO '
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'ENABLE RLF'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RLFTBL    - RESOURCE LIMIT FACILITY TABLE ID
         MVC   ZPRMCL16(Ø7),=CLØ7'RLFTBL='
         MVC   ZPRMCL16+Ø7(L'SYSPRLFT),SYSPRLFT GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'RESOURCE LIMIT FACILITY TABLE ID'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RLFAUTH   - RESOURCE LIMIT FACILITY
         MVC   ZPRMCL16(Ø8),=CLØ8'RLFAUTH='
         MVC   ZPRMCL16+Ø8(Ø8),SYSPRLFA   GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'RESOURCE LIMIT FACILITY'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RLFERR    - RESOURCE LIMIT FACILITY ERROR
         MVC   ZPRMCL16(Ø7),=CLØ7'RLFERR='
         L     R9,SYSPRLFN                GET LIMIT SU
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø7(15),D          PACK TO ZONE
         OI    ZPRMCL16+21,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø7       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø7(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TM    SYSPRLFR,32                IF NORUN FLAG              V6
         BZ    *+1Ø                          THEN IGNORE NUMBER AND  V6
         MVC   ZPRMCL16+Ø7(16),=CL16'NORUN' RLF=NORUN
         TM    SYSPRLFR,64                IF NOLIMIT FLAG            V6
         BZ    *+1Ø                          THEN IGNORE NUMBER AND  V6
         MVC   ZPRMCL16+Ø7(16),=CL16'NOLIMIT' RLF=NOLIMIT
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'RLF SU OVERRUN ACTION'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> ROUTCDE   - SYSTEM MESSAGE ROUTING CODE
         MVC   ZPRMCL16(Ø8),=CLØ8'ROUTCDE='
         MVC   WORKB16,SYSPSMRC           GET 32 BITS
         BAS   R14,BIT16RTN               CONVERT FIRST 16 BITS TO NUM
         CLI   WORKCHR1,C')'              IF ) MEANS ALL BITS ARE Ø
         BNE   *+1Ø                       N. GO ON
         MVC   WORKCHAR(Ø2),=CLØ2'NO'     Y. SAY NO HERE
         MVC   ZPRMCL16+Ø8(48),WORKCHAR
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         LA    RØ,2(,1)                   IF FLAGS EXTENT INTO TEXT
         C     RØ,=A(ZPRMCL4Ø)            THEN
         BH    *+1Ø                          DO NOT SET EXPLAINING TEXT
         MVC   ZPRMCL4Ø,=CL32'SYSTEM MESSAGE ROUTING CODE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SMFACCT   - SMF ACCOUNTING FLAGS
         MVC   ZPRMCL16(Ø8),=CLØ8'SMFACCT='
         MVC   WORKB32,SYSPSMFA           GET SMF ACCOUNTING 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   *+1Ø                       N. GO ON
         MVC   WORKCHAR(Ø2),=CLØ2'NO'     Y. SAY NO HERE
         MVC   ZPRMCL16+Ø8(48),WORKCHAR
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         LA    RØ,2(,1)                   IF FLAGS EXTENT INTO TEXT
         C     RØ,=A(ZPRMCL4Ø)            THEN
         BH    *+1Ø                          DO NOT SET EXPLAINING TEXT
         MVC   ZPRMCL4Ø,=CL32'SMF ACCOUNTING FLAGS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> SMFSTAT   - SMF STATISTICS FLAGS
         MVC   ZPRMCL16(Ø8),=CLØ8'SMFSTAT='
         MVC   WORKB32,SYSPSMFS           GET SMF STAATISTICS 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   *+1Ø                       N. GO ON
         MVC   WORKCHAR(Ø2),=CLØ2'NO'     Y. SAY NO HERE
         MVC   ZPRMCL16+Ø8(48),WORKCHAR
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         LA    RØ,2(,1)                   IF FLAGS EXTENT INTO TEXT
         C     RØ,=A(ZPRMCL4Ø)            THEN
         BH    *+1Ø                          DO NOT SET EXPLAINING TEXT
         MVC   ZPRMCL4Ø,=CL32'SMF STATISTICS FLAGS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> STATIME   - STATISTICS TIME
         MVC   ZPRMCL16(Ø8),=CLØ8'STATIME='
         SR    R9,R9                      ZERO REGISTER
         LH    R9,SYSPSTIM                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'STATISTICS TIME'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> STORPROC  - STORED PROCEDURE MVS NAME
```

```
         MVC   ZPRMCL16(Ø9),=CLØ9'STORPROC='
         MVC   ZPRMCL16+Ø9(Ø8),SYSPSPPN   GET PARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'STORED PROCEDURE MVS NAME'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> STORMXAB  - ALLOWABLE ABENDS FOR STORED PROCEDURES
         MVC   ZPRMCL16(Ø9),=CLØ9'STORMXAB='
         LH    R9,SYSPSPAB                GET ALLOWABLE ABENDS FOR STP
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'ALLOWABLE ABENDS FOR STORED PROC'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> STORTIME  - TIME TO WAIT BEFORE SQL FAILS
         MVC   ZPRMCL16(Ø9),=CLØ9'STORTIME='
         LH    R9,SYSPSPTO                GET STORPROC TIMEOUT VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'TIME TO WAIT BEFORE SQL FAILS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> URCHKTH   - UR CHECKPOINT THRESHOLD                               V5
         AIF   (NOT D'SYSPURCK).URCKTH                               V5
         MVC   ZPRMCL16(Ø8),=CLØ8'URCHKTH='                          V5
         SR    R9,R9                                                 V5
         IC    R9,SYSPURCK                UR CHECKPOINT VALUE        V5
         CVD   R9,D                       CONVERT DECIMAL            V5
         UNPK  ZPRMCL16+Ø8(Ø3),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
         MVC   ZPRMCL4Ø,=CL32'UR CHECKPOINT THRESHOLD'
         BAS   R14,ZWRTRTN             DO PRINT LINE                 V5
.URCKTH  ANOP                                                        V5
*> TRACLOC   - 4K ELEMENTS IN LOCAL TRACE TRABLE
         MVC   ZPRMCL16(Ø8),=CLØ8'TRACLOC='
         LH    R9,SYSPTLSZ                NUMBER OF 4K ELEMENTS
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'4K ELEMENTS IN LOCAL TRACE TRABLE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> TRACSTR   - MONITOR TRACING FLAG
         MVC   ZPRMCL16(Ø8),=CLØ8'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),=CLØ2'NO'     Y. SAY NO HERE
         MVC   ZPRMCL16+Ø8(48),WORKCHAR
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         LA    RØ,2(,1)                   IF FLAGS EXTENT INTO TEXT
         C     RØ,=A(ZPRMCL4Ø)            THEN
         BH    *+1Ø                          DO NOT SET EXPLAINING TEXT
         MVC   ZPRMCL4Ø,=CL32'MONITOR TRACING FLAG'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> IDXBPOOL  - DEF BP FOR INDEXES
         AIF   (NOT D'SYSPIXPL).IDXBPL                               V6
         MVC   ZPRMCL16(Ø9),=CLØ9'IDXBPOOL='
         MVC   ZPRMCL16+Ø9(Ø4),SYSPIXPL   GET BPOOL NAME
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'DEFAULT BP FOR INDEXES'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.IDXBPL  ANOP                                                        V6
*> TBSBPOOL  - DEF BP FOR TABELSPACES
         AIF   (NOT D'SYSPTBPL).TBSBPL                               V6
         MVC   ZPRMCL16(Ø9),=CLØ9'TBSBPOOL='
         MVC   ZPRMCL16+Ø9(Ø4),SYSPTBPL   GET BPOOL NAME
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'DEFAULT BP FOR TABLESPACES'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.TBSBPL  ANOP                                                        V6
*> LOGAPSTG  - FAST LOG APPLY STORAGE IN MB
         AIF   (NOT D'SYSPFLBZ).LOGAPST                              V6
         MVC   ZPRMCL16(Ø9),=CLØ9'LOGAPSTG='
         SR    R9,R9                                                 V6
         IC    R9,SYSPFLBZ                MB STORAGE FOR FAST LOG APPLY
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'FAST LOG APPLY STORAGE IN MB'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.LOGAPST ANOP                                                        V6
*> LBACKOUT  - NON DATA SHARING BACKOUT AUTO/YES/NO
         AIF   (NOT D'SYSPLMBO).LBACKOU                              V6
         MVC   ZPRMCL16(Ø9),=CLØ9'LBACKOUT='
         MVC   ZPRMCL16+Ø9(Ø4),SYSPLMBO   RESTART BACKOUT AUTO/YES/NO
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'RESTART BACKOUT OPTION NON DS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.LBACKOU ANOP                                                        V6
*> LOBVALA   - KB FOR LOB VALUES PER AGENT
         AIF   (NOT D'SYSPLVA).LOBVALA                               V6
         MVC   ZPRMCL16(Ø8),=CLØ8'LOBVALA='
         L     R9,SYSPLVA                 KB FOR LOB PER AGENT
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'KB FOR LOB VALUES PER AGENT'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.LOBVALA ANOP                                                        V6
*> LOBVALS   - MB FOR LOB VALUES PER SYSTEM
         AIF   (NOT D'SYSPLVS).LOBVALS                               V6
         MVC   ZPRMCL16(Ø8),=CLØ8'LOBVALS='
         L     R9,SYSPLVS                 MB FOR LOB PER SYSTEM
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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
         MVC   ZPRMCL4Ø,=CL32'MB FOR LOB VALUES PER SYSTEM'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.LOBVALS ANOP                                                        V6
```

```
*> WLMENV    - DEFAULT WLM ENVIRONMENT NAME
         AIF   (NOT D'SYSPWLME).WLMENV                               V6
         MVC   ZPRMCL16(Ø7),=CLØ7'WLMENV='
         MVC   ZPRMCL16+Ø7(L'SYSPWLME),SYSPWLME   GET WLM ENVIRONMENT
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'DEFAULT WLM ENVIRONMENT NAME'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.WLMENV  ANOP                                                        V6
*> TRACTBL   - NO OF 4K SEGMENTS IN LOCAL TRACETBL
         MVC   ZPRMCL16(Ø8),=CLØ8'TRACTBL='
         LH    R9,SYSPTRSZ                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø8(Ø7),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 - LAST CARD
         MVC   ZPRMCL4Ø,=CL32'4K SEGMENTS IN LOCAL TRACETBL'
         BAS   R14,ZWRTRTN             DO PRINT LINE
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               DSN6FAC                  '
*> FORMAT DSN6FAC     ************************************************
         USING DSN6FAC,R7
         L     R7,LOADMPTR
         LA    RØ,4
         LA    R1,255(,R7)
         CLC   =CLØ8'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),=CLØ7'DSN6FAC'
*> DDF       - DDF FLAG BYTE FOR STARTUP
         MVC   ZPRMCL16(Ø7),=CLØ7'DDF=NO,'
         CLI   FACSTART,C'A'
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø4(Ø5),=CLØ5'AUTO,'
         CLI   FACSTART,C'C'
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø4(Ø8),=CLØ8'COMMAND,'
         MVC   ZPRMCL4Ø,=CL32'DDF STARTUP'
         BAS   R14,ZWRTRTN             DO PRINT LINE
```

```
*> CMTSTAT   - DDF THREAD STATUS
         MVC   ZPRMCL16(Ø8),=CLØ8'CMTSTAT='
         MVC   ZPRMCL16+Ø8(Ø8),FACCMST    GET ZPARM VALUE
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'DDF THREAD STATUS'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> IDTHTOIN  - IDLE THREAD TIMEOUT INTERVAL
         MVC   ZPRMCL16(Ø9),=CLØ9'IDTHTOIN='
         LH    R9,FACTOIN                 GET TIMEOUT VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZEROHOLD(Ø7),D             PACK TO ZONE
         OI    ZEROHOLD+Ø6,X'FØ'          FIX LAST DIGIT
         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
         MVC   ZPRMCL4Ø,=CL32'DDF IDLE THREAD TIMEOUT'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> RESYNC    - MINUTES BETWEEN RESYNC PERIODS
         MVC   ZPRMCL16(Ø7),=CLØ7'RESYNC='
         LH    R9,FACRESYC                GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø7(Ø7),D          PACK TO ZONE
         OI    ZPRMCL16+13,X'FØ'          FIX LAST DIGIT
         MVC   ZEROHOLD,ZPRMCL16+Ø7       MOVE NUMBER IN HOLD AREA
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø7(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'DDF RESYNC PERIOD LENGTH (MIN)'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> POOLINAC  - SECONDS AN DDF INACTIVE THREAD CAN REMAIN IN POOL
         AIF   (NOT D'FACINAC).TCPKPAL                                V6
         MVC   ZPRMCL16(Ø9),=CLØ9'POOLINAC='
         LH    R9,FACINAC                 GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZEROHOLD(Ø7),D             PACK TO ZONE
         OI    ZEROHOLD+Ø6,C'Ø'           FIX LAST DIGIT
         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
         MVC   ZPRMCL4Ø,=CL32'DDF INACTIVE POOL TIME (SEC)'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.TCPKPAL ANOP                                                         V6
*> TCPKPALV  - TCP/IP STACK KEEP ALIVE TIME
         AIF   (NOT D'FACTCKA).TCALVER                                V6
         MVC   ZPRMCL16(Ø9),=CLØ9'TCPKPALV='
         LH    R9,FACTCKA                 GET ZPARM VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZPRMCL16+Ø9(Ø7),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
         LTR   R9,R9                                                 V6
         BNZ   *+1Ø                                                  V6
         MVC   ZPRMCL16+Ø9(16),=CL16'ENABLE'                         V6
         BNM   *+1Ø                                                  V6
         MVC   ZPRMCL16+Ø9(16),=CL16'DISABLE'                        V6
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'TCP/IP STACK KEEP ALIVE TIME'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.TCALVER ANOP
*> TCPALVER  - TCP/IP ALREADY VERIFIED USERID
         AIF   (NOT D'FACMISC).MAXTYP1                               V6
         MVC   ZPRMCL16(12),=CL12'TCPALVER=NO,'
         TM    FACMISC,B'1ØØØØØØØ'        TCP USERID ALREADY VERIF   V6
         BZ    *+1Ø                                                  V6
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'YES,'                           V6
         MVC   ZPRMCL4Ø,=CL32'TCP/IP USERID ALREADY VERIFIED'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.MAXTYP1 ANOP
*> MAXTYPE1  - MAXIMUM NUMBER OF INACTIVE TYPE1 DDF CONNECTIONS      V6
         AIF   (NOT D'FACMAX1).RLFERRD                               V6
         MVC   ZPRMCL16(Ø9),=CLØ9'MAXTYPE1='                         V6
         L     R9,FACMAX1                 GET ZPARM VALUE            V6
         CVD   R9,D                       CONVERT DECIMAL            V6
         UNPK  ZPRMCL16+Ø9(Ø7),D          PACK TO ZONE               V6
         OI    ZPRMCL16+15,X'FØ'          FIX LAST DIGIT             V6
         MVC   ZEROHOLD,ZPRMCL16+Ø9       MOVE NUMBER IN HOLD AREA   V6
         BAS   R14,DZERORTN               DROP LEADING ZEROS         V6
         MVC   ZPRMCL16+Ø9(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK V6
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK           V6
         MVI   Ø(1),C','                  PLUG COMMA HERE            V6
         MVC   ZPRMCL4Ø,=CL32'MAXIMUM NUMBER OF INACT TYP1 DDF'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.RLFERRD ANOP                                                        V6
*> RLFERRD   - LIMIT OF CPU SECONDS
         MVC   ZPRMCL16(16),=CL16'RLFERRD=NOLIMIT '
         TM    FACRLFER,128               NOLIMIT
         BO    DDFNONU
         MVC   ZPRMCL16(16),=CL16'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
         MVC   ZPRMCL4Ø,=CL32'RLF ERROR LIMIT (CPU SECONDS)'
         BAS   R14,WRITRTN             DO PRINT LINE
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               DSN6GRP                 '
*> FORMAT DSN6GRP     ************************************************
         USING DSN6GRP,R7
         L     R7,LOADMPTR
         LA    RØ,4
         LA    R1,255(,R7)
         CLC   =CLØ8'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   ZPRMLINE(Ø2),=CLØ2'* '                                V6
         MVC   ZTITNAME&A.,=CL8'DSN6GRP'                             V6
         MVC   ZTITLVL&A.,GRPLVL                                     V6
         BAS   R14,ZWRTRTN                                           V6
         MVC   ZPRMCLØ5(22),=CL22'DSN6GRP    DSHARE=YES,'
         TM    GRPDSHR,128
         BO    *+1Ø
         MVC   ZPRMCLØ5(22),=CL22'DSN6GRP    DSHARE=NO, '
         MVC   ZPRMCL4Ø,=CL32'DATASHARING DEFINITION'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> GROUPNAME - DATA SHARING
         MVC   ZPRMCL16(Ø8),=CLØ8'GRPNAME='
         MVC   ZPRMCL16+Ø8(Ø8),GRPNAME
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'DB2 GROUPNAME'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> COORDNTR  - DATA SHARING COORDINATOR FOR QUERY PARALLELISM
         AIF   (NOT D'GRPCOOR).MEMBNM                                V6
         MVC   ZPRMCL16(12),=CL12'COORDNTR=NO,'
         CLI   GRPCOOR,C'Y'                                          V6
         BNE   *+1Ø                                                  V6
         MVC   ZPRMCL16+Ø9(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'DS COORD. FOR QUERY PARALLELISM'
         BAS   R14,ZWRTRTN             DO PRINT LINE
```

```
*> ASSIST    - DATA SHARING ASSISTANT FOR QUERY PARALLELISM
         MVC   ZPRMCL16(1Ø),=CL1Ø'ASSIST=NO,'
         CLI   GRPASST,C'Y'                                          V6
         BNE   *+1Ø                                                  V6
         MVC   ZPRMCL16+Ø7(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'DS ASSIST FOR QUERY PARALLELISM'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.MEMBNM  ANOP                                                        V6
*> MEMBNAME  - DATA SHARING
         MVC   ZPRMCL16(Ø9),=CLØ9'MEMBNAME='
         MVC   ZPRMCL16+Ø9(Ø8),GRPMNAME
         MVC   ZPRMCL4Ø,=CL32'DS MEMBER NAME'
         MVI   ZPRMCL72,C' '              INDICATE LAST CARD
         BAS   R14,WRITRTN             DO PRINT LINE
         MVC   ZPRMCLØ5(11),=CL11'AGO   .EXIT'
         BAS   R14,WRITRTN             DO PRINT LINE
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               DSNHDECM LOAD            '
         BAS   R14,DECMINIT               DO INITIALIZE VALUES
         BAS   R14,DECMHDR                DO PRINT HEADER LINES
******** M A I N L I N E    FOR   D S N H D E C M
** LOAD DSNHDECP IN VIRTUAL STORAGE
         MVC   LOADNAME,=CLØ8'DSNHDECP'                              V6
         LOAD  EPLOC=LOADNAME,LOADPT=LOADMPTR,DCB=DSNDECP
         LTR   R15,15
         BNZ   ABEND2ØØ
         LR    R15,RØ
         LA    R15,Ø(,R15)
         ST    R15,LOADMPTR
         LR    R7,R15
         LA    R1,Ø(,R1)                REMOVE HIGH ORDER IN LENGTH
         ST    R1,LOADMPTR+4           SAVE LENGTH OF DSNHDECP
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               ANALYSE DSNHDECP             '
         L     R7,LOADMPTR
         USING DSNHDECP,R7                ESTABLISH ADDRESSABILITY
         L     R2,=A(DSNHDECP)         SECTION TO BE ANALYSED
         CLC   DECPID,DECPID-DSNHDECP(R2)
         BNE   ABEND2Ø1
         CLC   DECPEYE,DECPEYE-DSNHDECP(R2)
         BNE   ABEND2Ø2                SECTION DSNMDECP NOT FOUND
** DISPLAY ZPARM NAME AND ASSEMBLY DATE
         MVC   ZPRMLINE,ZPRMTITL                                     V6
         BAS   R14,WRITRTN                                           V6
         MVC   ZPRMLINE,REELINE                                      V6
         BAS   R14,WRITRTN                                           V6
         L     R2,=A(DSN6SPRM)         SECTION TO BE ANALYSED
** DISPLAY MDECP NAME AND VERSION / MACRO CHANGE LEVEL
         MVC   ZPRMLINE(Ø2),=CLØ2'* '                                V6
         MVC   ZTITNAME&A.,LOADNAME       DECP  LOAD MODULE NAME
         MVC   ZTITLVL&A.,DECPLVL         LEVEL COMPILED FOR
         MVC   ZTITLVLC&A.,DECPREL
         BAS   R14,ZWRTRTN             DO PRINT LINE
         MVC   ZPRMLINE,=(&ZPRMLNE)C'-'
         MVC   ZPRMLINE(Ø2),=CLØ2'* '
         MVI   ZPRMCL72,C' '
         MVC   ZPRMCL73,=CLØ8' '
         BAS   R14,ZWRTRTN
         MVC   ZPRMCLØ5(Ø8),=CLØ8'DSNHDECM'
*> DSNHDECM    CHARSET=ALPHANUM
         MVC   ZPRMCL16(Ø8),=CLØ8'CHARSET='
         MVC   ZPRMCL16+Ø8(L'DECPCHAR),DECPCHAR
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'DEFAULT SUBSYSTEM CHARACTER SET'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    COMPAT=OFF                 SERVICEBILITY PARAMETER
         AIF   (NOT D'DECPCMPT).COMPAT                            V6
         MVC   ZPRMCL16(1Ø),=CL1Ø'COMPAT=ON,'
         CLI   DECPCMPT,OFF
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø7(Ø4),=CLØ4'OFF,'
         MVC   ZPRMCL4Ø,=CL32'SERVICEBILITY PARAMETER'
         BAS   R14,ZWRTRTN             DO PRINT LINE
.COMPAT  ANOP                                                     V6
*> DSNHDECM    ASCCSID=85Ø                ASCII SINGLE BYTE CCSID
         MVC   ZPRMCL16(Ø8),=CLØ8'ASCCSID='
         SR    R9,R9
         ICM   R9,B'ØØ11',DECPASID        GET CCSIC FOR SINGLE BYTE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZEROHOLD(Ø7),D             PACK TO ZONE
         OI    ZEROHOLD+Ø6,C'Ø'           FIX LAST DIGIT
         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
         MVC   ZPRMCL4Ø,=CL32'ASCII SINGLE BYTE CHARSET ID'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    AMCCSID=65534              ASCII MIXED BYTE CCSID
         MVC   ZPRMCL16(Ø8),=CLØ8'AMCCSID='
         SR    R9,R9
         ICM   R9,B'ØØ11',DECPAMID        GET CCSIC FOR DOUBLE BYTE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZEROHOLD(Ø7),D             PACK TO ZONE
         OI    ZEROHOLD+Ø6,C'Ø'           FIX LAST DIGIT
         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
         MVC   ZPRMCL4Ø,=CL32'ASCII MIXED BYTE CHARSET ID'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    AGCCSID=65534              ASCII DOUBLE BYTE CCSID
         MVC   ZPRMCL16(Ø8),=CLØ8'AGCCSID='
         SR    R9,R9
         ICM   R9,B'ØØ11',DECPAGID        GET CCSIC FOR SINGLE BYTE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZEROHOLD(Ø7),D             PACK TO ZONE
         OI    ZEROHOLD+Ø6,C'Ø'           FIX LAST DIGIT
         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
         MVC   ZPRMCL4Ø,=CL32'ASCII DOUBLE BYTE CHARSET ID'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    SCCSID=273                 CCSID SINGLE BYTE
         MVC   ZPRMCL16(Ø7),=CLØ7'SCCSID='
         SR    R9,R9
         ICM   R9,B'ØØ11',DECPSID         GET CCSIC FOR SINGLE BYTE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZEROHOLD(Ø7),D             PACK TO ZONE
         OI    ZEROHOLD+Ø6,C'Ø'           FIX LAST DIGIT
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø7(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'EBCDIC SINGLE BYTE CHARSET ID'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    MCCSID=65534
         MVC   ZPRMCL16(Ø7),=CLØ7'MCCSID='
         SR    R9,R9
         ICM   R9,B'ØØ11',DECPMID         GET CCSIC FOR MIXED BYTE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZEROHOLD(Ø7),D             PACK TO ZONE
         OI    ZEROHOLD+Ø6,C'Ø'           FIX LAST DIGIT
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø7(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'EBCDIC MIXED BYTE CHARSET ID'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    GCCSID=65534
         MVC   ZPRMCL16(Ø7),=CLØ7'GCCSID='
         SR    R9,R9
         ICM   R9,B'ØØ11',DECPGID         GET CCSIC FOR DOUBLE BYTE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZEROHOLD(Ø7),D             PACK TO ZONE
         OI    ZEROHOLD+Ø6,C'Ø'           FIX LAST DIGIT
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø7(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                    PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'EBCDIC DOUBLE BYTE CHARSET ID'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    ENSCHEME=EBCDIC
         MVC   ZPRMCL16(15),=CL15'ENSCHEME=ASCII,'
         CLI   DECPENSC,EBCDIC
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø9(Ø7),=CLØ7'EBCDIC,'
         MVC   ZPRMCL4Ø,=CL32'DEFAULT ENCODING SCHEME'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    DATE=EUR
         MVC   ZPRMCL16(Ø5),=CLØ5'DATE='
         MVC   ZPRMCL16+Ø5(L'DECPDATE),DECPDATE
         TRT   ZPRMCL16,TRTABLE             FIND FIRST BLANK
         MVI   Ø(1),C','                    PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'DEFAULT DATE FORMAT'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    DATELEN=Ø                    DEFAULT DATE LENGTH
         MVC   ZPRMCL16(Ø8),=CLØ8'DATELEN='
         SR    R9,R9
         IC    R9,DECPDATL                  GET DATE LENGTH VALUE
         CVD   R9,D                         CONVERT DECIMAL
         UNPK  ZEROHOLD(Ø7),D               PACK TO ZONE
         OI    ZEROHOLD+Ø6,C'Ø'             FIX LAST DIGIT
         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
         MVC   ZPRMCL4Ø,=CL32'DEFAULT DATE LENGTH'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    DECARTH=DEC15                DECIMAL ARITHMETIC 15/31
         MVC   ZPRMCL16(14),=CL14'DECARTH=DEC15,'
         CLI   DECPARTH,DEC31
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø8(Ø6),=CLØ6'DEC31,'
         MVC   ZPRMCL4Ø,=CL32'DEFAULT DECIMAL PRECISION'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    DECIMAL=PERIOD COBOL AND DYNAMIC SQL COMMA SETTING
         MVC   ZPRMCL16(14),=CL14'DECIMAL=COMMA,'
         CLI   DECPDE,PERIOD
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø8(Ø7),=CLØ7'PERIOD,'
         MVC   ZPRMCL4Ø,=CL32'DEFAULT DECIMAL PERIOD'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    DEFLANG=COB2                 DEFAULT LANGUAGE
         MVC   ZPRMCL16(Ø8),=CLØ8'DEFLANG='
         MVC   ZPRMCL16+Ø8(L'DECPLANG),DECPLANG
         TRT   ZPRMCL16,TRTABLE             FIND FIRST BLANK
         MVI   Ø(1),C','                    PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'DEFAULT LANGUAGE'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    DELIM=APOST                HOST DELIMITER OPTION
         MVC   ZPRMCL16(12),=CL12'DELIM=APOST,'
         CLI   DECPDL,QUOTE
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø6(Ø6),=CLØ6'QUOTE,'
         MVC   ZPRMCL4Ø,=CL32'DEFAULT DELIMITER'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    MIXED=NO                   MIXED GRAPHIC DEFAULT
         MVC   ZPRMCL16(Ø9),=CLØ9'MIXED=NO,'
         CLI   DECPGRA,YES
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø6(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'DEFAULT MIXED GRAPHIC'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    SQLDELI=APOST              SQL DELIMITER OPTION
         MVC   ZPRMCL16(14),=CL14'SQLDELI=APOST,'
         CLI   DECPSDL,QUOTE
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø8(Ø6),=CLØ6'QUOTE,'
         MVC   ZPRMCL4Ø,=CL32'DEFAULT SQL DELIMITER'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    DSQLDELI=APOST
         MVC   ZPRMCL16(15),=CL15'DSQLDELI=APOST,'
         CLI   DECPDSD,QUOTE
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø9(Ø6),=CLØ6'QUOTE,'
         MVC   ZPRMCL4Ø,=CL32'DEFAULT DDF SQL DELIMITER'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    SSID=DB2T                  SUBSYSTEM ID
         MVC   ZPRMCL16(Ø5),=CLØ5'SSID='
         MVC   ZPRMCL16+Ø5(L'DECPSSID),DECPSSID
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'SUBSYSTEM ID'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    DYNRULS=YES
         MVC   ZPRMCL16(11),=CL11'DYNRULS=NO,'
         CLI   DECPDRUL,YES
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø8(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'DYNAMIC RULES FROM PRECOMPILER'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    LC_CTYPE=                  UPPER, LOWER, TRANSLATE OE
         AIF   (NOT D'DECPLCPT).LCCTYPE                              V6
         MVC   ZPRMCL16(Ø9),=CLØ9'LC_CTYPE='
         MVC   ZPRMCL16+Ø9(L'DECPLCTP),DECPLCTP
         CLI   ZPRMCL72,C' '                                         V6
         BE    *+2Ø                                                  V6
         MVI   ZPRMCL72,C'X'                                         V6
         MVC   ZPRMCL73,=CLØ8' '                                     V6
         BAS   R14,WRITRTN                                           V6
         CLI   DECPLCTP+72-16-Ø9,C' '                                V6
         BE    *+14                                                  V6
         MVC   ZPRMCL16(5Ø-72+16+Ø9),DECPLCTP+72-16-Ø9               V6
         BAS   R14,ZWRTRTN             DO PRINT LINE
.LCCTYPE ANOP                                                        V6
*> DSNHDECM    STDSQL=YES
         MVC   ZPRMCL16(1Ø),=CL1Ø'STDSQL=NO,'
         CLI   DECPSQL,DECPSA86
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø7(Ø3),=CLØ3'86,'
         CLI   DECPSQL,DECPSYES
         BNE   *+1Ø
         MVC   ZPRMCL16+Ø7(Ø4),=CLØ4'YES,'
         MVC   ZPRMCL4Ø,=CL32'USE 86 STD SQL, NOT DB2 SQL'
         BAS   R14,ZWRTRTN             DO PRINT LINE
*> DSNHDECM    TIME=JIS
         MVC   ZPRMCL16(Ø5),=CLØ5'TIME='
         MVC   ZPRMCL16+Ø5(L'DECPTIME),DECPTIME
         TRT   ZPRMCL16,TRTABLE           FIND FIRST BLANK
         MVI   Ø(1),C','                  PLUG COMMA HERE
         MVC   ZPRMCL4Ø,=CL32'TIME FORMAT'
         BAS   R14,WRITRTN             DO PRINT LINE
*> DSNHDECM    TIMELEN=Ø                  TIME LEGTH FORMAT DEFAULT
         MVC   ZPRMCL16(Ø8),=CLØ8'TIMELEN='
         SR    R9,R9
         IC    R9,DECPTIML                GET DATE LENGTH VALUE
         CVD   R9,D                       CONVERT DECIMAL
         UNPK  ZEROHOLD(Ø7),D             PACK TO ZONE
         OI    ZEROHOLD+Ø6,C'Ø'           FIX LAST DIGIT
         BAS   R14,DZERORTN               DROP LEADING ZEROS
         MVC   ZPRMCL16+Ø8(16),ZEROHOLD   MOVE TRUNCATED NUMBER BACK
         MVC   ZPRMCL4Ø,=CL32'TIME LENGTH'
         BAS   R14,WRITRTN
         MVC   ZPRMLINE(2Ø),=CL2Ø'.EXIT END   DSN6SPRM'
         BAS   R14,WRITRTN             DO PRINT LINE
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               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)
         SR    R15,R15
         BR    R14
RETCOD12 DS    ØH
```

```
         L     R13,SAVEAREA+4
         LM    R14,R12,12(R13)
         LA    R15,12
         BR    R14
RETCOD16 DS    ØH
         L     R13,SAVEAREA+4
         LM    R14,R12,12(R13)
         LA    R15,16
         BR    R14
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               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 FOR DSNZPARM
INITRTN  DS    ØH
         ST    R14,INITSAVE
         OPEN  (SYSPUNCH,OUTPUT,DSNZPRM)
         OPEN  (SNAPDUMP,OUTPUT)
         L     R14,INITSAVE
         BR    R14
** DECPINIT  - INITIALIZE VALUES FOR DSNHDECP
DECMINIT DS    ØH
         ST    R14,INITSAVE
         OPEN  DSNDECP
         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   LOADNAME,=CLØ8' '
         MVC   Ø+LOADNAME,2(R6)        GET PARMLIST VAL
         EX    R1,*-6                  AND FILL WITH BLANKS TO LENGTH 8
         L     R14,GETPSAVE
         BR    R14
** DECMHDR   - PRINT HEADER LINES FOR **** DSNHDECP ****
DECMHDR  DS    ØH
         ST    R14,HDRLSAVE
         MVC   ZPRMLINE,=(&ZPRMLNE)C'='
         MVI   ZPRMCL72,C' '
         MVC   ZPRMCL73,=CLØ8' '
         MVC   ZPRMLINE(Ø2),=CLØ2'* '
         BAS   R14,WRITRTN
         MVC   ZPRMLINE,STITLEL2
         BAS   R14,WRITRTN
         L     R14,HDRLSAVE
         BR    R14
** HDRLRTN   - PRINT HEADER LINES FOR DSNZPARM
HDRLRTN  DS    ØH
         ST    R14,HDRLSAVE
         MVC   ZPRMLINE,=(&ZPRMLNE)C'='
         MVI   ZPRMCL72,C' '
         MVC   ZPRMCL73,=CLØ8' '
         MVC   ZPRMLINE(Ø2),=CLØ2'* '
         BAS   R14,WRITRTN
         MVC   ZPRMLINE,STITLEL1
         BAS   R14,WRITRTN
         L     R14,HDRLSAVE
         BR    R14
** CLOSRTN   - CLOSE FILES
CLOSRTN  DS    ØH
         ST    R14,CLOSSAVE
         CLOSE (SYSPUNCH,,DSNZPRM,,DSNDECP)
         CLOSE (SNAPDUMP)
         L     R14,CLOSSAVE
         BR    R14
** WRITRTN   - WRITE TO SYSPUNCH
WRITRTN  DS    ØH
         ST    R14,WRITSAVE
         PUT   SYSPUNCH,ZPRMLINE
         MVC   ZPRMLINE,=(&ZPRMLNE)C' '
         L     R14,WRITSAVE
         BR    R14
** ZWRTRTN   - WRITE TO SYSPUNCH WITH CONTINUATION MARK
ZWRTRTN  DS    ØH
         ST    R14,ZWRTSAVE
         PUT   SYSPUNCH,ZPRMLINE
         MVC   ZPRMLINE,=(&ZPRMLNE)C' '
         MVI   ZPRMCL72,C'X'
         L     R14,ZWRTSAVE
         BR    R14
*> SET LIBRARY LEVEL IN HEADER INFORMATION ZTITLVL
LVLC     SAVE  (14,12)                                               V6
         L     R7,LOADMPTR
         LA    RØ,4
         LA    R1,255(,R7)
         CLC   =CLØ8'DSN6SYSP',4(R7)
         BE    *+12
         BXLE  R7,RØ,*-1Ø
         B     ABEND195
         L     R7,Ø(,R7)
         USING DSN6SYSP,R7
         LR    R15,R7
         LA    RØ,1
         LA    R1,DSN6SYSP+L'SYSPLVLC-1
         DROP  R7
         USING DSN6SYSP,R15
         CLI   SYSPLVLC,C' '
         BL    ABEND1Ø5
         BXLE  R15,RØ,*-8
         DROP  R15
         USING DSN6SYSP,R7
         MVC   ZTITLVLC&A.,SYSPLVLC       LEVEL COMPILED FOR
         DROP  R7                                                    V6
         RETURN (14,12)                                              V6
** DZERORTN  - DROP LEADING ZEROES
DZERORTN DS    ØH
         ST    R14,DZERSAVE
DZERO1ØØ CLI   ZEROHOLD,C'Ø'                IF ZERO
         BNE   DZEREXIT                     N.  EXIT WE ARE DONE
         MVC   ZEROHOLD,ZEROHOLD+1          Y.  SHIFT 1BYTE LEFT
         B     DZERO1ØØ                     CHECK NEXT BYTE
DZEREXIT EQU   *
         BH    *+8                                                   V6
         MVI   ZEROHOLD,C'Ø'                                         V6
         L     R14,DZERSAVE
         BR    R14
** BIT16RTN  - TEST BIT 16 AND CONVERT TO NUMERIC
BIT16RTN DS    ØH
         ST    R14,BIT16SAV
         MVC   WORKCHAR,=CL56' '            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),=CLØ2'1,'            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),=CLØ2'2,'            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),=CLØ2'3,'            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),=CLØ2'4,'            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),=CLØ2'5,'             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),=CLØ2'6,'             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),=CLØ2'7,'             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),=CLØ2'8,'             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),=CLØ2'9,'             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),=CLØ3'1Ø,'            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),=CLØ3'11,'            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),=CLØ3'12,'            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),=CLØ3'13,'            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),=CLØ3'14,'            Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                      BUMP 2
BIT16C15 TM    WORKHEX2,B'ØØØØØØ1Ø'          IF BIT ON
         BNO   BIT16C16                      N. GO ON
         MVC   1(3,R9),=CLØ3'15,'            Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                      BUMP 2
BIT16C16 TM    WORKHEX2,B'ØØØØØØØ1'          IF BIT ON
         BNO   BIT16END                      N. GO ON
         MVC   1(3,R9),=CLØ3'16,'            Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                      BUMP 2
BIT16END MVI   Ø(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),=CL03'17,'           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),=CL03'18,'           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),=CL03'19,'           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),=CL03'20,'           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),=CL03'21,'           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),=CL03'22,'           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),=CL03'23,'           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),=CL03'24,'           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),=CL03'25,'           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),=CL03'26,'           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),=CL03'27,'           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),=CL03'28,'           Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                     BUMP 2
```

```
BIT32C29 TM    WORKHEX4,B'ØØØØ1ØØØ'         IF BIT ON
         BNO   BIT32C3Ø                     N. GO ON
         MVC   1(3,R9),=CLØ3'29,'           Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                     BUMP 2
BIT32C3Ø TM    WORKHEX4,B'ØØØØØ1ØØ'         IF BIT ON
         BNO   BIT32C31                     N. GO ON
         MVC   1(3,R9),=CLØ3'3Ø,'           Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                     BUMP 2
BIT32C31 TM    WORKHEX4,B'ØØØØØØ1Ø'         IF BIT ON
         BNO   BIT32C32                     N. GO ON
         MVC   1(3,R9),=CLØ3'31,'           Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                     BUMP 2
BIT32C32 TM    WORKHEX4,B'ØØØØØØØ1'         IF BIT ON
         BNO   BIT32END                     N. GO ON
         MVC   1(3,R9),=CLØ3'32,'           Y. PLUG NUMBER AND COMMA
         LA    R9,3(R9)                     BUMP 2
BIT32END MVI   Ø(R9),C')'                   PLUG RIGHT PAREN
         L     R14,BIT32SAV
         BR    R14
** ABENDRTN  - HANDLE SOFT ABEND
ABNDRTN  DS    ØH
         SNAP  DCB=SNAPDUMP,STORAGE=(WSSTART,WSEND)
         MVC   RETCODE,=H'16'
         B     FINALIZE
         TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               CONSTANTS                '
**       W O R K I N G   S T O R A G E
ZPARMGV$ CSECT
SAVEAREA DC    18F'Ø'
WSSTART  DS    ØF
         DC    CL16'WORKING STORAGE '
WRKPFLG1 DC    AL1(Ø)
         DS    ØD
D        DS    D
FULLWORD DC    F'1Ø24'
** PARMLIST VALUES
PARMLEN  DS    XL2
LOADNAME DS    CL8
** SQUEEZE ZERO HOLD AREA
ZEROHOLD DC    CL16' ',C' '
** ZPARM OUTPUT LINE
ZPRMLINE DS    ØCL(&ZPRMLNE)        SYSPUNCH 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
ZPRMCL4Ø EQU   ZPRMCL16+24,32,C'C'
         DC    (&ZPRMLNE-*+ZPRMLINE)C' '  FILL REMINDER OF LINE
REELINE  DS    CL(&ZPRMLNE)               KEEP PRINT INFO ABOUT ME   V6
```

```
ZPRMTITL DC    CL(&ZPRMLNE)'*'            MODULE TITLE LINE          V6
         ORG   ZPRMTITL+4                                            V6
ZTITNAME DC    CL(L'LOADNAME)'NAME'                                  V6
         ORG   ZPRMTITL+17                                           V6
ZTITLVLC DC    CL(L'SYSPLVLC)'LEVEL'                                 V6
         ORG   ZPRMTITL+35                                           V6
ZTITLVL  DC    CL(L'SPRMLVL)'RELEASE'                                V6
         ORG   ZPRMTITL+63                                           V6
ZTITDATE DC    CL(L'SPRMDATE)'ASSEMBLY'                              V6
         ORG   ZPRMTITL+L'ZPRMTITL                                   V6
RETCODE  DS    H'Ø'
** 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 FOR DSNZPARM
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
ABEND133 ABEND 133,DUMP                SPRMVCAT EYE CATCHER NOT FOUND
ABEND134 ABEND 134,DUMP                SPRMDB   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
ABEND2ØØ ABEND 2ØØ                     LOAD FAILED FOR DSNHDECP
ABEND2Ø1 ABEND 2Ø1,DUMP
ABEND2Ø2 ABEND 2Ø2,DUMP                   SECTION DSNMDECP NOT FOUND
ABEND2Ø3 ABEND 2Ø3,DUMP                DSN6ARVP EYE CATCHER NOT FOUND
ABEND2Ø4 ABEND 2Ø4,DUMP                DSN6LOGP EYE CATCHER NOT FOUND
ABEND2Ø5 ABEND 2Ø5,DUMP                DSN6SYSP EYE CATCHER NOT FOUND
ABEND2Ø6 ABEND 2Ø6,DUMP                DSN6FAC  EYE CATCHER NOT FOUND
ABEND2Ø7 ABEND 2Ø7,DUMP                DSN6GRP  EYE CATCHER NOT FOUND
SYSPUNCH DCB
DSORG=PS,RECFM=FB,MACRF=(PM),DDNAME=SYSPUNCH,LRECL=&ZPRML54584Ø
               LNE
DSNZPRM  DCB   DSORG=PO,MACRF=R,DDNAME=DSNZPARM
DSNDECP  DCB   DSORG=PO,MACRF=R,DDNAME=DSNHDECP
SNAPDUMP DCB   DSORG=PS,MACRF=W,RECFM=VBA,LRECL=125,BLKSIZE=882,       X
               DDNAME=SNAPDUMP
**       STRING CONSTANTS
STITLEL1 DC    CL&ZPRMLNE'*               DISPLAY DSNZPARM VALUES'
STITLEL2 DC    CL&ZPRMLNE'*               DISPLAY DSNHDECP VALUES'
WSEND    DC    CLØ8'WSEND>>'
LOADMPTR DS    3A                  ZPARM LOAD ADDRESS, LENGTH, END
         LTORG
TRTABLE  DC    XL256'Ø'
         ORG   TRTABLE+X'4Ø'
         DC    X'4Ø'                BLANK
         ORG
         END   ZPARMV6
//AD.SYSIN DD  *
DSNHDECM TITLE 'LOEBEN - ANALYSIS OF DSNHDECM - Ø3.Ø8.99               *
               DSNHDECM EXAMPLE COMPILE '
         PRINT GEN
     DSNHDECM CHARSET=ALPHANUM,                                        X
               ASCCSID=875,                                            X
               AMCCSID=65534,            65534                         X
               AGCCSID=65534,            65534                         X
               SCCSID=275,                                             X
               MCCSID=65534,             65534                         X
               GCCSID=65534,             65534                         X
               ENSCHEME=EBCDIC,                                        X
               DATE=ISO,                                               X
               DATELEN=15,                                             X
               DECARTH=DEC31,                                          X
               DECIMAL=PERIOD,                                         X
               DEFLANG=CPP,              COBOL,COB2,ASM,PLI,C,CPP      X
               DELIM=APOST,              APOST,COMMA,DEFAULT           X
               MIXED=NO,                                               X
               SQLDELI=APOST,            APOST,COMMA,DEFAULT           X
               DSQLDELI=APOST,           APOST,COMMA,DEFAULT           X
               SSID=DB25,                                              X
               DYNRULS=NO,                                             X
               STDSQL=NO, COMPAT=OFF,     V6 86,YES,NO                 X
               TIME=EUR, LC_CTYPE=ANY.NAM5.YOU.NEED.FOR.XLATION, V6    X
               TIMELEN=15
         END
//LD.SYSIN DD  *
   INCLUDE DSNHDECM
   ORDER   DSNAA
   INCLUDE ADSNLOAD(DSNAA)
   INCLUDE ADSNLOAD(DSNARIB)
   INCLUDE ADSNLOAD(DSNHDECP)
   ENTRY   DSNHDECP
   MODE    AMODE(24),RMODE(24)
   NAME    DSNHDECP(R)
//AX.SYSIN DD  *
DSNZPARM TITLE 'LOEBEN - ANALYSIS OF DSNZPARM AND DSNHDECP - Ø3.Ø8.99  *
               DSNZPARM EXAMPLE COMPILE '
         PRINT GEN
    DSN6ENV    MVS=XA
    DSN6SPRM   RESTART,                                                X
               ALL,                                                    X
               ABEXP=NO,                                               X
               ABIND=NO,                                               X
               AUTH=NO,                                                X
               AUTHCACH=1Ø75,             Ø-4Ø96                       X
               BINDNV=BINDADD,            BIND, BINDADD                X
               BMPTOUT=5,                                              X
               CACHEDYN=NO,                                            X
               CACHEPAC=25, CACHERAC=65,  V6                           X
               CATALOG=DSNCssØ5,                                       X
               DBCHK=NO,                                               X
               CDSSRDEF=ANY,                                           X
               CHGDC=NO,                                               X
               CONTSTOR=NO,                                            X
               DECDIV3=NO,                                             X
               DEFLTID=MEõWHOõ5,                                       X
               DESCSTAT=NO,                                            X
               DLITOUT=3,                                              X
               DSMAX=4765,                                             X
               EDMPOOL=12345, EDMDSPAC=234765, V6                      X
               EDPROP=NO,HOPAUTH=NO,      V6:BOTH/RUNNER  V5:YES/NO    X
               IRLMAUT=NO,                                             X
               IRLMPRC=IRLMPRO5,                                       X
               IRLMSID=IRØ5,                                           X
               IRLMRWT=995,                                            X
               IRLMSWT=ØØ5, LEMAX=5,      V6                           X
               MAXRBLK=125,                                            X
               MINRBLK=5,                                              X
               MAXKEEPD=15,                                            X
```

```
           NUMLKTS=25,                                             X
           NUMLKUS=15, OPTHINTS=NO,   V6                           X
           PARAMDEG=254,                                           X
           PARTKEYU=YES,                                           X
           RECALL=NO,                                              X
           RECALLD=665,                                            X
           RELCURHL=NO,RETLWAIT=NO, V6:Ø-..    V5:YES/NO           X
           RETVLCFK=NO,                                            X
           RGFCOLID=RGCO5,                                         X
           RGFDBNAM=RGFD5,                                         X
           RGFDEDPL=NO,                                            X
           RGFDEFLT=ACCEPT,                                        X
           RGFESCP=:,                                              X
           RGFFULLQ=NO,                                            X
           RGFINSTL=NO,                                            X
           RGFNMORT=REGISTER_OBJT_MY5,                             X
           RGFNMPRT=REGISTER_APPL_MY5,                             X
           RRULOCK=NO,                                             X
           SEQCACH=SEQ,                                            X
           SEQPRES=NO,                                             X
           SITETYP=RECOVERYSITE,                                   X
           SRTPOOL=32632,                                          X
           SYSADM=SYSADM12,                                        X
           SYSADM2=SYSADM13,                                       X
           SYSOPR1=SYSOPR12,                                       X
           SYSOPR2=SYSOPR13,                                       X
           TRKRSITE=NO,                                            X
           UTIMOUT=1Ø5,                                            X
           XLKUPDLT=NO
DSN6ARVP   ALCUNIT=BLK,               CYL,TRK,BLK                  X
           ARCWRTC=(1,3,5),                                        X
           ARCWTOR=NO,                                             X
           ARCPFX1=DSNCss15,                                       X
           ARCPFX2=DSNCss25,                                       X
           ARCRETN=5,                                              X
           BLKSIZE=15555,                                          X
           CATALOG=NO,                                             X
           COMPACT=NO,                                             X
           MSVGP=GROUP5,                                         V6X
           MSVGP2=GROUP25,                                       V6X
           PRIQTY=5,                                               X
           PROTECT=NO,                                             X
           QUIESCE=5,                                              X
           SECQTY=5,                                               X
           TSTAMP=NO,                                              X
           UNIT=DASD,                                              X
           UNIT2=DASD#2
DSN6LOGP   DEALLCT=(3,5),                                          X
           MAXARCH=995,                                            X
           MAXRTU=5,                                               X
```

```
               OUTBUFF=3995,                                           X
               OFFLOAD=NO,                                             X
               TWOACTV=NO,                                             X
               TWOARCH=NO,                                             X
               TWOBSDS=NO,                                             X
               WRTHRSH=5 ,ARC2FRST=NO                                V6
    DSN6SYSP   AUDITST=5,CONDBAT=5,  BACKODUR=5,                     V6X
               CTHREAD=5, DBPROTCL=DRDA,  V6 PRIVATE,DRDA              X
               DLDFREQ=5, DSSTIME=5,EXTRAREQ=5,EXTRASRV=5,           V6X
               IDBACK=5,                                               X
               IDFORE=5, IDXBPOOL=BP5,LBACKOUT=NO,LOBVALA=5,LOBVALS=5,
X5
               LOGLOAD=15999995, LOGAPSTG=5, V6                        X
               MAXDBAT=5,MON=NO,MONSIZE=99995,        8192-1Ø48576     X
               PCLOSEN=5,PCLOSET=5,PTASKROL=NO,                      V6X
               RLF=NO,RLFTBL=Ø5,RLFERR=5,RLFAUTH=SYSIBM5,              X
               ROUTCDE=(1,2,5),EXTSEC=NO,SMFACCT=5,                    X
               SMFSTAT=NO,STATIME=5,STORMXAB=5,STORPROC=STOR5,         X
               STORTIME=5, TBSBPOOL=BP5,  V6                           X
               TRACSTR=NO,TRACLOC=5,TRACTBL=5,                       V6X
               URCHKTH=5 ,WLMENV=MASTER5  V6
    DSN6FAC    DDF=AUTO, NO,COMMAND,AUTO                               X
               CMTSTAT=ACTIVE,            INACTIVE,ACTIVE              X
               IDTHTOIN=5, POOLINAC=5,    V6                           X
               RESYNC=5,                                               X
               RLFERRD=4999995, NOLIMIT,NORUN, 1-5ØØØØØØ V6            X
               TCPALVER=NO ,MAXTYPE1=5,TCPKPALV=5 V6
    DSN6GRP    DSHARE=NO,                                              X
               GRPNAME=GROUP5,                                         X
               MEMBNAME=MEMBER5,                                       X
               COORDNTR=NO,                                            X
               ASSIST=NO
         ORG
         END   DSN6SPRM
//LX.SYSIN DD  *
           ORDER            DSNAA
           INCLUDE LOADSET
           INCLUDE ADSNLOAD(DSNAA)
           INCLUDE SDSNLOAD(DSNZPARM)
           ENTRY DSNZMSTR                                            V6
           NAME DSNZPARM                                             V6
//LY.SYSIN DD  *
           ORDER            DSNAA
           INCLUDE LOADSET
           INCLUDE ADSNLOAD(DSNAA)
           INCLUDE SDSNLOAD(DSNZPARM)
           ENTRY DSNZMSTR                                            V6
           NAME DSNZPARN                                             V6
//
```

Loeben (Germany) © Xephon 2000

DSN1COPY generator utility – part 2

This month we continue the REXX procedure, DCU, which generates several DSN1COPY JCL streams.

- DSN1CM0 – panel:

```
)attr default(%+_)
   [ type (output) intens(low)  color(green) caps(off)
   # type (output) intens(low)  color(white) caps(off)
   _ type (input)  intens(low)  color(yellow) caps(off) pad('_')
   + type (text)   intens(low)  color(green)
   / type (text)   intens(low)  color(yellow)
   ~ type (text)   intens(high) color(turquoise)
   @ type (text)   intens(high) color(red)   caps(off) hilite(reverse)
)body window(78,22) expand ($$)
/.......................................................................
                          + @ DSN1COPY Utility +
%Command ===>_zcmd                                                     +
/.......................................................................
+
            + _z+[field1                                               +
            + _z+[field2                                               +
            +
            + _z+[field3                                               +
            +
            + _z+[field4                                               +
            + _z+[field5                                               +
            +
            + _z+[field6                                               +
+
/.......................................................................
+
                 #msg                                                  +
/PF3 - End +                                              ~Avg 1999,"ZB"
)init
  .ZVARS = '(f1 f2 f3 f4 f5 f6)'
  &field1 = 'Create a backup copy of a DB2 dataset'
  &field2 = 'Restore a backup copy of a DB2 dataset'
  &field3 = 'Move a DB2 dataset to another DB2 dataset'
  &field4 = 'Perform validity checking on a DB2 dataset'
  &field5 = 'Perform validity checking on and print a DB2 dataset'
  &field6 = 'Restore a tablespace from an Image copy'
  &msg = 'Place cursor on choice and press <Enter>'
  IF (&kurs = F1,FIELD1)
      .attr (field1) = 'color (yellow) caps(on)'
  IF (&kurs = F2,FIELD2)
      .attr (field2) = 'color (yellow) caps(on)'
  IF (&kurs = F3,FIELD3)
      .attr (field3) = 'color (yellow) caps(on)'
  IF (&kurs = F4,FIELD4)
      .attr (field4) = 'color (yellow) caps(on)'
  IF (&kurs = F5,FIELD5)
      .attr (field5) = 'color (yellow) caps(on)'
  IF (&kurs = F6,FIELD6)
      .attr (field6) = 'color (yellow) caps(on)'
)proc
  &kurs = .CURSOR
  if (.pfkey = pf03) &pf3 = exit
)end
```

- DSN1CM1 – panel:

```
)Attr Default(%+_)
   | type(text)   intens(high) caps(on ) color(yellow)
   $ type(output) intens(high) caps(off) color(yellow)
   ] type(output) intens(high) caps(on)  color(green) hilite(reverse)
   ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
   # type(text)   intens(high) caps(off) hilite(reverse)
   } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
   [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ]field                                                   +|-/-/-
%Command ===>_zcmd
+
+
#PARAMETER  #PARAMETER VALUE                   #PROMPT
+
+
+SSID     =>[db2 +                             DB2 Sub-System Identifier
+Creator  =>[crec    +                         Table Creator
+Name     =>[tabc             +                Table Name
+Tsname   =>[tsnc    +                         Tablespace Name
+Dbname   =>[dbnc    +                         Database   Name
+Stopts   =>[sts+                              Stop tablespace|YES+or|NO
+Devt     =>[devt+                             Device type|3390+or|TAPE
+Retpd    =>[rpd+                              Days Retained
+Withindx =>[wix+                              Include Index|YES+or|NO
+
            $msg
+
+
                               } PF3 Return +
)Init
  if (&poz='F4' | &poz='F5')
      &devt='3390'
      &rpd =' '
      &wix ='NO'
      .attr (devt) = 'type(output) color(white)'
      .attr (rpd)  = 'type(output) color(white)'
      .attr (wix)  = 'type(output) color(white)'
  if (&db2 ¬= ' ')
      .attr (db2) = 'pad(nulls)'
  if (&crec ¬= ' ')
      .attr (crec) = 'pad(nulls)'
  if (&tabc ¬= ' ')
      .attr (tabc) = 'pad(nulls)'
  if (&tsnc ¬= ' ')
      .attr (tsnc) = 'pad(nulls)'
  if (&dbnc ¬= ' ')
      .attr (dbnc) = 'pad(nulls)'
  if (&sts ¬= ' ')
      .attr (sts) = 'pad(nulls)'
  if (&devt ¬= ' ')
      .attr (devt) = 'pad(nulls)'
  if (&rpd¬= ' ')
      .attr (rpd) = 'pad(nulls)'
  if (&wix ¬= ' ')
      .attr (wix) = 'pad(nulls)'
  &msg='Enter values for the DSN1COPY service |'
)Reinit
)Proc
  &st = TRUNC(&sts,' ')
  if (&st='Y' | &st='YE') &sts = 'YES'
  if (&st='N')            &sts = 'NO'
  &de = TRUNC(&devt,' ')
  if (&de='3' | &de='33' | &de='339') &devt = '3390'
  if (&de='T' | &de='TA' | &de='TAP') &devt = 'TAPE'
  &wi = TRUNC(&wix,' ')
  if (&wi='Y' | &wi='YE') &wix = 'YES'
  if (&wi='N')            &wix = 'NO'
  VPUT (db2 crec tabc tsnc dbnc sts devt rpd wix) PROFILE
)End
```

- DSN1CM2 – panel:

```
)Attr Default(%+_)
   | type(text)   intens(high) caps(on ) color(yellow)
   $ type(output) intens(high) caps(off) color(yellow)
   ] type(output) intens(high) caps(off) color(green) hilite(reverse)
   ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
   # type(text)   intens(high) caps(off) hilite(reverse)
   } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
   [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ]field                                                   +|-/-/-
%Command ===>_zcmd
+
+
#PARAMETER  #PARAMETER VALUE                 #PROMPT
```

```
+
+
+SSID     =>[db2 +                            DB2 Sub-System Identifier
+Tosystem =>[sysi+                            To Sub-System Id
+Location =>[loc     +                        Location name for Tosystem
+Creator  =>[crec    +                        Table Creator
+Name     =>[tabc             +               Table Name
+Tsname   =>[tsnc    +                        Tablespace Name
+Dbname   =>[dbnc    +                        Database   Name
+Stopts   =>[sts+                             Stop tablespace|YES+or|NO
+Withindx =>[wix+                             Include Index|YES+or|NO
+Runstats =>[rus+                             Runstats |YES+or|NO
+
            $msg
+
+
                               } PF3 Return +
)Init
  &devt='3390'
  &rpd =' '
  if (&db2 ¬= ' ')
      .attr (db2) = 'pad(nulls)'
  if (&loc ¬= ' ')
      .attr (loc) = 'pad(nulls)'
  if (&sysi ¬= ' ')
      .attr (sysi) = 'pad(nulls)'
  if (&crec ¬= ' ')
      .attr (crec) = 'pad(nulls)'
  if (&tabc ¬= ' ')
      .attr (tabc) = 'pad(nulls)'
  if (&tsnc ¬= ' ')
      .attr (tsnc) = 'pad(nulls)'
  if (&dbnc ¬= ' ')
      .attr (dbnc) = 'pad(nulls)'
  if (&sts ¬= ' ')
      .attr (sts) = 'pad(nulls)'
  if (&wix ¬= ' ')
      .attr (wix) = 'pad(nulls)'
  if (&rus ¬= ' ')
      .attr (rus) = 'pad(nulls)'
  &msg='Enter values for the DSN1COPY service |'
)Reinit
)Proc
  &st = TRUNC(&sts,' ')
  if (&st='Y' | &st='YE') &sts = 'YES'
  if (&st='N')            &sts = 'NO'
  &wi = TRUNC(&wix,' ')
  if (&wi='Y' | &wi='YE') &wix = 'YES'
  if (&wi='N')            &wix = 'NO'
  &ru = TRUNC(&rus,' ')
  if (&ru='Y' | &ru='YE') &rus = 'YES'
  if (&ru='N')            &rus = 'NO'
  VPUT (db2 sysi loc crec tabc tsnc dbnc sts wix rus) PROFILE
)End
```

- DSN1CM3 – panel:

```
)Attr Default(%+_)
   ( type(text  ) intens(high) hilite(reverse)
   ] type(text  ) intens(high) hilite(reverse) color(green)
   / type(text  ) intens(high) hilite(reverse) color(yellow)
   ~ type(output) intens(high)                color(red)
   [ type(output) intens(high) hilite(reverse) color(green) caps(off)
   + type(text  ) intens(low )
   _ type( input) intens(high) caps(on ) just(left )
   ¬ type(output) intens(low ) caps(off) just(asis )
)Body window(55,19)
/ DSN1COPY Utility - Selection Result +
+
+Command ===>_zcmd                  +Scroll ===>_amt +
+
+Press]Enter+to have this service continue.
+Press]End  +to respecify your  PARAMETERS.
+
]Obj]Dbname  ]Tsname  ]Part ]         Card] # Tables+
+
)Model
¬z  ¬z       ¬z       ¬z  + ¬z           +    ¬z  +
)Init
  &ZWINTTL = '* Dsn1copy *'
  .ZVARS = '(ob v1 v2 v3 v4 v5)'
  &amt = PAGE
)Reinit
)Proc
)End
```

- DSN1CM4 – panel:

```
)Attr Default(%+_)
   ( type(text  ) intens(high) hilite(reverse)
   ] type(text  ) intens(high) hilite(reverse) color(green)
   / type(text  ) intens(high) hilite(reverse) color(yellow)
   ) type(text  ) intens(high) hilite(reverse) color(white)
   ~ type(output) intens(high)                color(red)
   [ type(output) intens(high) hilite(reverse) color(green) caps(off)
   } type(output) intens(high) color(white)
   { type(output) intens(high) color(green)
   + type(text  ) intens(low )
   _ type( input) intens(high) caps(on ) just(left )
   ¬ type(output) intens(low ) color(green) caps(off) just(asis )
   @ type(output) intens(low ) color(white) caps(off) just(asis )
)Body
/ DSN1COPY Utility - Selection Result +
+
+Command ===>_zcmd                                  +Scroll ===>_amt +
+
+Press]Enter+to have this service continue.
+Press]End  +to respecify your  PARAMETERS.
+
---------------{db2 +--------------- --------------}sysi+---------------
]Dbname  ]Tsname  ]Pr]Table name      )Dbname  )Tsname  )Pr)Table name
+
)Model
¬z       ¬z       ¬z ¬z                 @z       @z       @z @z
)Init
  &ZWINTTL = '* Dsn1copy *'
  .ZVARS = '(v1 v2 v3 v4 v5 v6 v7 v8)'
  &amt = PAGE
)Reinit
)Proc
)End
```

- DSN1CM5 – panel:

```
)Attr Default(%+_)
   | type(text)   intens(high) caps(on ) color(yellow)
   $ type(output) intens(high) caps(off) color(yellow)
   ] type(output) intens(high) caps(on)  color(green) hilite(reverse)
   ? type(text)   intens(high) caps(on ) color(green) hilite(reverse)
   # type(text)   intens(high) caps(off) hilite(reverse)
   } type(text)   intens(high) caps(off) color(yellow) hilite(reverse)
   [ type( input) intens(high) caps(on ) color(green) pad(_)
)Body  Expand(//)
|-/-/- ]field                                                   +|-/-/-
%Command ===>_zcmd
+
+
#PARAMETER  #PARAMETER VALUE                  #PROMPT
+
+
+SSID     =>[db2 +                            DB2 Sub-System Identifier
+Creator  =>[crec    +                        Table Creator
+Name     =>[tabc            +                Table Name
+Stopts   =>[sts+                             Stop tablespace|YES+or|NO
+
            $msg
+
+
                               } PF3 Return +
)Init
  if (&db2 ¬= ' ')
      .attr (db2) = 'pad(nulls)'
  if (&crec ¬= ' ')
      .attr (crec) = 'pad(nulls)'
  if (&tabc ¬= ' ')
```

```
      .attr (tabc) = 'pad(nulls)'
  if (&sts ¬= ' ')
      .attr (sts) = 'pad(nulls)'
  &msg='Enter values for the DSN1COPY service |'
)Reinit
)Proc
  &st = TRUNC(&sts,' ')
  if (&st='Y' | &st='YE') &sts = 'YES'
  if (&st='N')            &sts = 'NO'
  VPUT (db2 crec tabc sts) PROFILE
)End
```

- DSN1CM6 – panel:

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

- DSN1UT – panel:

```
)attr default($+_)
| type (text)   intens(low)  color(white)
@ type (text)   intens(high) color(red)   caps(off) hilite(reverse)
```

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

- DSN1CP1 – PL/I program:

```
* PROCESS GS,OFFSET,OPT(TIME);
 DSN1CP1:PROC(PARMS)OPTIONS(MAIN) REORDER;
 /******************************************************************/
 /* DESCRIPTION: PL/I PROGRAM FOR DSN1COPY UTILITY                 */
 /******************************************************************/
  DCL PARMS CHAR(100) VAR;
  DCL SYSPRINT    FILE STREAM OUTPUT;
  DCL CARD          BIN FIXED(31);
  DCL NTABLES       BIN FIXED(15);
  DCL HNTABLES      PIC 'Z9';
  DCL MCARD         PIC'--.---.---.--9';
  DCL HDBNAME       CHAR(8)  VAR;
  DCL HTSNAME       CHAR(8)  VAR;
  DCL VCAT          CHAR(8);
  DCL HPART         BIN FIXED(15);
  DCL PART          PIC'ZZ9';
  DCL 1 WORKST,
      2 CREC        CHAR(8)  VAR,
      2 TABC        CHAR(18) VAR,
      2 TSNC        CHAR(8)  VAR,
      2 DBNC        CHAR(8)  VAR,
      2 POZ         CHAR(2),
      2 WIX         CHAR(3);
  DCL (SUBSTR,DATE,TIME,NULL,ADDR,LENGTH,INDEX) BUILTIN;
  DCL IC            BIN FIXED(15);
  DCL OUT           CHAR(18) VAR;
  EXEC SQL INCLUDE SQLCA;
  IF SUBSTR(PARMS,1,8)=' ' THEN CREC='%';
  ELSE DO;
     CALL FUNC(SUBSTR(PARMS,1,8),OUT);
     CREC=OUT;
     IF LENGTH(CREC) < 8 THEN CREC=CREC||'%';
  END;
  IF SUBSTR(PARMS,9,18)=' ' THEN TABC='%';
  ELSE DO;
```

```
   CALL FUNC(SUBSTR(PARMS,9,18),OUT);
   TABC=OUT;
   IF LENGTH(TABC) < 18 THEN TABC=TABC||'%';
END;
IF SUBSTR(PARMS,27,8)=' ' THEN TSNC='%';
ELSE DO;
   CALL FUNC(SUBSTR(PARMS,27,8),OUT);
   TSNC=OUT;
   IF LENGTH(TSNC) < 8 THEN TSNC=TSNC||'%';
END;
IF SUBSTR(PARMS,35,8)=' ' THEN DBNC='%';
ELSE DO;
   CALL FUNC(SUBSTR(PARMS,35,8),OUT);
   DBNC=OUT;
   IF LENGTH(DBNC) < 8 THEN DBNC=DBNC||'%';
END;
POZ=SUBSTR(PARMS,43,2);
WIX=SUBSTR(PARMS,45,3);
EXEC SQL DECLARE C1 CURSOR WITH HOLD FOR
SELECT DISTINCT S.DBNAME,S.NAME,P.PARTITION,P.VCATNAME,
       NTABLES, P.CARD
FROM SYSIBM.SYSTABLES T,
     SYSIBM.SYSTABLESPACE S,
     SYSIBM.SYSTABLEPART  P
WHERE T.CREATOR LIKE :CREC
  AND T.NAME    LIKE :TABC
  AND T.TSNAME  LIKE :TSNC
  AND T.DBNAME  LIKE :DBNC
  AND T.TSNAME  = S.NAME
  AND T.DBNAME  = S.DBNAME
  AND P.TSNAME  = S.NAME
  AND P.DBNAME  = S.DBNAME
  AND T.TYPE = 'T'
ORDER BY 1, 2, 3
FOR FETCH ONLY;
EXEC SQL OPEN C1;
EXEC SQL FETCH C1
INTO :HDBNAME, :HTSNAME, :HPART, :VCAT, :NTABLES, :CARD;
IF SQLCODE=Ø
THEN PUT SKIP LIST (' SQLCODE= Ø');
ELSE DO;
   PUT SKIP LIST (' SQLCODE= '||SQLCODE);
   GOTO VEN;
END;
DO WHILE (SQLCODE=Ø);
   PART=HPART;
   HNTABLES=NTABLES;
   MCARD=CARD;
   PUT SKIP LIST ('TS '||HDBNAME||' '||HTSNAME||' '||PART||' '||VCAT
   ||' '||HNTABLES||' '||MCARD);
   EXEC SQL FETCH C1
   INTO :HDBNAME, :HTSNAME, :HPART, :VCAT, :NTABLES, :CARD;
  END;
  EXEC SQL CLOSE C1;
  IF WIX='YES' THEN DO;
     EXEC SQL DECLARE C2 CURSOR WITH HOLD FOR
     SELECT I.DBNAME,I.NAME,P.PARTITION,
            P.VCATNAME,P.CARD
     FROM SYSIBM.SYSTABLES T,
          SYSIBM.SYSINDEXES I,
          SYSIBM.SYSINDEXPART P
     WHERE T.CREATOR LIKE :CREC
       AND T.NAME    LIKE :TABC
       AND T.TSNAME  LIKE :TSNC
       AND T.DBNAME  LIKE :DBNC
       AND T.CREATOR   = I.TBCREATOR
       AND T.NAME      = I.TBNAME
       AND P.IXCREATOR = I.CREATOR
       AND P.IXNAME    = I.NAME
       AND T.TYPE = 'T'
     ORDER BY 1, 2, 3
     FOR FETCH ONLY;
     EXEC SQL OPEN C2;
     EXEC SQL FETCH C2
     INTO :HDBNAME, :HTSNAME, :HPART, :VCAT, :CARD;
     DO WHILE (SQLCODE=Ø);
        PART=HPART;
        MCARD=CARD;
        PUT SKIP LIST ('IX '||HDBNAME||' '||HTSNAME
        ||' '||PART||' '||VCAT||' - '||MCARD);
        EXEC SQL FETCH C2
        INTO :HDBNAME, :HTSNAME, :HPART, :VCAT, :CARD;
     END;
     EXEC SQL CLOSE C2;
  END;
  FUNC:PROC(INP,OUT);
       DCL INP CHAR(18);
       DCL OUT CHAR(18) VAR;
       DO IC=1 TO 18 BY 1 WHILE (SUBSTR(INP,IC,1) ¬=' ');
       END;
       OUT=SUBSTR(INP,1,IC-1);
   END FUNC;
  VEN:
 END DSN1CP1;
```

- DSN1CP2 – PL/I program:

```
* PROCESS GS,OFFSET,OPT(TIME);
 DSN1CP2:PROC(PARMS)OPTIONS(MAIN) REORDER;
 /******************************************************************/
 /* DESCRIPTION: PL/I PROGRAM FOR DSN1COPY UTILITY                 */
 /******************************************************************/
  DCL PARMS CHAR(1ØØ) VAR;
  DCL SYSPRINT    FILE STREAM OUTPUT;
  DCL DBID          BIN FIXED(15);
  DCL PSID          BIN FIXED(15);
  DCL OBID          BIN FIXED(15);
  DCL TDBID         BIN FIXED(15);
  DCL TPSID         BIN FIXED(15);
  DCL TOBID         BIN FIXED(15);
  DCL HDBNAME       CHAR(8)  VAR;
  DCL HTSNAME       CHAR(8)  VAR;
  DCL TDB           CHAR(8)  VAR;
  DCL TTS           CHAR(8)  VAR;
  DCL TAB           CHAR(18) VAR;
  DCL CRE           CHAR(8)  VAR;
  DCL VCAT          CHAR(8);
  DCL TVC           CHAR(8);
  DCL HPARTS        BIN FIXED(15);
  DCL HPART         BIN FIXED(15);
  DCL PART          PIC'ZZ9';
  DCL PARTS         PIC'ZZ9';
  DCL HDBID         PIC'ZZZ9';
  DCL HPSID         PIC'ZZZ9';
  DCL HOBID         PIC'ZZZ9';
  DCL TDBI          PIC'ZZZ9';
  DCL TPS           PIC'ZZZ9';
  DCL TOB           PIC'ZZZ9';
  DCL 1 WORKST,
      2 CREC        CHAR(8)  VAR,
      2 TABC        CHAR(18) VAR,
      2 TSNC        CHAR(8)  VAR,
      2 DBNC        CHAR(8)  VAR,
      2 WIX         CHAR(3),
      2 LOC         CHAR(8)  VAR;
  DCL QUERY         CHAR(4ØØ) VAR INIT(' ');
  DCL FIELDI        CHAR(133) VAR INIT(' ');
  DCL (SUBSTR,DATE,TIME,NULL,ADDR,LENGTH,INDEX) BUILTIN;
  DCL IC            BIN FIXED(15);
  DCL OUT           CHAR(18) VAR;
  EXEC SQL INCLUDE SQLCA;
  IF SUBSTR(PARMS,1,8)=' ' THEN CREC='%';
  ELSE DO;
     CALL FUNC(SUBSTR(PARMS,1,8),OUT);
     CREC=OUT;
     IF LENGTH(CREC) < 8 THEN CREC=CREC||'%';
  END;
  IF SUBSTR(PARMS,9,18)=' ' THEN TABC='%';
  ELSE DO;
     CALL FUNC(SUBSTR(PARMS,9,18),OUT);
     TABC=OUT;
     IF LENGTH(TABC) < 18 THEN TABC=TABC||'%';
  END;
  IF SUBSTR(PARMS,27,8)=' ' THEN TSNC='%';
  ELSE DO;
     CALL FUNC(SUBSTR(PARMS,27,8),OUT);
     TSNC=OUT;
     IF LENGTH(TSNC) < 8 THEN TSNC=TSNC||'%';
  END;
  IF SUBSTR(PARMS,35,8)=' ' THEN DBNC='%';
  ELSE DO;
     CALL FUNC(SUBSTR(PARMS,35,8),OUT);
     DBNC=OUT;
     IF LENGTH(DBNC) < 8 THEN DBNC=DBNC||'%';
  END;
  WIX=SUBSTR(PARMS,43,3);
  IF SUBSTR(PARMS,46,8)¬=' '
  THEN DO;
     CALL FUNC(SUBSTR(PARMS,46,8),OUT);
     LOC=OUT;
  END;
  EXEC SQL DECLARE C1 CURSOR WITH HOLD FOR
  SELECT DISTINCT S.DBNAME,S.NAME,P.PARTITION,P.VCATNAME,
         T.CREATOR, T.NAME, S.DBID, S.PSID, T.OBID, S.PARTITIONS
  FROM SYSIBM.SYSTABLES T,
       SYSIBM.SYSTABLESPACE S,
       SYSIBM.SYSTABLEPART  P
  WHERE T.CREATOR LIKE :CREC
    AND T.NAME    LIKE :TABC
    AND T.TSNAME  LIKE :TSNC
    AND T.DBNAME  LIKE :DBNC
    AND T.TSNAME  = S.NAME
    AND T.DBNAME  = S.DBNAME
    AND P.TSNAME  = S.NAME
    AND P.DBNAME  = S.DBNAME
    AND T.TYPE = 'T'
  ORDER BY 1, 2, 3
  FOR FETCH ONLY;
  EXEC SQL OPEN C1;
  EXEC SQL FETCH C1 INTO :HDBNAME,:HTSNAME,:HPART,
                :VCAT,:CRE,:TAB,:DBID,:PSID,:OBID,:HPARTS;
  IF SQLCODE=Ø
  THEN PUT SKIP LIST (' SQLCODE= Ø');
  ELSE DO;
     PUT SKIP LIST (' SQLCODE= '||SQLCODE||' '||LOC);
     GOTO VEN;
  END;
  DO WHILE (SQLCODE=Ø);
     PART=HPART;
     PARTS=HPARTS;
     HDBID=DBID;
     HPSID=PSID;
     HOBID=OBID;
     CALL LOCAL;
     PUT SKIP LIST
     ('TS '||HDBNAME||' '||HTSNAME||' '||PART||' '||VCAT||' '||PARTS
     ||' '||CRE||' '||TAB||' '||HDBID||' '||HPSID||' '||HOBID||' '||
     TDB||' '||TTS||' '||TVC||' '||TDBI||' '||TPS||' '||TOB);
     IF WIX='YES'
     THEN CALL IX;
     ELSE PUT SKIP LIST(' ');
     EXEC SQL FETCH C1 INTO :HDBNAME,:HTSNAME,:HPART,
                   :VCAT,:CRE,:TAB,:DBID,:PSID,:OBID,:HPARTS;
  END;
  EXEC SQL CLOSE C1;
  LOCAL:PROC;
    QUERY='SELECT DISTINCT S.DBNAME,S.NAME,P.VCATNAME,'||
          'S.DBID, S.PSID, T.OBID FROM '||
       LOC||'.SYSIBM.SYSTABLES T,'||
       LOC||'.SYSIBM.SYSTABLESPACE S,'||
       LOC||'.SYSIBM.SYSTABLEPART P '||
       'WHERE T.CREATOR='''||CRE||''' '||
        'AND T.NAME='''||TAB||''' '||
        'AND T.TSNAME=S.NAME '||
        'AND T.DBNAME=S.DBNAME '||
        'AND P.TSNAME=S.NAME '||
        'AND P.DBNAME=S.DBNAME '||
        'AND T.TYPE='''||'T'||'''';
    EXEC SQL DECLARE C2 CURSOR FOR STMT;
    EXEC SQL PREPARE STMT FROM QUERY;
    EXEC SQL OPEN C2;
    EXEC SQL FETCH C2 INTO :TDB,:TTS,:TVC,:TDBID,:TPSID,:TOBID;
    IF SQLCODE=Ø THEN DO;
       TDBI=TDBID;
       TPS=TPSID;
       TOB=TOBID;
    END;
    ELSE DO;
       TDB='-';
       TTS='-';
    END;
    EXEC SQL CLOSE C2;
  END LOCAL;
  IX:PROC;
    QUERY='SELECT DISTINCT S.DBNAME,S.INDEXSPACE,P.VCATNAME,'||
          'S.OBID, S.DBID, S.ISOBID FROM '||
          'SYSIBM.SYSTABLES T,'||
          'SYSIBM.SYSINDEXES S,'||
          'SYSIBM.SYSINDEXPART P '||
          'WHERE T.CREATOR='''||CRE||''' '||
            'AND T.NAME='''||TAB||''' '||
            'AND T.NAME=S.TBNAME '||
            'AND T.CREATOR=S.TBCREATOR '||
            'AND P.IXNAME=S.NAME '||
            'AND P.IXCREATOR=S.CREATOR '||
            'AND P.PARTITION='||HPART||' '||
            'AND T.TYPE='''||'T'||'''';
    EXEC SQL DECLARE C3 CURSOR FOR STMTI;
    EXEC SQL PREPARE STMTI FROM QUERY;
    EXEC SQL OPEN C3;
```

```
   EXEC SQL FETCH C3 INTO :TDB,:TTS,:TVC,:TOBID,:TDBID,:TPSID;
   DO WHILE (SQLCODE=Ø);
      TDBI=TDBID;
      TPS=TPSID;
      TOB=TOBID;
      FIELDI='IX '||TDB||' '||TTS||' '||PART||' '||TVC||
             ' '||PARTS||' '||TDBI||' '||TPS||' '||TOB;
      QUERY='SELECT DISTINCT S.DBNAME,S.INDEXSPACE,P.VCATNAME,'||
            'S.OBID, S.DBID, S.ISOBID FROM '||
            LOC||'.SYSIBM.SYSTABLES T,'||
            LOC||'.SYSIBM.SYSINDEXES S,'||
            LOC||'.SYSIBM.SYSINDEXPART P '||
            'WHERE T.CREATOR='''||CRE||''' '||
              'AND T.NAME='''||TAB||''' '||
              'AND T.NAME=S.TBNAME '||
              'AND T.CREATOR=S.TBCREATOR '||
              'AND P.IXNAME=S.NAME '||
              'AND P.IXCREATOR=S.CREATOR '||
              'AND S.INDEXSPACE='''||TTS||''' '||
              'AND P.PARTITION='||HPART||' '||
              'AND T.TYPE='''||'T'||'''';
      EXEC SQL DECLARE C4 CURSOR FOR STMTIR;
      EXEC SQL PREPARE STMTIR FROM QUERY;
      EXEC SQL OPEN C4;
      EXEC SQL FETCH C4 INTO :TDB,:TTS,:TVC,:TOBID,:TDBID,:TPSID;
      IF SQLCODE=Ø THEN DO;
         TDBI=TDBID;
         TPS=TPSID;
         TOB=TOBID;
         FIELDI=FIELDI||' '||TDB||' '||TTS||' '||' '||TVC||
                       ' '||TDBI||' '||TPS||' '||TOB;
         PUT SKIP LIST (FIELDI);
      END;
      ELSE PUT SKIP LIST ('SQLCODE='||SQLCODE);
      EXEC SQL CLOSE C4;
      EXEC SQL FETCH C3 INTO :TDB,:TTS,:TVC,:TOBID,:TDBID,:TPSID;
   END;
   EXEC SQL CLOSE C3;
 END IX;

 FUNC:PROC(INP,OUT);
      DCL INP CHAR(18);
      DCL OUT CHAR(18) VAR;
      DO IC=1 TO 18 BY 1 WHILE (SUBSTR(INP,IC,1) ¬=' ');
      END;
      OUT=SUBSTR(INP,1,IC-1);
  END FUNC;
 VEN:
END DSN1CP2;
```

- DSN1CP3 – PL/I program:

```
* PROCESS GS,OFFSET,OPT(TIME);
 DSN1CP3:PROC(PARMS)OPTIONS(MAIN) REORDER;
 /******************************************************************/
 /* DESCRIPTION: PL/I PROGRAM FOR DSN1COPY UTILITY                 */
 /******************************************************************/
  DCL PARMS CHAR(1ØØ) VAR;
  DCL SYSPRINT    FILE STREAM OUTPUT;
  DCL HDBNAME       CHAR(8)  VAR;
  DCL HTSNAME       CHAR(8)  VAR;
  DCL VCAT          CHAR(8);
  DCL DATUM         CHAR(1Ø);
  DCL CAS           CHAR(8);
  DCL DEV           CHAR(4);
  DCL DSN           CHAR(44);
  DCL HPART         BIN FIXED(15);
  DCL 1 WORKST,
      2 CREC        CHAR(8)  VAR,
      2 TABC        CHAR(18) VAR;
  DCL (SUBSTR,DATE,TIME,NULL,ADDR,LENGTH,INDEX) BUILTIN;
  DCL IC            BIN FIXED(15);
  DCL OUT           CHAR(18) VAR;
  EXEC SQL INCLUDE SQLCA;
  CREC=SUBSTR(PARMS,1,8);
  CALL FUNC(SUBSTR(PARMS,9,18),OUT);
  TABC=OUT;
  /* TABLE ROW                                                      */
  EXEC SQL SELECT DISTINCT S.DBNAME, S.NAME, P.VCATNAME, PARTITIONS
  INTO :HDBNAME, :HTSNAME, :VCAT, :HPART
  FROM SYSIBM.SYSTABLES T,
       SYSIBM.SYSTABLESPACE S,
       SYSIBM.SYSTABLEPART  P
  WHERE T.CREATOR = :CREC
    AND T.NAME    = :TABC
    AND T.TSNAME  = S.NAME
    AND T.DBNAME  = S.DBNAME
    AND P.TSNAME  = S.NAME
    AND P.DBNAME  = S.DBNAME
    AND T.TYPE = 'T';
  IF SQLCODE=Ø & HPART>Ø
   THEN DO;
     PUT SKIP LIST (' SQLCODE= 9999');
     GOTO VEN;
  END;
  IF SQLCODE=Ø
  THEN PUT SKIP LIST (' SQLCODE= Ø');
  ELSE DO;
     IF SQLCODE=-2Ø4
     THEN PUT SKIP LIST (' SQLCODE= '||SQLCODE||' '||'TABLE NOT FOUND');
     ELSE PUT SKIP LIST (' SQLCODE= '||SQLCODE);
     GOTO VEN;
  END;
  /* IMAGE COPY ROWS                                               */
  EXEC SQL DECLARE C1 CURSOR WITH HOLD FOR
  SELECT DATE(TIMESTAMP),TIME(TIMESTAMP),DEVTYPE,DSNAME
  FROM SYSIBM.SYSCOPY
  WHERE DBNAME=:HDBNAME
    AND TSNAME=:HTSNAME
    AND ICTYPE='F'
    AND ICUNIT='D'
  ORDER BY 1 DESC;
  EXEC SQL OPEN C1;
  EXEC SQL FETCH C1 INTO :DATUM, :CAS, :DEV, :DSN;
  IF SQLCODE¬=Ø
  THEN DO;
     IF SQLCODE=1ØØ
     THEN PUT SKIP LIST (' SQLCODE= '||SQLCODE||' '||'IC NOT FOUND');
     ELSE PUT SKIP LIST (' SQLCODE= '||SQLCODE);
     GOTO VEN;
  END;
  DO WHILE (SQLCODE=Ø);
     PUT SKIP LIST ('IC '||DATUM||' '||CAS||' '||DEV||' '||DSN||
                  ' '||HDBNAME||' '||HTSNAME||' '||VCAT);
     EXEC SQL FETCH C1 INTO :DATUM, :CAS, :DEV, :DSN;
  END;
  EXEC SQL CLOSE C1;
  FUNC:PROC(INP,OUT);
       DCL INP CHAR(18);
       DCL OUT CHAR(18) VAR;
       DO IC=1 TO 18 BY 1 WHILE (SUBSTR(INP,IC,1) ¬=' ');
       END;
       OUT=SUBSTR(INP,1,IC-1);
   END FUNC;
  VEN:
 END DSN1CP3;
```

- DSN1COP1 – JCL skeleton:

```
)TBA 72
)CM -------------------------------------------------------------------
)CM Skeleton to generate DSN1COPY utility                            --
)CM -------------------------------------------------------------------
//&user.X JOB (12ØØ-12Ø5-ØØ),'&option',
//             NOTIFY=&user,REGION=4M,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//* *******************************************************************
//*    &title
//*    GENERATION DATE AND TIME : &date AT: &time
//*
//*    CALCULATING TIME IS &ctime SECONDS.
//*
//*    DSN1COPY  - WAS RUN WITH THE FOLLOWING PARAMETERS:
```

```
//*    PARAMETER   PARAMETER VALUE
//*    ----------  ---------------------
//*    SSID     :  &db2
//*    Creator  :  &creC
//*    Name     :  &tabc
//*    Tsname   :  &tsnc
//*    Dbname   :  &dbnc
//*    Stopts   :  &sts
//*    Devt     :  &devt
//*    Retpd    :  &rpd
//*    Withindx :  &wix
//* *******************************************************************
//*  NUM  DATABASE  TABLESPACE   TRACKS   PART
//*  ---  --------  ----------  -------   ----
)DOT "ALIST"
//* &detail
)ENDDOT
//*  ---  --------  ----------  -------   ----
//*                       TOTAL:&tot TRACKS OR
//*                             &cyl CYLINDERS
//*----------------------------------------------------
//*  NAMING CONVENTION USED WITH DSN1COPY DATASETS:
//*     PART 1=&user..DCU
//*          2=DSN1CXXX, WHERE XXX = PARTITION NUMBER
//*          3=DBNAME  , WHERE DBNAME = DATABASE NAME
//*          4=TSNAME  , WHERE TSNAME = TABLESPACE NAME
//*----------------------------------------------------
//*---- DSN1COPY - TABLESPACES/INDEXSPACES
//*----------------------------------------------------
)SEL &sts = YES
//*---- STOP TABLESPACES/INDEXSPACES ------------------
//STOPTS  EXEC PGM=IKJEFTØ1,COND=(4,LT)
//STEPLIB  DD DSN=DSN51Ø.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
   DSN SYSTEM(&db2)
)DOT "ALIST"
)SEL &pr = Ø
   -STOP DATABASE(&db) SPACENAM(&ts)
)ENDSEL
)SEL &pr > Ø
   -STOP DATABASE(&db) SPACENAM(&ts) PART(&pr)
)ENDSEL
)ENDDOT
//*
)ENDSEL
)SEL &devt EQ 339Ø AND &poz EQ F1
//*----------------------------------------------------
//*---- DSN1COPY - SCRATCH DASD DATASETS  -------------
//*----------------------------------------------------
//SCRATCH  EXEC PGM=IDCAMS,COND=(4,LT)
```

```
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
)DOT "ALIST"
   DELETE '&user..DCU.DSN1C&pr1..&db..&ts'
)ENDDOT
   SET MAXCC = Ø
/*
)ENDSEL
)DOT "ALIST"
//*----------------------------------------------------
)SEL &pr = Ø
//*  DSN1COPY - OF &db..&ts
)ENDSEL
)SEL &pr > Ø
//*  DSN1COPY - OF &db..&ts PART &pr
)ENDSEL
//*----------------------------------------------------
)SEL &poz = F1
//COPY&scu EXEC PGM=DSN1COPY,COND=(4,LT)
)ENDSEL
)SEL &poz = F4
//COPY&scu EXEC PGM=DSN1COPY,PARM='CHECK',COND=(4,LT)
)ENDSEL
)SEL &poz = F5
//COPY&scu EXEC PGM=DSN1COPY,PARM='CHECK,PRINT',COND=(4,LT)
)ENDSEL
//STEPLIB  DD DSN=DSN51Ø.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=&catn..DSNDBD.&db..&ts..IØØØ1.A&pr1,
//         DISP=OLD
)SEL &devt EQ 339Ø AND &poz EQ F1
//SYSUT2   DD DISP=(NEW,CATLG,CATLG),
//         UNIT=339Ø,
//         DCB=BLKSIZE=28672,
//         SPACE=(TRK,(&pri,&sec,),RLSE),
//         DSN=&user..DCU.DSN1C&pr1..&db..&ts
)ENDSEL
)SEL &poz EQ F4 OR &poz EQ F5
//SYSUT2   DD DUMMY
)ENDSEL
)SEL &devt = TAPE
//SYSUT2   DD DISP=(,KEEP),
//         UNIT=TAPE,
//         DCB=BLKSIZE=28672,
//         LABEL=(&scu,RETPD=14),
)SEL &scu = 1
//         VOL=(PRIVATE,RETAIN),
)ENDSEL
)SEL &scu > 1
//         VOL=(PRIVATE,RETAIN,REF=*.COPY1.SYSUT2),
)ENDSEL
```

```
//         DSN=&user..DCU.DSN1C&pr1..&db..&ts
)ENDSEL
)ENDDOT
/*
)SEL &sts = YES
//*---- START TABLESPACES/INDEXSPACES -----------------
//STARTS  EXEC PGM=IKJEFTØ1,COND=(4,LT)
//STEPLIB  DD DSN=DSN51Ø.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
   DSN SYSTEM(&db2)
)DOT "ALIST"
)SEL &pr = Ø
   -START DATABASE(&db) SPACENAM(&ts)
)ENDSEL
)SEL &pr > Ø
   -START DATABASE(&db) SPACENAM(&ts) PART(&pr)
)ENDSEL
)ENDDOT
   -DIS   DATABASE(*) SPACENAM(*) RESTRICT
//*
)ENDSEL
```

Editor's note: this article will be concluded in the next issue.

Bernard Zver
Database Administrator
Informatika Maribor (Slovenia)

DB2 news

NEON Systems has integrated its Diplomat product with Shadow Direct, providing application integration between different customer relationship management and help desk applications, as well as direct access to System/390 data and transactions.

Diplomat enables EAI and B2B event-driven integration for different mainframe, Windows NT, and Unix applications in mixed cross-function environments.

With Diplomat, event-driven integration is streamlined among intra-office business units as well as among B2B and B2C interactions.

Shadow Direct provides access to System/390 IMS, CICS, VSAM, ADABAS, and DB2 data and transactions.

For further information contact:
NEON Systems, 14100 Southwest Freeway, #500 Sugarland, TX 77478, USA.
Tel: (281) 491 4200.
URL: http://www.neonsys.com.

* * *

IBM is promising that the next version of DB2 will include free integration and analysis tools when it ships this summer. The tools will be similar to the software that Microsoft plans to bundle with the next version of SQL Server (also due in the summer).

The products that IBM is planning to give free include Data Joiner, which allows users to query both relational and non-relational data concurrently, and Hyperion's Essbase OLAP server.

New features expected in the new release include 64-bit support, Native XML, Java, spatial data support, and the ability to reload data and restart the database at any point.

For further information contact your local IBM representative.
http://www.software.ibm.com/data/db2.

* * *

IBM has begun shipping the DB2 Universal Database, both DB2 UDB Enterprise Edition and Enterprise – Extended Edition (EEE), for its Intel processor-based NUMA-Q servers, acquired through Sequent.

DB2 UDB EE scales from single processor systems to the data centre, while UDB EEE provides a high performance mechanism to support large databases, possibly previously housed on clustered servers. It's said to be ideal for applications like data warehousing, and business intelligence. It's also been tweaked to use the linear scalability of NUMA-Q.

IBM is also delivering a DB2 Software Developer's Kit, a run-time client, and DB2 Connect Enterprise Edition for NUMA-Q. DB2 Connect provides direct access from NUMA-Q servers and gateways to DB2 databases on hosts such as OS/390, VM, and OS/400.

Net.Data, MQ Series, and Tivoli Storage Manager have been ported to NUMA-Q, while the DB2 XML Extender is also being ported to NUMA-Q.

For further information contact your local IBM representative.
http://www.software.ibm.com/data/db2.