68

DB2

June 1998

In this issue

update

DB2 Update

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

North American office
Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75067
USA
Telephone: 940 455 7050

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

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

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

Printed in England.

Finding DB2 information using the Internet

The data processing world is increasingly becoming an on-line world. This phenomenon is being driven by the Internet. Let's examine some of the best places to look for DB2 information on the Internet.

When discussing the Internet, most people limit themselves to the World Wide Web. However, there are many components that make up the Internet. For the purpose of this article, I will discuss the three primary components most useful to DB2 professionals – the World Wide Web, mailing lists, and Usenet Newsgroups.

THE WORLD WIDE WEB

The World Wide Web (WWW) uses a graphical interface and hypertext protocol to display information in a point-and-click environment. Using a Web browser, users can navigate through the Internet, accessing Web pages and FTP and gopher sites.

There are many Web pages providing useful DB2 information. Foremost, of course, is IBM's DB2 Family Web page at http://www.software.ibm.com/data/db2. From this page, you will be able to access DB2 version and release information, technical information, DB2 manuals on-line, and add-on product information. Another useful IBM site is the redbook site. IBM's International Technical Support Organization (ITSO) publishes many books on technical topics. The IBM ITSO Redbook site can be accessed at http://www.redbooks.ibm.com/redbooks. It provides a searchable on-line catalogue and the ability to order redbooks directly from IBM over the Web.

Three other Web sites that you should visit and bookmark are Ron Raberd's DB2 reference site, Eric Loriaux's MVS site, and the JED-SP S/390 site. Ron Raberd's DB2 reference page can be reached at http://www.webcom.com/~raberd/db2info.html. Eric Loriaux's MVS site can be reached at http://www.ping.be/~ping1475. The System/390 home page is at http://www.jedsp.com/s390.

Many DB2 experts and consultants have their own Web sites that contain useful tips, tricks, and techniques, as well as their speaking schedules and copies of their presentations. One of the best of these sites is Richard Yevich's RYC Inc site, which can be reached at http://www.ryci.com. Another useful page is the database tips site that can be reached at http://www.platinum.com/dbtips. It contains useful DB2 tips, tricks, and guidelines.

Several of the DB2 user groups also have Web sites. These sites contain many useful DB2 resources, such as meeting schedules, newsletters, DB2 tips, and presentations. The International DB2 User Group (IDUG) Web site is one that every DB2 professional should visit regularly. It contains information on forthcoming conferences, as well as an on-line version of its DB2-related magazine, *IDUG Solutions Journal*. The IDUG Web site can be reached at http://www.idug.org.

Finally, most of the third-party DB2 tool vendors also have Web sites that contain information on DB2 and their product offerings.

MAILING LISTS

Mailing lists are a sort of community bulletin board. You can think of mailing lists as equivalent to a mass mailing. There are around 40,000 mailing lists available on the Internet, and they operate using a list server. A list server is a program that automates the mailing list subscription requests and messages. The two most common list servers are Listserv and Majordomo. Listserv is also a common synonym for a mailing list, but it is actually the name of a particular list server program.

If you subscribe to a mailing list, information is sent directly to your e-mail in-box. After subscribing, articles will begin to arrive in your e-mail box.

To subscribe to a mailing list, simply send an e-mail to the appropriate subscription address requesting a subscription. The DB2 mailing list can be subscribed to by sending a message to the subscription address, LISTSERV@AMERICAN.EDU. The message should read as follows:

```
SUBSCRIBE DB2-L
```

After issuing the preceding command, the list server will send you a message asking you to confirm the subscription. When you do so, information will quickly begin flowing into your e-mail box (perhaps at a much faster rate than you can reasonably digest). Literally hundreds of messages may be sent to you every week. To sign off from the newsgroup, send the following message to the same subscription address:

```
SIGNOFF DB2-L
```

In addition to a subscription address, mailing lists also have a posting address. This is the address to which mailing list posts must be sent. You should never send subscription requests to the list's posting address, nor send post to the subscription address.

The posting address for the DB2-L mailing list is DB2-L@AMERICAN.EDU. When a message is sent to this address, it will automatically be forwarded to everyone currently subscribing to the list.

You may also want to consider digesting your DB2 mailing list e-mails. A digest is an accumulation of the day's messages sent as one big e-mail. The benefit of digesting is that instead of receiving multiple daily messages from a mailing list, only one daily digest is sent. Because the DB2 list is usually quite active, you may receive dozens of e-mails daily if you do not choose the digest option.

To request digesting, simply send an e-mail to the subscription address. The digest request must be made after you have successfully subscribed to the mailing list.

For the DB2 mailing list, send the following message to the subscription address, LISTSERV@AMERICAN.EDU:

```
SET DB2-L DIGEST
```

The drawbacks to digests are that threads can be hard to follow, it is more difficult to respond to messages, and they can become quite large.

Finally, contributions sent to the DB2 mailing list are automatically

archived. The archive can be searched to find e-mails that were posted to the mailing list in the past. You can get a list of the available archive files by sending the following command to LISTSERV@AMERICAN.EDU:

```
INDEX DB2-L
```

The files returned can be ordered using the following command:

```
GET DB2-L LOGxxxx
```

USENET NEWSGROUPS

Another very fertile source of information on the Internet is found in various Usenet Newsgroups. Usenet, an abbreviation for User Network, is a large collection of discussion groups called newsgroups. Each newsgroup is a collection of articles pertaining to a single pre-determined topic. Newsgroup names usually reflect their focus. For

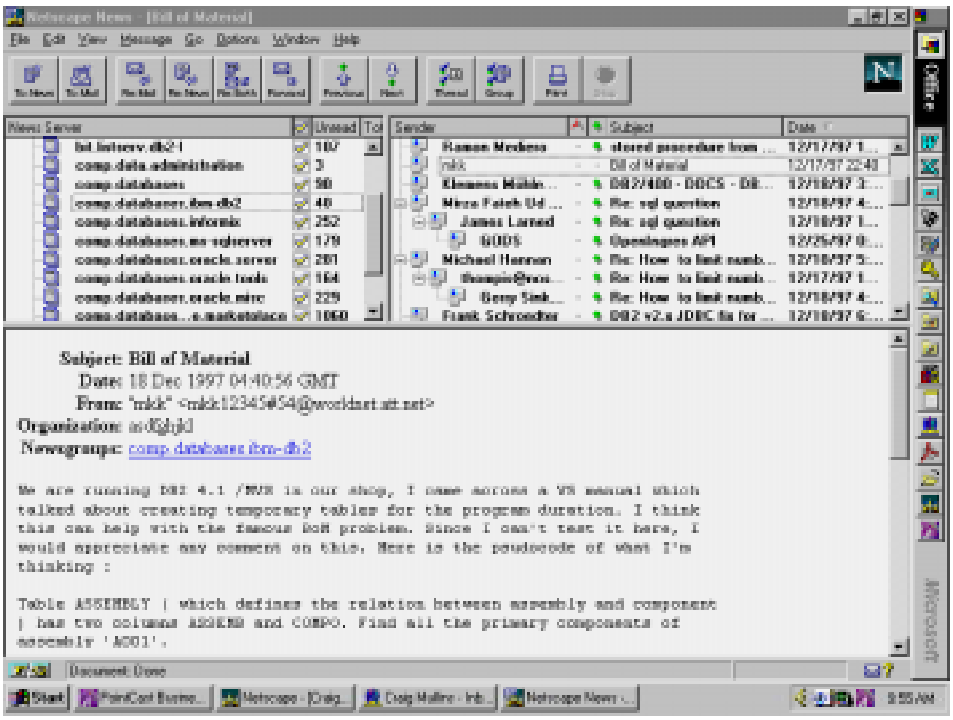


Figure 1: A Newsgroup posting

Newsgroup name	Description
comp.client-server	Information on client/server technology
comp.compression.research	Research on data compression techniques
comp.databases	Issues regarding databases and data management
comp.databases.ibm-db2	Information on IBM's DB2 family of products
comp.databases.informix	Information on the Informix DBMS
comp.databases.ingres	Information on the CA-Ingres DBMS
comp.databases.object	Information on object-oriented database systems
comp.databases.oracle.server	Information on the Oracle DBMS
comp.databases.sybase	Information on the Sybase DBMS
comp.databases.theory	Discussions on database technology and theory
comp.edu	Computer science education
comp.infosystems	General discussion of information systems
comp.misc	General computer-related topics
comp.os.os2.announce	OS/2 related announcements
comp.os.os2.apps	Information on OS/2 applications
comp.unix.admin	Unix administration discussions
comp.unix.aix	Information on IBM AIX
comp.unix.questions	Question and answer forum for Unix novices
comp.unix.solaris	Information pertaining to Sun Solaris
bit.listserv.aix-1	Information pertaining to AIX
bit.listserv.appc-1	Information pertaining to APPC
bit.listserv.cics-1	Information pertaining to CICS
bit.listserv.db2-1	Information pertaining to DB2
bit.listserv.ibm-main	IBM mainframe newsgroup
bit.listserv.os2-1	Information pertaining to OS/2
bit.listserv.power-1	Information pertaining to RS/6000s
bit.listserv.sqlinfo	Information pertaining to SQL/DS (DB2 for VSE & VM)

Figure 2: Useful Usenet Newsgroups

example, comp.databases.ibm-db2 contains discussions about the DB2 family of products.

Using most Web browsers or specialized news reader software, any Internet user can access a newsgroup and read the information it contains. Refer to Figure 1 for an example of a newsgroup posting to comp.databases.ibm-db2.

There are newsgroups available to satisfy just about every interest, particularly DB2. There are three primary newsgroups that DB2 users can access for DB2 news and information:

- comp.databases
- bit.listserv.db2-l
- comp.databases.ibm-db2.

Generic database information can be found on the comp.databases newsgroup. Some DB2 users post questions, comments, and information to this newsgroup because, for a long time, there was no newsgroup devoted to DB2. The only other option was to use the DB2 mailing list.

The bit.listserv.db2-1 newsgroup is very active with DB2 discussions and information. However, this newsgroup is a copy of the DB2 mailing list. If you subscribe to the mailing list, the information in this newsgroup will not be new.

The third, and newest, newsgroup is comp.databases.ibm-db2. This was instituted in early 1995 to offload the DB2 traffic from the comp.databases newsgroup and to provide a dedicated newsgroup for DB2 users. However, the postings to this newsgroup predominantly pertain to the DB2 Universal Database (as opposed to DB2 for OS/390). Other Usenet Newsgroups that may be of interest to DB2 users are listed in Figure 2.

Craig S Mullins (USA)

Generation of unload and load utilities

INTRODUCTION

There are times when an UNLOAD is the best back-up a DBA can have. Often a table must be dropped in order to perform some alteration. Dropping a table invalidates the image copies associated with it, so a LOAD is required for the table to be reloaded once it is recreated.

I sometimes encounter situations similar to this except that I want to unload all the tables in a database. This is a particularly frequent occurrence in development environments where a large batch of database changes would require many individual ALTERs if done one at a time. These changes are much easier to do by editing the SQL script that creates the database, unloading all the tables, dropping the database, running the script to recreate the database, and then reloading the tables.

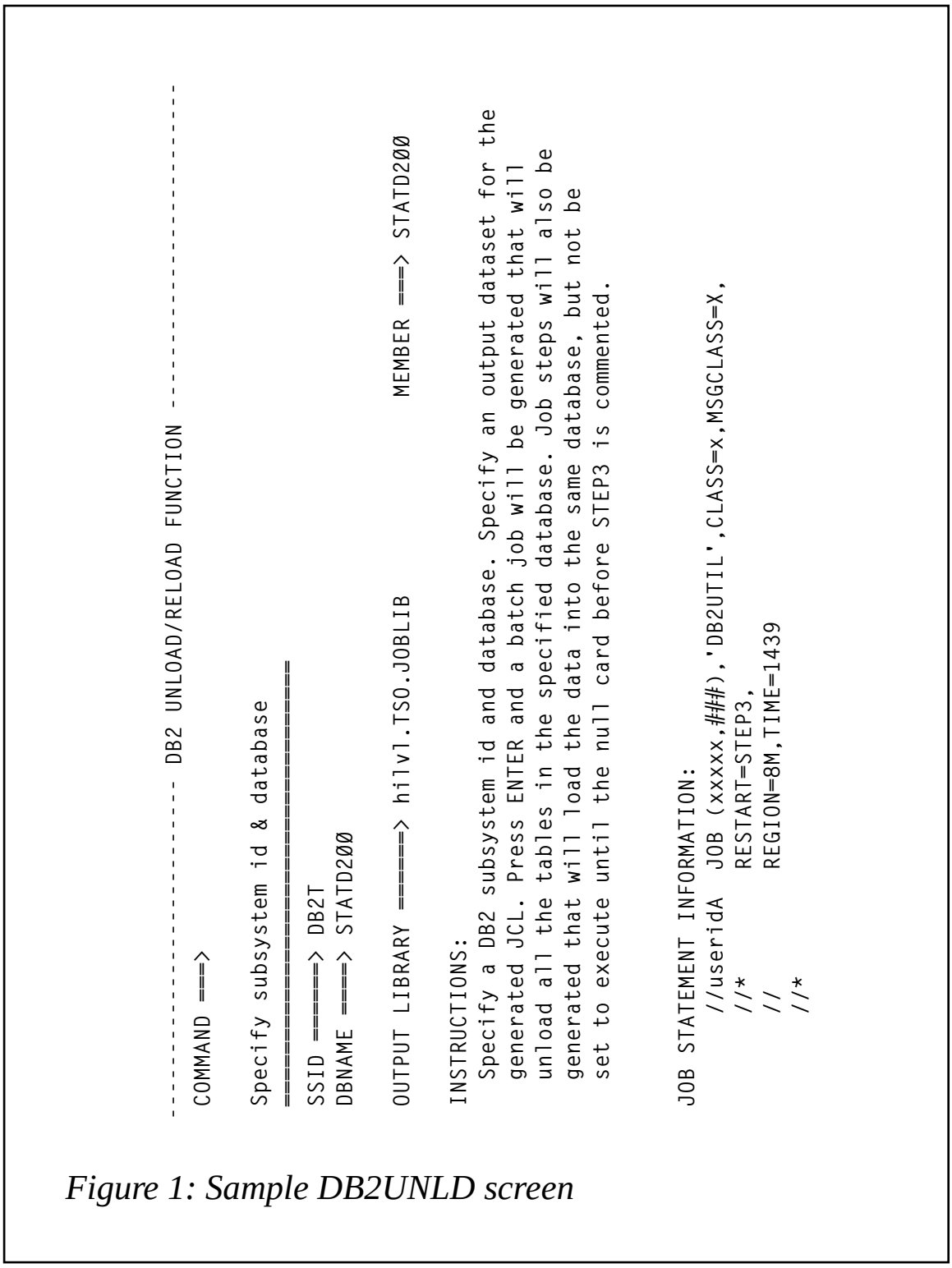
THE DB2UNLD FUNCTION

I created the DB2UNLD function to help me unload and reload all tables in a database. It consists of a REXX EXEC, DB2UNLD, and an ISPF panel, PDB2UNLD. The panel requires the entry of a valid DB2 subsystem-id and a database in that subsystem. The REXX EXEC generates the JCL that will perform an UNLOAD on every table in that database, and a LOAD on the same tables using modified load control cards and unload datasets from the UNLOAD step. A null card is inserted in the JCL after the UNLOAD step to prevent the LOAD steps from executing. This way the JCL can be run twice – once to unload the data and a second time to reload the data.

I used Open Software Technologies' REXXTOOLS/MVS product for executing the SQL statement in the DB2UNLD REXX EXEC. There are other methods of executing SQL statements from within a REXX EXEC – *DB2 Update* has published several articles describing REXX/DB2 interfaces that could be used instead.

RESULTS

This function saves a lot of time. I use it for generating extra back-ups during those times when I feel it is good insurance to have them. I also



```
------------------------- DB2 UNLOAD/RELOAD FUNCTION -------------------------

COMMAND ===>

Specify subsystem id & database
===============================
SSID ========> DB2T
DBNAME =====> STATD200

OUTPUT LIBRARY ========> hilvl.TSO.JOBLIB          MEMBER ===> STATD200

INSTRUCTIONS:
   Specify a DB2 subsystem id and database. Specify an output dataset for the
   generated JCL. Press ENTER and a batch job will be generated that will
   unload all the tables in the specified database. Job steps will also be
   generated that will load the data into the same database, but not be
   set to execute until the null card before STEP3 is commented.

JOB STATEMENT INFORMATION:
      //useridA  JOB (xxxxx,###),'DB2UTIL',CLASS=x,MSGCLASS=X,
      //*         RESTART=STEP3,
      //          REGION=8M,TIME=1439
      //*
```

Figure 1: Sample DB2UNLD screen

use it frequently when making mass changes to development databases by dropping and recreating the entire database.

Figure 1 shows a sample DB2UNLD screen.

DB2UNLD REXX EXEC

```
/* REXX ***************************************************************/
/*                                                                    */
/* DB2UNLD  - DB2 unload/reload function                              */
/*                                                                    */
/********************************************************** REXX */

  address "ISPEXEC" "CONTROL ERRORS RETURN"

  address "ISPEXEC" "VGET (bjc1 bjc2 bjc3 bjc4) PROFILE"
  address "ISPEXEC" "VGET (db2s dbunld) PROFILE"
  address "ISPEXEC" "VGET (jdsn jmem) PROFILE"

do forever
  address "ISPEXEC" "DISPLAY PANEL(PDB2UNLD)"

  if rc > Ø then exit

  address "ISPEXEC" "CONTROL DISPLAY LINE START(27)"

  address TSO
  output_file = "'"JDSN33"("33JMEM33")'"
  "ALLOC DD(OUTPUT) DSN("output_file") SHR REUSE"

  if db2s = 'DB2T' then
    do
      if dsnali("OPEN", "DB2T", "REXXTOOL") <> Ø then do
        say "Open for plan failed.  RC="rc" REASON="reason
        exit rc
      end
    end
  else
    if db2s = 'DB2P' then
      do
        if dsnali("OPEN", "DB2P", "REXXTOOL") <> Ø then do
          say "Open for plan failed.  RC="rc" REASON="reason
          exit rc
        end
      end
    else
      do
        say 'Invalid DB2 subsystem specified'
```

```
      say 'Check it and try again'
      iterate
    end

address SQL
"SELECT CREATOR, NAME, CARD, RECLENGTH, TSNAME",
  "FROM SYSIBM.SYSTABLES",
 "WHERE DBNAME='"dbunld"'",
   "AND TYPE='T'",
 "ORDER BY CREATOR, NAME"
if sqlca.sqlrows > 100 then
  do
    say 'More than 100 tables in this database'
    say 'See your DBA'
    iterate
  end

if dsnali("CLOSE", "SYNC") <> 0 then do
   say "Close for plan failed.  RC="rc" REASON="reason
   exit rc
end

count = 0
do while count < sqlca.sqlrows
  count = count + 1
  space_pri = card.count * reclength.count % 50000 % 15
  if space_pri < 1 then space_pri = 1
  space_sec = space_pri % 10
  if space_sec < 1 then space_sec = 1
end

queue bjc1
queue bjc2
queue bjc3
queue bjc4
queue "//*************************************************"
queue "//*                                               "
queue "//* THIS JOB WILL UNLOAD ALL TABLES IN DATABASE:  "
queue "//* "db2s" "dbunld".                              "
queue "//*                                               "
queue "//* AND WILL RELOAD THE TABLES IF THE NULL CARD   "
queue "//* BEFORE STEP3 IS COMMENTED OUT.                 "
queue "//*                                               "
queue "//* THIS JOB IS DESIGNED TO BE RUN IN 2 PARTS:    "
queue "//* STEP1-2 BEFORE THE TABLES HAVE BEEN DROPPED &  "
queue "//* RECREATED, STEP3-6 AFTER THE TABLES HAVE BEEN  "
queue "//* DROPPED & RECREATED.                           "
queue "//*                                               "
queue "//*************************************************"
queue "//JOBLIB   DD DSN=hilvl.SDSNLOAD,DISP=SHR"
```

```
queue "//************************************************"
queue "//* Unload all tables in database "dbunld
queue "//************************************************"
queue "//STEP1    EXEC PGM=IKJEFT01,"
queue "//         DYNAMNBR=20,"
queue "//         REGION=7M,"
queue "//         TIME=1439"
queue "//SYSPRINT DD SYSOUT=*"
queue "//SYSTSPRT DD SYSOUT=*"
queue "//SYSPUNCH DD DSN=&SYSUID..SYSPUNCH,"
queue "//            DISP=(NEW,CATLG,DELETE),"
queue "//            UNIT=SYSDA,"
queue "//            DCB=(LRECL=80,BLKSIZE=8000,RECFM=FB),"
queue "//            SPACE=(CYL,(1,1),RLSE)"
count = 0
do while count < sqlca.sqlrows
  count = count + 1
  space_pri = card.count * reclength.count % 50000 % 15
  if space_pri < 1 then space_pri = 1
  space_sec = space_pri % 10
  if space_sec < 1 then space_sec = 1
  chrcnt = count - 1
  if chrcnt < 10 then chrcnt = "0"33chrcnt
  queue "//SYSREC"chrcnt" DD DSN=&SYSUID..TAB"chrcnt".DATA,"
  queue "//            UNIT=SYSDA,"
  queue "//            DISP=(NEW,CATLG,DELETE),"
  queue "//            SPACE=(CYL,("space_pri","space_sec"),RLSE)"
end
queue "//SYSTSIN  DD *"
queue "  DSN SYSTEM("db2s") RETRY(0) TEST(0)"
queue "  RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) -"
queue "  LIB('"db2s".RUNLIB.LOAD')"
queue "  END"
queue "/*"
queue "//SYSIN    DD *"
count = 0
do while count < sqlca.sqlrows
  count = count + 1
  queue "  "STRIP(creator.count)"."name.count
end
queue "/*"
queue "//************************************************"
queue "//* Modify load control cards: REPLACE ENFORCE NO"
queue "//************************************************"
queue "//STEP2    EXEC PGM=IRXJCL,PARM='DB2UNLDE'"
queue "//SYSTSPRT DD SYSOUT=*"
queue "//SYSEXEC  DD DSN=SYS1.TSO.DB2.CLIST,DISP=SHR"
queue "//INFILE   DD DSN=&SYSUID..SYSPUNCH,DISP=SHR"
queue "//OUTFILE  DD DSN=&SYSUID..SYSPUNCH.MODIFIED,"
queue "//            DISP=(NEW,CATLG,DELETE),"
```

```
queue "//            UNIT=SYSDA,"
queue "//            DCB=(LRECL=8Ø,BLKSIZE=8ØØØ,RECFM=FB),"
queue "//            SPACE=(CYL,(1,1),RLSE)"
queue "//"
queue "//*************************************************"
queue "//* Terminate utility (for rerunability)"
queue "//*************************************************"
queue "//STEP3    EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø"
queue "//STEPLIB  DD DSN=hilvl.SDSNLOAD,DISP=SHR"
queue "//SYSPRINT DD SYSOUT=*"
queue "//SYSTSPRT DD SYSOUT=*"
queue "//SYSUDUMP DD SYSOUT=*"
queue "//SYSTSIN  DD *"
queue "DSN SYSTEM ("db2s")"
queue "-TERM UTIL("sysvar(sysuid)".DB2UNLD)"
queue "END"
queue "/*"
queue "//*************************************************"
queue "//* Load all tables in database "dbunld
queue "//*************************************************"
queue "//STEP4    EXEC PGM=DSNUTILB,"
queue "//         COND=(4,LT),REGION=7M,"
queue "//         PARM=('"db2s"','"sysvar(sysuid)".DB2UNLD','')"
queue "//STEPLIB  DD DSN=hilvl.SDSNLOAD,DISP=SHR"
queue "//DSNTRACE DD SYSOUT=*"
queue "//UTPRINT  DD SYSOUT=*"
queue "//SYSPRINT DD SYSOUT=*"
queue "//SYSUDUMP DD SYSOUT=*"
queue "//SORTOUT  DD SPACE=(CYL,(5Ø,5Ø)),UNIT=SYSDA"
queue "//SYSUT1   DD SPACE=(CYL,(5Ø,5Ø)),UNIT=SYSDA"
queue "//SYSMAP   DD SPACE=(CYL,(5Ø,5Ø)),UNIT=SYSDA"
count = Ø
do while count < sqlca.sqlrows
  count = count + 1
  chrcnt = count - 1
  if chrcnt < 1Ø then chrcnt = "Ø"33chrcnt
  queue "//SYSREC"chrcnt" DD DISP=SHR,DSN=&SYSUID..TAB"chrcnt".DATA"
end
queue "//SYSIN    DD DSN=&SYSUID..SYSPUNCH.MODIFIED,DISP=SHR"
queue "//*************************************************"
queue "//* Start force all tablespaces"
queue "//*************************************************"
queue "//STEP5    EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø"
queue "//STEPLIB  DD DSN=hilvl.SDSNLOAD,DISP=SHR"
queue "//SYSPRINT DD SYSOUT=*"
queue "//SYSTSPRT DD SYSOUT=*"
queue "//SYSOUT   DD SYSOUT=*"
queue "//SYSUDUMP DD SYSOUT=*"
queue "//SYSTSIN  DD *"
queue "DSN SYSTEM ("db2s")"
```

```
count = Ø
do while count < sqlca.sqlrows
  count = count + 1
  queue "-STA DB("dbunld") SPACENAM("tsname.count") ACCESS(FORCE)"
end
queue "END"
queue "/*"
queue "//************************************************"
queue "//* Delete all work datasets"
queue "//************************************************"
queue "//STEP6    EXEC PGM=IEFBR14"
queue "//SYSOUT   DD SYSOUT=*"
queue "//SYSPRINT DD SYSOUT=*"
count = Ø
do while count < sqlca.sqlrows
  count = count + 1
  chrcnt = count - 1
  if chrcnt < 1Ø then chrcnt = "Ø"33chrcnt
  queue "//DD"chrcnt"     DD DSN=&SYSUID..TAB"chrcnt".DATA,"
  queue "//            DISP=(OLD,DELETE,DELETE)"
end
queue "//DDPUNCH  DD DSN=&SYSUID..SYSPUNCH,"
queue "//            DISP=(OLD,DELETE,DELETE)"
queue "//DDPUNCHM DD DSN=&SYSUID..SYSPUNCH.MODIFIED,"
queue "//            DISP=(OLD,DELETE,DELETE)"
queue "//"

address TSO
how_many = queued()
"EXECIO "how_many" DISKW OUTPUT ( FINIS"
"FREE DD(OUTPUT)"

address "ISPEXEC"
"LMINIT DATAID(ID1) DATASET('"jdsn"') ENQ(SHRW)"
IF RC > Ø THEN
  DO
    ZEDSMSG = "LMINIT "33rc
    address "ISPEXEC" "SETMSG MSG(ISRZØØ1)"
    EXIT RC
  END
"EDIT DATAID("ID1") MEMBER("jmem")"
IF RC > 4 THEN
  DO
    ZEDSMSG = "EDIT "33rc
    address "ISPEXEC" "SETMSG MSG(ISRZØØ1)"
    EXIT RC
  END

address "ISPEXEC" "VPUT (db2s dbunld) PROFILE"
address "ISPEXEC" "VPUT (jdsn jmem) PROFILE"
```

```
  address "ISPEXEC" "VPUT (bjc1 bjc2 bjc3 bjc4) PROFILE"

end
```

PDB2UNLD ISPF PANEL

```
%-------------------------- DB2 UNLOAD/RELOAD FUNCTION -----------------
-------
%COMMAND ===>_ZCMD
+
+
%Specify subsystem id & database
%===================================
+SSID%======>_DB2S+
+DBNAME%====>_DBUNLD  +
+
+OUTPUT LIBRARY%======>_JDSN                          + MEMBER%===>_JMEM
+
+
%INSTRUCTIONS:
+  Specify a DB2 subsystem-id and database. Specify an output dataset
+  for the generated JCL. Press ENTER and a batch job will be generated
+  that will unload all the tables in the specified database. Job steps
+  will also be generated that will load the data into the same
+  database, but not be set to execute until the null card before STEP3
+  is commented.
+
+
%JOB STATEMENT INFORMATION:
+      _BJC1
+      _BJC2
+      _BJC3
+      _BJC4
)INIT
  .CURSOR = ZCMD
  &ZCMD = ' '
)PROC
  VER (&ZCMD LIST,END,' ')
  VER (&DB2S NONBLANK LIST,DB2T,DB2P)
  VER (&DBUNLD NONBLANK NAME)
  VER (&BJC1 NONBLANK)
  VER (&BJC2 NONBLANK)
  VER (&BJC3 NONBLANK)
  VER (&BJC4 NONBLANK)
  VER (&JDSN NONBLANK)
  VER (&JMEM NONBLANK NAME)
)END
```

DB2UNLDE REXX EXEC

```
/* REXX ***************************************************************/
/*                                                                    */
/* DB2UNLDE - MODIFY LOAD CONTROL CARDS                               */
/*            Add REPLACE & ENFORCE NO parms.                        */
/*                                                                    */
/********************************************************** REXX */
  arg tcreator tname

  "alloc ddname(INFILE) shr reuse"
  "alloc ddname(OUTFILE) shrw reuse"

  "newstack"

   eof_sw = 'n'

   call read_input

   do while eof_sw = 'n'
     if substr(record.1,13,6) = 'LOG NO' then
       do
         tempstr3 = substr(record.1,2Ø,3Ø)
         tempstr2 = 'REPLACE ENFORCE NO '
         tempstr1 = substr(record.1,1,19)
         record.1 = tempstr133tempstr233tempstr3
       end
     call write_output
     call read_input
   end

   "execio Ø diskr INFILE (finis"        /* close input file          */
   "execio Ø diskw OUTFILE (finis"       /* close output file         */
   "free ddname(INFILE)"                 /* free input file           */
   "free ddname(OUTFILE)"                /* free output file          */

/*********************************************************************/
/*  read 1 input record                                              */
/*********************************************************************/
read_input:
  "execio 1 diskr INFILE (STEM record."
  if rc <> Ø then eof_sw = 'y'
  return
/*********************************************************************/
/*  write 1 output record                                            */
/*********************************************************************/
write_output:
  "execio 1 diskw OUTFILE (STEM record."
  return
```

Accessing DB2 data over the Internet

Enabling Web-based access to corporate data stored in DB2 makes the data more readily accessible to more people. Companies can obtain a competitive advantage by making their data available to employees over an intranet, or to customers and partners over an extranet.

An intranet is a special Internet adaptation that can only be accessed by internal employees. Likewise, an extranet extends the accessibility outside the corporation in a secure manner only to authorized individuals.

IBM provides two options for accessing DB2 data over the Web – DB2 WWW and Net.Data.

USING THE DB2 WWW CONNECTION

DB2 WWW is an IBM product for connecting DB2 databases to the Web. Using a Web browser and DB2 WWW, companies can use the Internet as a front end to DB2 databases – data stored in DB2 tables is presented to users in the style of a Web page. This lets users familiar with the Internet quickly come up to speed at accessing DB2 data. DB2 WWW includes a procedural language to map between standard HTML and SQL, as well as a full-function graphing engine to return results to the Web browser in the form of mixed text and graphics.

DB2 WWW provides two-tier and three-tier client/server environments. In a two-tier environment, the database resides on the Internet server and client Web browsers access the data. For DB2 running on OS/390, this is applicable only if you use the mainframe as your Internet server. In a three-tier environment, the data can reside on both the local Internet server and a remote platform. This requires DDCS (Distributed Database Connection Services), CAE (Client Application Enabler), or DataJoiner. The three-tier set-up is useful when your Internet server is a Unix or Windows NT machine and you need to access DB2 data from the mainframe.

DB2 WWW uses a CGI run-time engine, which processes the input

from HTML forms on the Web and sends SQL statements to a DB2 WWW application. This application consists of a macro file containing HTML input and report form definitions, SQL statements, and variable definitions. The application user sees only the Web page using the Web browser of his or her choice. To the end user, the application

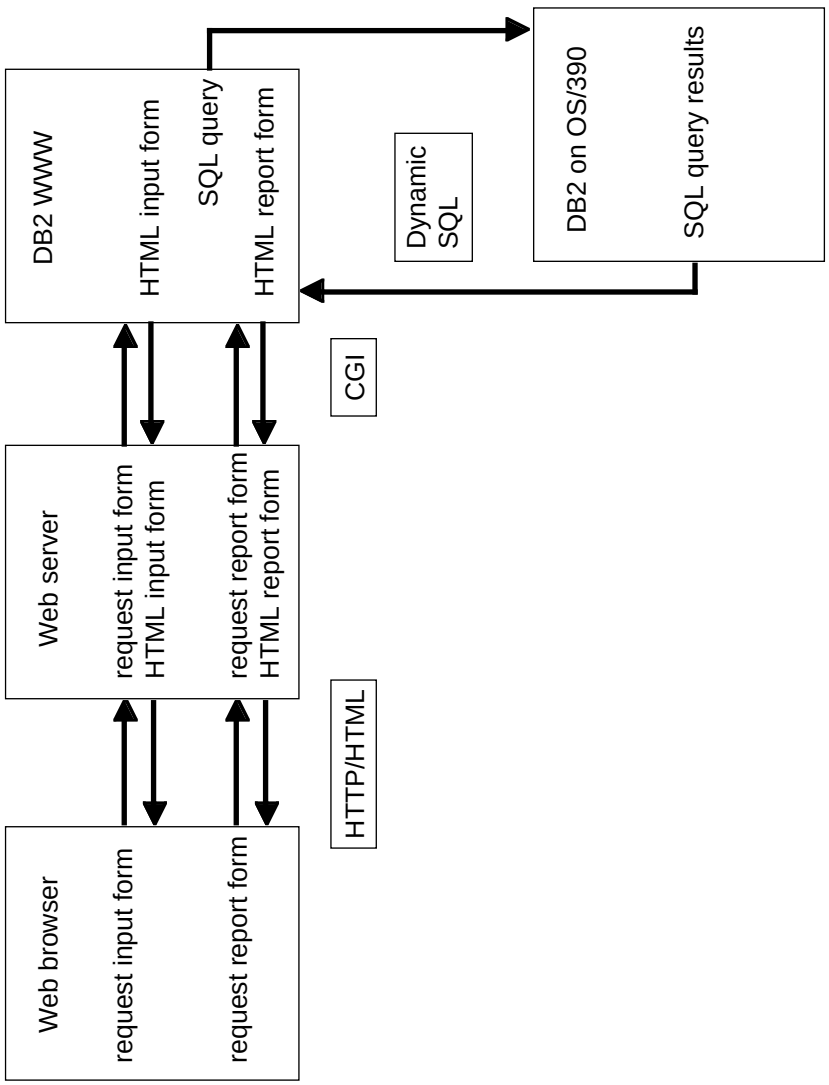


Figure 1: How DB2 WWW works

functions just like any other Web page. This is illustrated in Figure 1.

Because DB2 WWW applications use native HTML and SQL, developers do not need to learn complex new languages and syntax to connect DB2 databases to the Web. Furthermore, SQL SELECT, INSERT, UPDATE, and DELETE statements are supported for both data query and modification.

NET.DATA

Net.Data, another IBM product, is an upwardly-compatible follow-on version to DB2 WWW. DB2 WWW applications are compatible with Net.Data (but not necessarily *vice versa*). Net.Data enhances the functionality of DB2 WWW in two ways.

The first is as a CGI application. In this case, it is invoked like DB2 WWW. The second is as an API application. In this case, the server calls Net.Data as a DLL (Dynamic Linked Library) or shared library.

Net.Data supports Java by enabling calls to Java applets and JavaScripts for client-side processing. Using Java, you can create dynamic, complex Web-based applications. An additional benefit of Net.Data's support for JavaScripts is that data can be verified at the client's Web browser as it is entered, instead of on the server. This can enhance performance by reducing network traffic.

For additional server-side processing, your Web application can call scripts and functions written in SQL, Perl, REXX, or C/C++. Additionally, an API is available to extend Net.Data functionality.

Net.Data ships free with DB2 Version 5.

INTERNET ACCESS GUIDELINES

When accessing DB2 data over the Internet, consider the following helpful tips and techniques:

1 Design Web applications with the user in mind.

Be aware that the equipment on which you are developing your

Web-based applications is probably more state-of-the-art than the equipment on which the application will be used. It is common for developers to have access to high resolution monitors and a lot of memory. Be sure to test the application on PC set-ups with less memory and on monitors of varying dot pitch and resolution.

2 Plan your security requirements.

When developing DB2 applications that are accessible using the Internet, be sure to plan adequate security into the application. DB2 WWW and Net.Data each provide authorization features that should be utilized to ensure that only authorized users are permitted access.

3 Consider Net.Data live connections.

Before a query can be executed, the process must identify itself and connect to DB2. This can cause performance problems.

Net.Data can be used to establish a live connection by continuously running processes to perform the start-up tasks. Once started, the process waits to execute subsequent requests.

Live connections are required for API connections, but can be used for CGI connections, too.

SYNOPSIS

The Internet is infiltrating every aspect of information technology. And DB2 is most definitely impacted. Surely, an increasing number of new and existing DB2 databases will be hooked up to the Web. Learning how to accomplish DB2 Web enablement today can help to ensure continuing effective DB2 data access at your company.

Craig S Mullins (USA)

Collecting accounting information

PROBLEM

In many DB2 installations, the SMF accounting information being collected is traditionally at the plan level. However, as more applications are being written to take advantage of packages, or when multiple DBRMs share a single plan, there is a growing need to collect and report the accounting infomation at the package or DBRM level.

Our DB2 shop decided to collect information at the package/DBRM level. However, we discovered that none of our existing tools (and we thought that we had a fairly good set) can readily produce reports which display this level of information. Although the data is there, our tools do not have the 'canned' reports to display them.

SOLUTION

To address this need, two programs were written to produce customized reports. These programs are described below:

- SMF101P1 – this reads the SMF daily dataset and extracts specific files from the accounting records and prints the detail report.
- SMF101P2 – this produces a summary report based on the detail data extracted by SMF101P1.

Figures 1 and 2 show sample reports from these programs.

SMF101P1

```
//YOUR JOB CARD HERE
//ASM     EXEC PGM=ASMA90,PARM='OBJECT,NODECK'
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//         DD  DSN=SSID.DB2.DSNMACS,DISP=SHR
//         DD  DSN=SSID.DB2.DSNSAMP,DISP=SHR
//SYSLIN   DD  DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(800,(1000,1000)),DCB=(BLKSIZE=800)
//SYSPRINT DD  SYSOUT=*
```

```
PLANNAME DATE     TIME     TPNN PACKAGEID/DBRMNAME      ELAPSED     TCB-CPU
-------- -------- -------- ---- -------------------- ------------ ------------
DSNTIAUL 97.308 01:05:34 RM01 DSNTIAUL                   13.064        7.822
DSNTIAUL 97.308 01:05:44 RM01 DSNTIAUL                    5.270        3.064
CMNPLAN  97.308 01:11:31 PK01 CMNDB2SQ                    1.045         .438
CMNPLAN  97.308 04:06:26 PK01 CMNDB2SQ                    4.053         .565
DSNTIAUL 97.308 06:02:58 RM01 DSNTIAUL                   94.276       32.010
CMNPLAN  97.308 06:04:38 PK01 CMNDB2SQ                    3.058         .742
QMF320   97.308 06:42:21 PK01 DSQ9RDBR                     .133         .002
QMF320   97.308 06:42:21 PK02 DSQ9DYSQ                     .521         .011
QMF320   97.308 06:42:21 PK03 DSQ9SUSR                     .104         .001

 ELAPSED     TCB-CPU     I/O-WAIT     L/L-WAIT   ASYNC-READ   SYNCH-READ
---------- ---------- ------------ ------------ ------------ ------------
    .615        .211         .305         .000         .048         .000
   1.084        .669         .045         .000         .030         .000
    .062        .021         .040         .000         .000         .000
    .086        .028         .053         .000         .000         .000
   1.934       1.083         .043         .000         .043         .000
    .158        .043         .043         .000         .044         .000
    .739        .373         .070         .000         .031         .000
   1.459       1.023         .037         .000         .038         .000
    .105        .035         .040         .000         .000         .000
```

Figure 1: Sample SMF101P1 report

Note: actual report is 132 characters in width.

```
DB2 ACCOUNTING SMF RECORDS - PACKAGE/DBRM
PLANNAME PACKAGEID/DBRMNAME TYPE     COUNT      ELAPSED      TCB-CPU
-------- ----------------- ---- ---------- ------------ ------------
CMNPLAN  CMNDB2SQ            PK         33     1.393        0.558
DSNESPCS DSNESM68            PK          2     1.573        0.119
DSNESPRR DSNESM68            PK         19     9.399        3.594
DSNTEP2  DSNTEP2             RM         41     4.957        0.976
DSNTIAUL DSNTIAUL            PK         21    11.200        5.368
F2PLN371 ADSQID00            PK          1     0.667        0.018
F2PLN371 CWSQLPRO            PK         12     7.582        1.045
F2PLN371 OSSQLCAT            PK         13     0.034        0.007
F2PLN371 OSSQLC41            PK         10     0.162        0.021

                    RUN DATE: 98.028  TIME: 15:40:33
    I/O-WAIT     L/L-WAIT    ASYNC-READ    SYNCH-READ
------------ ------------ ------------ ------------
       0.522        0.001        0.046        0.072
       0.112        0.003        1.324        0.001
       0.753        0.005        2.886        0.001
       0.221        0.008        2.125        0.134
       0.910        0.001        1.711        0.100
       0.104        0.000        0.020        0.440
       1.071        0.018        4.597        0.578
       0.009        0.000        0.000        0.004
       0.109        0.000        0.009        0.006
```

Figure 2: Sample SMF101P2 report

Note: actual report is 132 characters in width.

```
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  SPACE=(800,(1000,1000),,,ROUND),UNIT=SYSDA
//SYSIN    DD  *
*************************************************************
* SMF101P1 :   READ SMF ACCOUNTING RECORDS
*
* FUNCTION :   EXTRACT AND DISPLAY PACKAGE AND
*              DBRM LEVEL ACCOUNTING INFORMATION
*
* INPUT    :   SYSIN   DD - SELECTION CRITERIA TO FILTER RECORDS
*              SMFDATA DD - THE SMF DATASET COLLECTED FOR THE DAY
*
* OUTPUT   :   SMFSOUT DD - SYSOUT INFO ABOUT THE RUN
*              SMFDOUT DD - DETAIL REPORT OUTPUT
*              SMFSUMM DD - OUTPUT RECORD TO BE SUMMARIZED BY
*                           PROGRAM SMF101P2
* PROCESS FLOW:
*    INITIALIZE VALUES
*    GET SELECTION CRITERIA FROM SYSIN
*    PRINT COLUMN HEADERS
*    READ SMFDATA UNTIL END OF FILE
*       IF TYPE 101 RECORD
*          THEN PROCEED
*          ELSE SKIP THIS RECORD
*       IF SSID IS OK
*          THEN PROCEED
*          ELSE SKIP THIS RECORD
*       PROCESS RECORD
*          NAVIGATE TO VARIOUS SECTIONS:
*             QWA0  - SELF-DEFINING SECTION
*             QWHS  - MAIN HEADER   SECTION
*             QWHC  - CORRELATION HEADER SECTION
*             QWAC  - IFI ACCOUNTING SECTION
*             QPAC  - PACKAGE/DBRM ACCOUNTING SECTION
*                LOOP TO ALL THE QPAC TRACES
*             IF IFCID 239
*                THEN THIS IS AN OVERFLOW RECORD
*                PROCESS ACCORDINGLY
*       PRINT TO SMFDOUT DD
*       WRITE TO SMFSUMM DD
*   FINALIZE
*************************************************************
SMF101P1 CSECT                    ESTABLISH CSECT
         SAVE  (14,12),,SMF101P1&SYSDATE&SYSTIME
         LR    R11,R15             LOAD A(ENTRY POINT)
         USING SMF101P1,R11,R12
         GETMAIN RU,LV=WORKLEN     GET WORK AREA
         ST    R13,4(,R1)
         ST    R1,8(,R13)
```

```
         LR    R13,R1
         USING WORKAREA,R13
         LA    R12,2048(,R11)     SET UP SECOND BASE REGISTER
         LA    R12,2048(,R12)     SET UP SECOND BASE REGISTER
         B     INITIALZ           GO AROUND SAVEAREA
***************************************** INITIALIZATION ******
**       I N I T I A L I Z E
INITIALZ DS    0H
         BAL   R14,INITS000       INITIALIZE STUFF
         BAL   R14,GETSY000       GET PASSED PARM
         BAL   R14,COLHD000       PRINT COLUMN HEADERS
***************************************** MAINLINE     ********
**       M A I N L I N E
MAINLINE DS    0H
         BAL   R14,READS000       READ SMF RECORD
         CLI   READSFLG,X'FF'     IF END OF FILE?
         BE    FINALIZE              THEN FINALIZE
*                                    ELSE ...
         BAL   R14,CHECK000       CHECK RECTYPE AND SSID
         CLI   CHECKFLG,X'FF'     IF RECORD NOT OK
         BE    MAINLINE              THEN READ NEXT RECORD
*                                    ELSE ...
         BAL   R14,PROCE000       PROCESS THIS RECORD
         B     MAINLINE           READ NEXT RECORD
***************************************** FINALIZATION ********
**       F I N A L I Z E
FINALIZE DS    0H
         BAL   R14,FINAL000
         LR    R1,R13             LOAD A(SAVEAREA) FOR FREEMAIN
         L     R13,4(,R13)        LOAD A(CALLERS SAVEAREA)
         FREEMAIN RU,LV=WORKLEN,A=(R1) FREE WORK AREA
         RETURN (14,12),RC=0      RETURN TO OS WITH RETCODE=0
***************************************** SUBROUTINES *********
**       S U B - R O U T I N E S
***************************************** INITS000 ************
** PROCESS THE ACCOUNTING RECORD
**    NAVIGATE TO VARIOUS SECTIONS
**    LOOP THROUGH THE QPAC SECTION BASED ON COUNT
**    IF THIS IS AN OVERFLOW RECORD (IFCID=239)
**       THEN PROCESS AS SUCH
PROCE000 DS    0H
         ST    R14,PROCESAV
         MVC   O3RECORD,=132CL1' '  CLEAR OUTPUT LINE
**
** REGISTER USAGE CONVENTION
** R3 IS THE BASE REGISTER FOR SM101
** R4 IS THE POINTER TO THE SELF DEFINING SECTION WHICH
**       CONTAINS OFFSETS TO VARIOUS SECTIONS
** R5 IS THE BASE REGISTER FOR THE CURRENT SECTION BEING
```

```
**       ACCESSED
**
PSM1Ø1   DS    ØH           SM1Ø1 SMF HEADER SECTION
         USING SM1Ø1,R3             ASSIGN THIS TO R3
PQWAØ    DS    ØH            QWAØ SELF DEFINING SECTION
         LA    R4,SM1Ø1END          POINT REG TO THIS SECTION
         USING QWAØ,R4              ASSIGN QWAØ TO R4
PQWHS    DS    ØH            QWHS HEADER (MAIN) SECTION
         LR    R5,R3                INIT R5 TO START OF RECORD
         A     R5,QWAØ1PSO          ADD THE OFFSET
         USING QWHS,R5              ASSIGN QWHS TO R5
         BAL   R14,FQWHSØØØ         FORMAT DATA
PQWHC    DS    ØH            QWHC HEADER (CORRELATION) SECTION
         LA    R5,QWHSEND           BUMP TO END OF HEADER SECTION
         USING QWHC,R5              ASSIGN QWHS TO R5
         BAL   R14,FQWHCØØØ         FORMAT DATA
PQWAC    DS    ØH            QWAC IFC ACCOUNTING SECTION
         LR    R5,R3                INIT R5 TO START OF RECORD
         A     R5,QWAØ1R1O          ADD THE OFFSET
         USING QWAC,R5              ASSIGN QWAC TO R5
         BAL   R14,FQWACØØØ         FORMAT DATA
         CLC   WHIFCID,=X'ØØEF'  IF IFCID=293
         BE    PQWAC239          THEN THIS IS AN OVERFLOW RECORD
         CLC   QWACPKGN,=H'Ø' IF PACK/DBRM COUNT = Ø
         BE    PROCE999
PQPAC    DS    ØH            QPAC PACKAGE/DBRM LEVEL ACCOUNTING
         LR    R5,R3                INIT R5 TO START OF RECORD
         A     R5,QWAØ1R8O          ADD THE OFFSET
         USING QPAC,R5              ASSIGN QPAC TO R5
         BAL   R14,FQPACØØØ         FORMAT PACKAGE/DBRM DATA
         B     PROCE999
PQWAC239 DS    ØH            QPAC PACKAGE/DBRM LEVEL ACCTG(OVERFLOW)
         LA    R4,SM1Ø1END
         USING QWA1,R4
         LR    R5,R3                INIT R5 TO START OF RECORD
         A     R5,QWA11R1O          ADD THE OFFSET
         USING QPKG,R5              ASSIGN QPKG TO R5
         SR    R9,R9
         MVC   WHPKRECN,QPKGPKGN
PQPAC239 DS    ØH            QPAC PACKAGE/DBRM LEVEL ACCTG(OVERFLOW)
         LR    R5,R3                INIT R5 TO START OF RECORD
         A     R5,QWA11R2O          ADD THE OFFSET
         USING QPAC,R5              ASSIGN QPAC TO R5
         BAL   R14,FQPACØØØ         FORMAT PACKAGE/DBRM DATA
         B     PROCE999
PROCE999 DS    ØH
         L     R14,PROCESAV
         BR    R14
PROCESAV DS    F'Ø'
```

```
************************************** FQWHS000 *************
** GET SMF HEADER DATA FROM QWHS SECTION
**     DATE, TIME AND IFCIDE
FQWHS000 DS    0H
         ST    R14,FQWHSSAV
         USING QWHS,R5              THIS IS THE TOP OF OF QWHS
         MVC   WORKFWD1,SM101TME    GET TIME
         MVC   WORKFWD2,SM101DTE    GET DATE
         BAL   R14,PDATE000         MAKEM PRINTABLE
         MVC   O3DATE,WORKCL07+1    MOVEM TO OUTAREA
         MVC   O3TIME,WORKCL10+2    MOVEM TO OUTAREA
         MVC   O1DATE,WORKCL07+1    MOVEM TO OUTAREA
         MVC   O1TIME,WORKCL10+2    MOVEM TO OUTAREA
         MVC   WHIFCID,QWHSIID      GET IFCID
FQWHS999 DS    0H
         L     R14,FQWHSSAV
         BR    R14
FQWHSSAV DS    F'0'
************************************** FQWHC000 *************
** GET CORRELATION HEADER DATA FROM QWHC SECTION
**     GET PLAN NAME
**     IF PLAN IS BLANK THEN THIS MUST BE A DB2 COMMAND
**     DISCARD DSNUTIL AND *COMMAND PLANS AS THESE DO NOT HAVE
**           PACKAGE/DBRM INFO
**     FILTER PLAN IF SPECIFIED BY THE USER
FQWHC000 DS    0H
         ST    R14,FQWHCSAV
         USING QWHC,R5              THIS IS THE TOP OF OF QWHC
FQWHC005 DS    0H
         MVC   O1PLAN,QWHCPLAN      GET PLAN
         MVC   O3PLAN,QWHCPLAN      GET PLAN
         CLC   O3PLAN,=C'        '  IF PLAN BLANK?
         BNE   FQWHC010             N. GO ON
         MVC   O3PLAN,=C'*COMMAND'  Y. REPLACE WITH THIS TEXT
         MVC   O4PLAN,O3PLAN
         CLC   O3PLAN,=C'DSNUTIL '  IF PLAN DSNUTIL
         B     MAINLINE             Y. SKIP IT
         CLC   O3PLAN,=C'*COMMAND'  IF PLAN *COMMAND
         B     MAINLINE             Y. SKIP IT
FQWHC010 DS    0H
         CLC   CRITPLAN,=C'        '  IF USER SPECIFIED A PLAN?
         BE    FQWHC020               N. GO ON
         CLC   CRITPLAN,O3PLAN        Y. IF PLAN MATCH
         BE    FQWHC020                  Y. GO ON
         B     MAINLINE                  N. GETNEXT RECORD
FQWHC020 DS    0H
         AP    COUNTER3,=P'1'       COUNT SELECTED RECORDS
FQWHC999 DS    0H
         L     R14,FQWHCSAV
```

```
         BR    R14
FQWHCSAV DS    F'0'
************************************* FQWAC000 *************
** GET IFI ACCOUNTING DATA FROM QWAC SECTION
**     GET THE NUMBER OF PACKAGES/DBRM INFO IN THIS RECORD
**     THIS NUMBER IS USED AS A LOOP COUNTER WHEN COLLECTING
**        DATA ABOUT THE PACKAGE/DBRM
FQWAC000 DS    0H
         ST    R14,FQWACSAV
         USING QWAC,R5              THIS IS THE TOP OF OF QWAC
         SR    R9,R9
         MVC   WHPKRECN,QWACPKGN
FQWAC999 DS    0H
         L     R14,FQWACSAV
         BR    R14
FQWACSAV DS    F'0'
************************************* FQPAC000 *************
** FORMAT PACKAGE/DBRM DATA FROM QPAC SECTION
**     LOOP THROUGH EACH OF THE PACKAGE/DBRM
**        UP TO MAX OF 10 PER RECORD
**        BEYOND 10 AN OVERFLOW RECORD IS CUT AS IFCID=239
**     THE DATA MAY BE A PACKAGE OR DBRM. MARK ACCORDINGLY
**     GET THE PACKAGE ID, LOCATION, COLLECTION ID
**     GET VARIOUS TIMES AND WAITS
**     CONVERT THE TIME FROM STCK VALUE TO THOUSANDTHS OF SECONDS
FQPAC000 DS    0H
         ST    R14,FQPACSAV
         USING QPAC,R5              THIS IS THE TOP OF OF QPAC
         LA    R8,10                INIT LOOP COUNTER TO 10
FQPAC001 SR    R9,R9
         TM    QPACFLGS+1,B'00000001' IF THIS IS A DBRM
         BO    FQPAC011               Y.
         TM    QPACFLGS+1,B'00000010' IF THIS IS A PAKG
         BO    FQPAC012               Y.
         MVC   O3TYPE,=C'  '
         B     FQPAC020
FQPAC011 MVC   O3TYPE,=C'RM'           MARK AS DBRM
         B     FQPAC020
FQPAC012 MVC   O3TYPE,=C'PK'           MARK AS PAKG
FQPAC020 LH    R9,QPACRECN          FORMAT RECORD NO
         CVD   R9,WORKDWD1
         MVC   WORKCL03,=X'212020'
         ED    WORKCL03,WORKDWD1+6
         MVC   O3RECN,WORKCL03+1
FQPAC030 MVC   O3PKID,QPACPKID      FORMAT PACKAGE/DBRM NAME
*        MVC   O3LOCN,QPACLOCN      FORMAT LOCATION NAME
*        MVC   O3COLN,QPACCOLN      FORMAT COLLECTION NAME
         STCKCONV STCKVAL=QPACSCT,                                     X
               CONVVAL=WORKCL16,                                       X
```

```
      TIMETYPE=DEC,                                                X
      DATETYPE=YYYYMMDD
BAL   R14,CONV2SEC
MVC   O3ACSCT,=X'2020202020202020214B202020'
ED    O3ACSCT,WORKPL8A+2
MVC   O4ACSCT,WORKPL8A
STCKCONV STCKVAL=QPACTJST,                                         X
      CONVVAL=WORKCL16,                                            X
      TIMETYPE=DEC,                                                X
      DATETYPE=YYYYMMDD
BAL   R14,CONV2SEC
MVC   O3ACTJST,=X'2020202020202020214B202020'
ED    O3ACTJST,WORKPL8A+2
MVC   O4ACTJST,WORKPL8A
STCKCONV STCKVAL=QPACAWTI,                                         X
      CONVVAL=WORKCL16,                                            X
      TIMETYPE=DEC,                                                X
      DATETYPE=YYYYMMDD
BAL   R14,CONV2SEC
MVC   O3ACAWTI,=X'2020202020202020214B202020'
ED    O3ACAWTI,WORKPL8A+2
MVC   O4ACAWTI,WORKPL8A
STCKCONV STCKVAL=QPACAWTL,                                         X
      CONVVAL=WORKCL16,                                            X
      TIMETYPE=DEC,                                                X
      DATETYPE=YYYYMMDD
BAL   R14,CONV2SEC
MVC   O3ACAWTL,=X'2020202020202020214B202020'
ED    O3ACAWTL,WORKPL8A+2
MVC   O4ACAWTL,WORKPL8A
STCKCONV STCKVAL=QPACAWTR,                                         X
      CONVVAL=WORKCL16,                                            X
      TIMETYPE=DEC,                                                X
      DATETYPE=YYYYMMDD
BAL   R14,CONV2SEC
MVC   O3ACAWTR,=X'2020202020202020214B202020'
ED    O3ACAWTR,WORKPL8A+2
MVC   O4ACAWTR,WORKPL8A
STCKCONV STCKVAL=QPACAWTE,                                         X
      CONVVAL=WORKCL16,                                            X
      TIMETYPE=DEC,                                                X
      DATETYPE=YYYYMMDD
BAL   R14,CONV2SEC
MVC   O3ACAWTE,=X'2020202020202020214B202020'
ED    O3ACAWTE,WORKPL8A+2
MVC   O4ACAWTE,WORKPL8A
MVC   PRNTLINE,O3RECORD
PUT   SMFSOUT,PRNTLINE
MVC   O4PLAN,O3PLAN
```

```
         MVC   O4PKID,O3PKID
         MVC   O4TYPE,O3TYPE
         PUT   SMFDOUT,O4RECORD
         MVC   O3RECORD,=132CL1' '
         MVC   O4RECORD,=8ØCL1' '
         CLC   QPACRECN,WHPKRECN    IF PKGNO => MAX PACKGES
         BNL   FQPAC999             Y. GET OUT OF LOOP
         LA    R5,QPACEND           N. BUMP TO NEXT PKG/DBRM
         MVC   O3PLAN,O1PLAN        REMEMBER PREV INFO
         MVC   O3DATE,O1DATE           DITTO
         MVC   O3TIME,O1TIME           DITTO
         BCT   R8,FQPAC ØØ1         LOOP IF LESS THAN 1Ø
FQPAC999 DS    ØH
         L     R14,FQPACSAV
         BR    R14
FQPACSAV DS    F'Ø'
*****************************************************************
** CONVERT TO SECONDS
** INPUT IS A PLØ8 'ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØC'
**            MASK 'XX XX DD HH MM SS TT TC'
**           WHERE        :  :  :  :  :--->THOUSANDTH OF A SEC
**                        :  :  :  :------>SECONDS
**                        :  :  :---------->MINUTES
**                        :  :------------->HOURS
**                        :--------------->DAYS
** OUTPUT IS  PLØ8 'SS SS SS SS SS SS TT TC'
CONV2SEC DS    ØH
         ST    R14,CONVSAVE
         MVC   WORKPL8A,WORKCL16          MOVE TO A PACKED FIELD
         MVI   WORKPL8A+7,X'ØC'           MAKE LAST BYTE PACKED
         SRP   WORKPL8A,58,5              SHFT 7 DIGITS RIGHT & ROUND
         CP    WORKPL8A,=P'6ØØØØ'         IF TIME IS MORE THAN 6Ø SECS
         BL    CONV2SEX                   N. GET OUT
*                                         Y. CONVERT TO SECONDS
*******************************************************
** CONVERT ØØ ØØ DD HH MM SS TT TC                   **
**      TO SS SS SS SS SS SS TT TC                   **
*******************************************************
* UNPACK INTO ZONED FIELD
         UNPK  WORKZL15,WORKPL8A
* INIT ACCUMULATOR TO ZERO
         ZAP   WORKPL8A,=P'Ø'
* GET SSTTT AND PACK IT
         PACK  WORKPL8A+5(3),WORKZL15+1Ø(5)
* INIT WORK PACK FIELD
         ZAP   WORKPL8B,=P'Ø'
* GET MM AND PACK IT
         PACK  WORKPL8B+6(2),WORKZL15+8(2)
* MULTIPLY MM BY 6Ø TO CONVERT IN SECONDS
```

```
         MP    WORKPL8B,=P'60000'
* ADD IT TO THE ACCUMULATOR
         AP    WORKPL8A,WORKPL8B
* INIT WORK PACK FIELD
         ZAP   WORKPL8B,=P'0'
* GET HH AND PACK IT
         PACK  WORKPL8B+6(2),WORKZL15+6(2)
* MULTIPLY HH BY 3600 TO CONVERT IN SECONDS
         MP    WORKPL8B,=P'3600000'
* ADD IT TO THE ACCUMULATOR
         AP    WORKPL8A,WORKPL8B
* INIT WORK PACK FIELD
         ZAP   WORKPL8B,=P'0'
* GET DD AND PACK IT
         PACK  WORKPL8B+6(2),WORKZL15+4(2)
* MULTIPLY DD BY 86400 TO CONVERT IN SECONDS
         MP    WORKPL8B,=P'86400000'
* ADD IT TO THE ACCUMULATOR
         AP    WORKPL8A,WORKPL8B
CONV2SEX EQU   *
         L     R14,CONVSAVE
         BR    R14
CONVSAVE DS    F'0'
***************************************** PDATE000 *************
** CONVERT DATE AND TIME FROM STCK TO EXTERNAL FORMAT
PDATE000 DS    0H
         ST    R14,PDATESAV
         L     R1,WORKFWD1          LOAD FWORD TO A REG
         CVD   R1,WORKDWD1          CONVERT DEC TO A DWORD
         DP    WORKDWD1,=P'100'
         DP    WORKDWD1(6),=P'3600'
         ZAP   WORKDWD2,WORKDWD1(3)
         MP    WORKDWD2,=P'10000'
         ZAP   WORKDWD3,WORKDWD1+3(3)
         DP    WORKDWD3,=P'60'
         ZAP   WORKDWD1,WORKDWD3(6)
         MP    WORKDWD1,=P'100'
         AP    WORKDWD2,WORKDWD1
         AP    WORKDWD2,WORKDWD3+6(2)
         ZAP   WORKDWD1,WORKDWD2
         MVC   WORKCL10,=X'402120207A20207A2020'
         ED    WORKCL10,WORKDWD1+4
         MVC   WORKCL07,=X'4021204B202020'
         ED    WORKCL07(7),WORKFWD2+1
PDATE999 DS    0H
         L     R14,PDATESAV
         BR    R14
PDATESAV DS    F'0'
***************************************** INITS000 *************
```

```
** INITIALIZE STUFF
INITSØØØ DS    ØH
         ST    R14,INITSSAV
         OPEN  (SYSIN,INPUT)
         OPEN  (SYSPRINT,OUTPUT)
         OPEN  (SNAPDUMP,OUTPUT)
         OPEN  (SMFDATA,INPUT)
         OPEN  (SMFSOUT,OUTPUT)
         OPEN  (SMFDOUT,OUTPUT)
         ZAP   COUNTER1,=P'Ø'
         ZAP   COUNTER2,=P'Ø'
         ZAP   COUNTER3,=P'Ø'
INITS999 DS    ØH
         L     R14,INITSSAV
         BR    R14
INITSSAV DS    F'Ø'
****************************************** COLHDØØØ *************
** PRINT COLUMN HEADERS
COLHDØØØ DS    ØH
         ST    R14,COLHDSAV
         MVC   O3PLAN,=C'PLANNAME'
         MVC   O3DATE,=C'DATE  '
         MVC   O3TIME,=C'TIME    '
         MVC   O3TYPE,=C'TP'
         MVC   O3RECN,=C'NN'
         MVC   O3PKID,=C'PACKAGEID/DBRMNAME'
         MVC   O3ACSCT,=C'     ELAPSED'
         MVC   O3ACTJST,=C'     TCB-CPU'
         MVC   O3ACAWTI,=C'    I/O-WAIT'
         MVC   O3ACAWTL,=C'    L/L-WAIT'
         MVC   O3ACAWTR,=C'  ASYNC-READ'
         MVC   O3ACAWTE,=C'  SYNCH-READ'
         MVC   PRNTLINE,O3RECORD
         PUT   SMFSOUT,PRNTLINE
         MVC   O3PLAN,=C'--------'
         MVC   O3DATE,=C'------'
         MVC   O3TIME,=C'--------'
         MVC   O3TYPE,=C'--'
         MVC   O3RECN,=C'--'
         MVC   O3PKID,=C'------------------'
         MVC   O3ACSCT,=C'------------'
         MVC   O3ACTJST,=C'------------'
         MVC   O3ACAWTI,=C'------------'
         MVC   O3ACAWTL,=C'------------'
         MVC   O3ACAWTR,=C'------------'
         MVC   O3ACAWTE,=C'------------'
         MVC   PRNTLINE,O3RECORD
         PUT   SMFSOUT,PRNTLINE
         MVC   O3RECORD,=132CL1' '
```

```
         MVC   PRNTLINE,O3RECORD
         PUT   SMFSOUT,PRNTLINE
COLHD999 DS    ØH
         L     R14,COLHDSAV
         BR    R14
COLHDSAV DS    F'Ø'
*************************************** FINALØØØ *************
** FINALIZE   STUFF
FINALØØØ DS    ØH
         ST    R14,FINALSAV
         MVC   PRNTLINE,=132CL1' '
         PUT   SYSPRINT,PRNTLINE
         MVC   PRNTLINE(25),=C' TOTAL SMF RECORDS READ :'
         MVC   COUNTERØ,=X'2Ø2Ø2Ø2Ø2Ø2Ø2Ø2Ø21'
         ED    COUNTERØ,COUNTER1
         MVC   PRNTLINE+25(9),COUNTERØ
         PUT   SYSPRINT,PRNTLINE
         MVC   PRNTLINE(25),=C' TOTAL ACCOUNTNG RECORDS:'
         MVC   COUNTERØ,=X'2Ø2Ø2Ø2Ø2Ø2Ø2Ø2Ø21'
         ED    COUNTERØ,COUNTER2
         MVC   PRNTLINE+25(9),COUNTERØ
         PUT   SYSPRINT,PRNTLINE
         MVC   PRNTLINE(25),=C' TOTAL RECORDS SELECTED :'
         MVC   COUNTERØ,=X'2Ø2Ø2Ø2Ø2Ø2Ø2Ø2Ø21'
         ED    COUNTERØ,COUNTER3
         MVC   PRNTLINE+25(9),COUNTERØ
         PUT   SYSPRINT,PRNTLINE
         CLOSE SYSIN
         CLOSE SYSPRINT
         CLOSE SNAPDUMP
         CLOSE SMFDATA
         CLOSE SMFSOUT
         CLOSE SMFDOUT
FINAL999 DS    ØH
         L     R14,FINALSAV
         BR    R14
FINALSAV DS    F'Ø'
*************************************** GETSYØØØ ************
** GETSYØØØ - READ SYSIN FOR PASSED PARMS
** SYNTAX RULES:
** 1. VALID KEYWORDS ARE: SSID=,PLAN=,PAKG=,DATE=
** 2. KEYWORDS SHOULD START IN COLUMN 1
** 3. DO NOT USE COMMA TO TERMINATE A KEY VALUE
** 4. PLAN AND PAKG CAN ACCEPT PREFIXED INPUT IE XXXX%
** 5. DATE FORMAT IS YYYYMMDD
GETSYØØØ DS    ØH
         ST    R14,GETSYSAV
         MVC   KWRDSSID,=C'SSID='
         MVC   CRITSSID,=C'    '
```

```
         MVC   KWRDPLAN,=C'PLAN='
         MVC   CRITPLAN,=C'        '
         MVC   KWRDPAKG,=C'PAKG='
         MVC   CRITPAKG,=C'                '
         MVC   KWRDDATE,=C'DATE='
         MVC   CRITDATE,=C'        '
GETSY01Ø BAL   R14,READN000         READ SYSIN CARD
         CLI   READNFLG,X'FF'       IF EOF
         BE    GETSY900             Y. EXIT
         CLC   I1KEYWRD,KWRDSSID    IF SSID KEYWORD
         BE    GETSY02Ø
         CLC   I1KEYWRD,KWRDPLAN    IF PLAN KEYWORD
         BE    GETSY03Ø
         CLC   I1KEYWRD,KWRDPAKG    IF PAKG KEYWORD
         BE    GETSY04Ø
         CLC   I1KEYWRD,KWRDDATE    IF DATE KEYWORD
         BE    GETSY05Ø
         B     GETSY01Ø
GETSY02Ø MVC   CRITSSID,I1PARM
         B     GETSY01Ø
GETSY03Ø MVC   CRITPLAN,I1PARM
         B     GETSY01Ø
GETSY04Ø MVC   CRITPAKG,I1PARM
         B     GETSY01Ø
GETSY05Ø MVC   CRITDATE,I1PARM
         B     GETSY01Ø
GETSY900 MVC   PRNTLINE(25),=C' SELECTION CRITERIA USED:'
         PUT   SYSPRINT,PRNTLINE
         MVC   PRNTLINE,=132CL1' '
GETSY91Ø CLC   CRITSSID,=C'    '
         BE    GETSY92Ø
         MVC   PRNTLINE+1(5),KWRDSSID
         MVC   PRNTLINE+6(4),CRITSSID
         PUT   SYSPRINT,PRNTLINE
GETSY92Ø CLC   CRITPLAN,=C'        '
         BE    GETSY93Ø
         MVC   PRNTLINE+1(5),KWRDPLAN
         MVC   PRNTLINE+6(8),CRITPLAN
         PUT   SYSPRINT,PRNTLINE
GETSY93Ø CLC   CRITPAKG(8),=C'        '
         BE    GETSY94Ø
         MVC   PRNTLINE+1(5),KWRDPAKG
         MVC   PRNTLINE+6(8),CRITPAKG
         PUT   SYSPRINT,PRNTLINE
GETSY94Ø CLC   CRITDATE,=C'        '
         BE    GETSY999
         MVC   PRNTLINE+1(5),KWRDDATE
         MVC   PRNTLINE+6(8),CRITDATE
         PUT   SYSPRINT,PRNTLINE
```

```
GETSY999 DS    ØH
         MVC   PRNTLINE,=132CL1' '
         PUT   SYSPRINT,PRNTLINE           PRINT BLANK LINE
         L     R14,GETSYSAV
         BR    R14
GETSYSAV DS    F'Ø'
***************************************** READNØØØ *************
** READ SYSIN CARDS
READNØØØ DS    ØH
         ST    R14,READNSAV
         GET   SYSIN,I1RECORD
READN999 DS    ØH
         L     R14,READNSAV
         BR    R14
READNEOF DS    ØH
         MVI   READNFLG,X'FF'
         B     READN999
READNSAV DS    F'Ø'
READNFLG DS    X'Ø'
***************************************** READSØØØ *************
** READ SMF RECORDS
** R3 POINTS TO THE BUFFER OF THE RECENTLY READ RECORD
READSØØØ DS    ØH
         ST    R14,READSSAV
         GET   SMFDATA
         LR    R3,R1             SAVE BUFFER AFTER READ
         USING SM1Ø1,R3          ANCHOR R3 TO SMF1Ø1 RECORD LAYOUT
         AP    COUNTER1,=P'1'    COUNT RECORDS READ
**       PUT   SMFDUMP,Ø(R3)
READS999 EQU   *
         L     R14,READSSAV
         BR    R14
READSEOF EQU   *
         MVI   READSFLG,X'FF'
         B     READS999
READSSAV DS    F'Ø'
READSFLG DS    X'Ø'
***************************************** CHECKØØØ ************
** CHECK RECORD TYPE IS 1Ø1
**   AND SSID
CHECKØØØ DS    ØH
         ST    R14,CHECKSAV
         MVI   CHECKFLG,X'ØØ'     ASSUME RECORD IS GOOD
         SR    R15,R15            CLEAR REG
         ICM   R15,1,SM1Ø1RTY     INSERT RECORD TYPE
         CH    R15,=H'1Ø1'        IF TYPE 1Ø1
         BNE   CHECKØ1Ø           N. SKIP RECORD
         CLC   SM1Ø1SSI,CRITSSID  IF SUBSYSTEM IS CORRECT
         BNE   CHECKØ1Ø           N. SKIP RECORD
```

```
         AP    COUNTER2,=P'1'     COUNT RECORDS PROCESSED
         B     CHECK999
CHECKØ1Ø MVI   CHECKFLG,X'FF'     MARK FOR SKIPPING
CHECK999 DS    ØH
         L     R14,CHECKSAV
         BR    R14
CHECKSAV DS    F'Ø'
CHECKFLG DS    X'Ø'
**       W O R K I N G   S T O R A G E
**       D C B
SYSIN    DCB   DSORG=PS,MACRF=(GM),                                    X
               DDNAME=SYSIN,EODAD=READNEOF
SMFDATA  DCB   DSORG=PS,MACRF=GL,                                      X
               DDNAME=SMFDATA,EODAD=READSEOF,BFTEK=A
SMFSOUT  DCB   DSORG=PS,RECFM=F,MACRF=(PM),LRECL=132,BLKSIZE=3Ø36,     X
               DDNAME=SMFSOUT
SMFDOUT  DCB   DSORG=PS,RECFM=FB,MACRF=(PM),LRECL=8Ø,BLKSIZE=2344Ø,    X
               DDNAME=SMFDOUT
SYSPRINT DCB   DSORG=PS,RECFM=F,MACRF=(PM),LRECL=132,BLKSIZE=3Ø36,     X
               DDNAME=SYSPRINT
SNAPDUMP DCB   DSORG=PS,RECFM=VBA,MACRF=(W),LRECL=125,BLKSIZE=1632,    X
               DDNAME=SNAPDUMP
SNAPAREA DS    ØH
         DS    18F'Ø'
         DS    2F
SNAPAEND DS    ØH
***************************************** WORKAREA DSECT ************
WORKAREA DSECT
SAVEAREA DC    18F'Ø'
WHPKRECN DS    H
WHIFCID  DS    H
         DS    ØD
WORKCL16 DS    CL16
WORKZL15 DS    ZL15
WORKZLØ1 DS    ZLØ1
WORKPL8A DS    PLØ8
WORKPL8B DS    PLØ8
WORKPL8C DS    PLØ8
CRITERIA DS    ØCL61
KWRDSSID DS    CLØ5
CRITSSID DS    CLØ4
         DS    CLØ1
KWRDPLAN DS    CLØ5
CRITPLAN DS    CLØ8
         DS    CLØ1
KWRDPAKG DS    CLØ5
CRITPAKG DS    CL18
         DS    CLØ1
KWRDDATE DS    CLØ5
```

```
CRITDATE DS    CL08
COUNTER0 DS    CL20        TEXT
COUNTER1 DS    PL05        RECORDS READ
COUNTER2 DS    PL05        RECORDS PROCESSED
COUNTER3 DS    PL05        RECORDS SELECTED
         DS    0F
WORKFWD1 DS    F
WORKFWD2 DS    F
WORKDWD1 DS    D
WORKDWD2 DS    D
WORKDWD3 DS    D
WORKCL03 DS    CL03
WORKCL07 DS    CL07
WORKCL10 DS    CL10
         DS    0H
PRNTLINE DS    0CL132              SYSPRINT
         DS    CL132
PRNTLENG EQU   *-PRNTLINE          LENGTH OF PRINTLINE
I1RECORD DS    0CL80               SYSIN
I1KEYWRD DS    CL05
I1PARM   DS    CL18
         DS    CL57
I1LENGTH EQU   *-I1RECORD          LENGTH OF SYSIN
         DS    0H
O1RECORD DS    0CL132              OUTREC 1
O1TYPE   DS    CL04
         DS    CL01
O1PLAN   DS    CL08
         DS    CL01
O1PKGN   DS    CL03
         DS    CL80
         DS    CL15
O1IFCID  DS    CL03
         DS    CL01
O1DATE   DS    CL06
         DS    CL01
O1TIME   DS    CL08
         DS    CL01
O1LENGTH EQU   *-O1RECORD          LENGTH OF OUTREC 1
         DS    0H
O2RECORD DS    0CL132              OUTREC2
         DS    CL05
O2TYPE   DS    CL02
O2RECN   DS    CL02
         DS    CL01
O2PKID   DS    CL18
         DS    CL70
O2LOCN   DS    CL16
O2COLN   DS    CL18
```

```
O2LENGTH EQU   *-O2RECORD           LENGTH OF OUTREC 2
         DS    ØH
O3RECORD DS    ØCL132              OUTREC3
O3PLAN   DS    CLØ8
         DS    CLØ1
O3DATE   DS    CLØ6
         DS    CLØ1
O3TIME   DS    CLØ8
         DS    CLØ1
O3TYPE   DS    CLØ2
O3RECN   DS    CLØ2
         DS    CLØ1
O3PKID   DS    CL18
         DS    CLØ1
O3ACSCT  DS    CL12                TOT ELAPSED TIME
         DS    CLØ1
O3ACTJST DS    CL12                TOT TCB CPU TIME
         DS    CLØ1
O3ACAWTI DS    CL12                TOT I/O WAIT TIME
         DS    CLØ1
O3ACAWTL DS    CL12                TOT LOCK/LATCH WAIT TIME
         DS    CLØ1
O3ACAWTR DS    CL12                TOT ASYNC READ
         DS    CLØ1
O3ACAWTE DS    CL12                TOT SYNCH READ
         DS    CL2Ø
O3LENGTH EQU   *-O3RECORD          LENGTH OF OUTREC 3
O4RECORD DS    ØCL8Ø               OUTREC4
O4PLAN   DS    CLØ8
O4PKID   DS    CL18
O4TYPE   DS    CLØ2
O4ACSCT  DS    PLØ8                TOT ELAPSED TIME
O4ACTJST DS    PLØ8                TOT TCB CPU TIME
O4ACAWTI DS    PLØ8                TOT I/O WAIT TIME
O4ACAWTL DS    PLØ8                TOT LOCK/LATCH WAIT TIME
O4ACAWTR DS    PLØ8                TOT ASYNC READ
O4ACAWTE DS    PLØ8                TOT SYNCH READ
         DS    CLØ4
O4LENGTH EQU   *-O4RECORD          LENGTH OF OUTREC 4
WORKLEN  EQU   *-WORKAREA
****************************************** SM1Ø1 DSECT ***************
**  THIS DSECT DEFINES THE RECORD LAYOUT OF SMF TYPE 1Ø1 RECORD
**  WHICH IS THE DB2 ACCOUNTING TRACE RECORD
SMFRECRD DSNDQWAS DSECT=YES,SUBTYPE=ALL
****************************************** REGISTER EQUATES **********
RØ       EQU   Ø
R1       EQU   1
R2       EQU   2
R3       EQU   3
```

```
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         END
//LKED    EXEC PGM=IEWL,PARM='XREF',
//             COND=((4,LT,ASM))
//SYSLIB   DD  DISP=SHR,DSN=SSID.DB2.DSNLOAD
//SYSLIN   DD  DSN=&&LOADSET,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DSN=MYTSOID.LOAD(SMF1Ø1P1),DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  SPACE=(1Ø24,(5Ø,5Ø)),UNIT=SYSDA
//SYSIN   DD *
  NAME SMF1Ø1P1(R)
//*
//RUN     EXEC PGM=SMF1Ø1P1,
//             COND=((4,LT,ASM),(4,LT,LKED))
//STEPLIB  DD  DISP=SHR,DSN=MYTSOID.LOAD
//SNAPDUMP DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SMFDATA  DD  DISP=SHR,DSN=MYTSOID.SMFDATA
//SMFSOUT  DD  SYSOUT=*
//SMFDOUT  DD  DISP=SHR,DSN=MYTSOID.SMFSUMM
//SYSIN    DD  *
SSID=SSID
//
```

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

Larry Prestosa (USA)

Rebind and convert plans and packages

The PREB procedure enables you to rebind plans and packages or convert plans to packages.

Figure 1 shows main menu of this procedure.

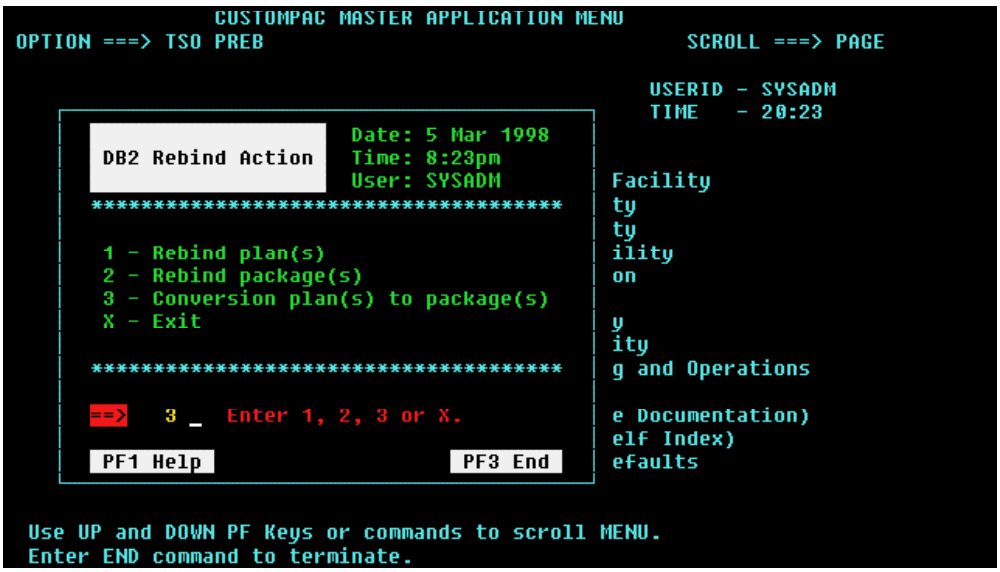


Figure 1: Main menu

BUILDING REBIND SUBCOMMANDS

The DSN subcommand REBIND PLAN/PACKAGE rebinds an application plan/package when you make changes that affect the plan/package (create a new index or RUNSTATS), but do not change the SQL statements in the programs.

My procedure uses SELECT to select specific plans or packages to be rebound. If the SELECT statement returns rows, then the IBM install program DSNTIAUL generates REBIND subcommands for the plans or packages identified in the returned rows.

Put those subcommands in a sequential dataset, where the next step (the EDIT step in the PLANREB skeleton for the JCL) can then edit

them. If the SELECT statement returns no qualifying rows, then DSNTIAUL does not generate REBIND subcommands.

The Rebind Plans subprocedure enables:

- Rebind plan – one or more plans.
- Rebind all plans.
- Rebind all plans bound before a given date and time.
- Rebind all plans bound since a given date and time.
- Rebind all plans bound within a given date and time range.
- Rebind all invalid plans.
- Rebind all inoperative plans.
- Rebind all plans bound with isolation level of cursor stability.

The Rebind Packages subprocedure enables:

- Rebind package – one or more packages.
- Rebind all versions of all packages.
- Rebind all versions of all packages bound before a given date and time.
- Rebind all versions of all packages bound since a given date and time.
- Rebind all versions of all packages bound within a given date and time range.
- Rebind all invalid versions of the packages.
- Rebind all inoperative versions of the packages.
- Rebind all versions of all packages that allow CPU and/or I/O parallelism.

CONVERTING A PLAN TO A PACKAGE

This subprocedure generates a batch job stream that will convert your plans to packages.

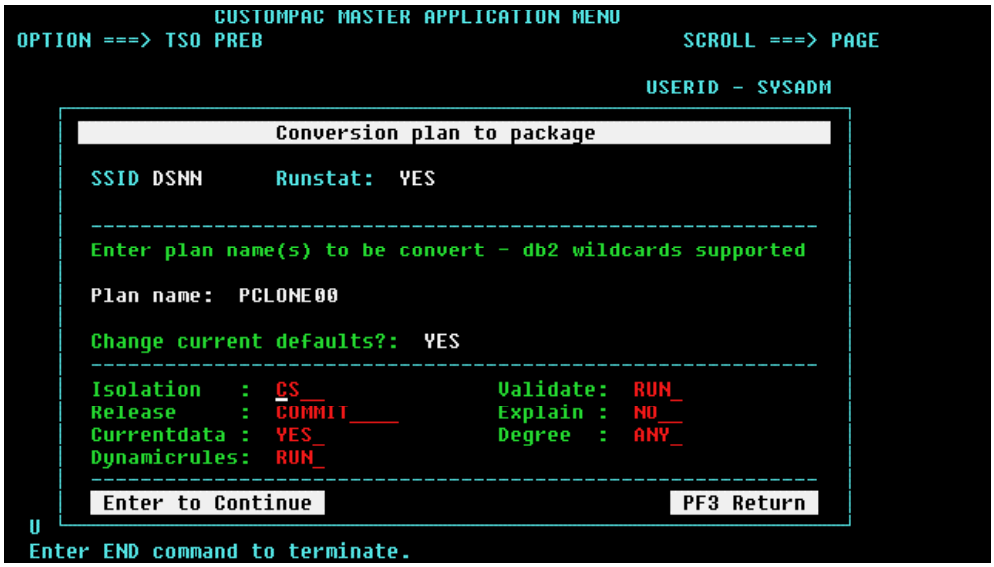


Figure 2: Converting plan to package

Figure 2 shows the panel for conversion plan/package process.

Notes:

- The owner of the plan will be the owner of the package.
- There is one collection per plan.
- The names of plans and collections are the same.
- The name of package and plan is the same.
- The job stream has two steps: Bind of Package and Bind of Plan (include PKLIST).

Advantages of packages over plans:

- Binding is quick, only one DBRM.
- Packages can be bound when the plan is in use.
- Plans do not need rebinding when adding a new package.
- A package can exist in multiple versions.

- A package can be bound to multiple collections with different options.

THE COMPONENTS OF PREB

The following are the components of PREB:

SYSPROC:

- PREB – driver procedure
- PLREB – rebind plans procedure
- PAREB – rebind packages procedure
- PLPA – conversion plan to package procedure
- RSQL – on-line SQL procedure.

ISPPLIB:

- REBP00 – main menu
- REBP00H – help
- REBP01 – rebind plans panel
- REBP02 – rebind packages panel
- REBP01H – plan/package help panel
- REBP03 – conversion plan to package panel part 1
- REBP04 – conversion plan to package panel part 2
- RSQLPAN – on-line SQL output panel.

ISPMLIB:

- PREB00 – preb message.

ISPPLIB:

- PREBPL – PLI source code for rebind plans
- PREBPA – PLI source code for rebind packages
- PREBCO – PLI source code for conversion plans to packages

- PREBRU – PLI source code for runstat.

ISPSLIB:

- PLANREB – JCL skeleton for rebind plans/packages
- CONVERSE – JCL skeleton for conversion plans to packages.

PREB – DRIVER PROCEDURE

```
/* REXX *//* TRACE R */ ZPFCTL = 'OFF' ADDRESS ISPEXEC 'VPUT (ZPFCTL)
PROFILE' ADDRESS ISPEXEC 'ADDPOP ROW(3) COLUMN(1)'
 TOP:
 date=DATE()
 time=TIME(C)
 ADDRESS ISPEXEC "DISPLAY PANEL(REBPØØ)"
 DO WHILE RC=Ø
    action=Ø
    SELECT
      WHEN(X='1') THEN DO
         ADDRESS ISPEXEC REMPOP ALL
         "%PLREB"
         action = rc
         ADDRESS ISPEXEC REMPOP ALL
         ADDRESS ISPEXEC 'ADDPOP ROW(3) COLUMN(1)'
      END
      WHEN(X='2') THEN DO
         ADDRESS ISPEXEC REMPOP ALL
         "%PAREB"
         action = rc
         ADDRESS ISPEXEC REMPOP ALL
         ADDRESS ISPEXEC 'ADDPOP ROW(3) COLUMN(1)'
      END
      WHEN(X='3') THEN DO
         ADDRESS ISPEXEC REMPOP ALL
         "%PPLPA"
         action = rc
         ADDRESS ISPEXEC REMPOP ALL
         ADDRESS ISPEXEC 'ADDPOP ROW(3) COLUMN(1)'
      END
      WHEN(X='X') THEN DO
         ADDRESS ISPEXEC REMPOP ALL
         EXIT
      END
      OTHERWISE RC=Ø
    END
    date=DATE()
    time=TIME(C)
    IF action=Ø THEN ADDRESS ISPEXEC "DISPLAY PANEL(REBPØØ)"
```

```
 END
 EXIT
```

PLREB – REBIND PLANS PROCEDURE

```
/* REXX */
/* trace r */
X=MSG("OFF")
ZPFCTL = 'OFF'
ADDRESS ISPEXEC 'VPUT (ZPFCTL) PROFILE'
ADDRESS ISPEXEC 'VGET (db2) PROFILE'
ADDRESS ISPEXEC 'ADDPOP ROW(3) COLUMN(1)'
rst='NO'
ans='NO'
CUR='ff'
top:
ADDRESS ISPEXEC "DISPLAY PANEL(REBPØ1) CURSOR("CUR")"
Call dsn
option='REBIND PLAN'
DO WHILE RC=Ø
   text=''
   line1="Select substr('REBIND PLAN('concat name"
   line2="concat')                         ',1,45)"
   line3="from sysibm.sysplan                     "
   line4=''
   line5=''
   SELECT
     WHEN(ff='1') THEN DO
      CUR='ppla'
      title="Rebind plan(s)"
      if ppla = '' then do
         message="Enter plan name - db2 wildcards supported"
         ADDRESS ISPEXEC "SETMSG MSG(PREBØØ1)"
         SIGNAL top
      end
      vname='%'
      if ppla ¬= '' then vname=ppla!!'%'
      if length(ppla) = 8 then vname=ppla
      line4="where name like '"vname"'"!!';'
      Call Generate_jcl
     END
     WHEN(ff='1?') THEN DO
      head ="1-Rebind plan(s)"
      text ="Enter plan name - db2 wildcards supported"
      vname='%'
      if ppla ¬= '' then vname=ppla!!'%'
      if length(ppla) = 8 then vname=ppla
      line4="where name like '"vname"'"
      CUR='ppla'
```

```
 Call Help
 ff=1
END
WHEN(ff='2') THEN DO
 title="Rebind all plans"
 line3=line3!!';'
 CUR='ff'
 Call Generate_jcl
END
WHEN(ff='2?') THEN DO
 head ="2-Rebind all plans"
 CUR='ff'
 Call Help
 ff=2
END
WHEN(ff='3') THEN DO
 Call Numeric 1
 if ind = '1' then SIGNAL top
 title="Rebind all plans before a given date and time"
 Call fields
 line4="where binddate <= '"vdate1"'"
 line5="and   bindtime <= '"vtime1"'"!!';'
 CUR='ff'
 Call Generate_jcl
END
WHEN(ff='3?') THEN DO
 Call Numeric 1
 head ="3-Rebind all plans before a given date and time"
 Call fields
 line4="where binddate <= '"vdate1"'"
 line5="and   bindtime <= '"vtime1"'"
 CUR='ff'
 Call Help
 ff=3
END
WHEN(ff='4') THEN DO
 Call Numeric 1
 if ind = '1' then SIGNAL top
 Call fields
 title="Rebind all plans since a given date and time"
 line4="where binddate >= '"vdate1"'"
 line5="and   bindtime >= '"vtime1"'"!!';'
 CUR='ff'
 Call Generate_jcl
END
```

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

Bernard Zver
Database Administrator
Informatika Maribor (Slovenia)

Cisco has announced its Enterprise 2000 initiative for IBM networks, including new products and a partners programme aimed at providing blueprints for integrating varied network requirements.

The idea is that products are used to develop a single network infrastructure allowing secure access to IBM mainframe data from any client including SNA emulators, TN3270 emulators, and Web browsers.

For DB2 sites, there's DataBase Connection, a new IOS feature that enables access to DB2 databases from end users at TCP/IP workstations which has been jointly developed by Cisco and StartQuest.

For further information contact:
Cisco, 5305 Gulf Drive, Suite 1, New Port Richey, FL 34652, USA.
Tel: (813) 817 0131.

* * *

Platinum Technology has announced the OnlineReorg and LogCompress utilities for DB2 management, along with the Index Expert index design tool, all for OS/390-based systems.

OnlineReorg, as its name suggests, is for on-line reorganization of fragmented databases, while LogCompress is for reducing the size of archive log files by up to 97% using compression technology.

Index Expert, meanwhile, is claimed to be the first index design tool available for mainframe-based DB2 systems. It's designed to significantly speed response time for queries to DB2-based applications by automating the design of efficient indexes, and so providing direct access paths to the databases supplying information to applications.

Specifically, the product performs automated analysis to identify indexes that should be created to improve application performance. And the information it gathers about databases allows DBAs to generate 15 pre-defined reports, such as impact analysis and column usage reports, perform automated year 2000 compliance analysis, and identify columns and tables that are never referenced.

It can also accept database statistics from other Platinum DB2 tools.

For further information contact:
Platinum Technology, 1815 S Meyer Rd, Oakbrook Terrace, IL 60181-5241, USA.
Tel: (630) 620 5000.
Platinum Technology, Platinum House, North Second Street, Milton Keynes, MK9 1BZ, UK.
Tel: (01908) 248400.

* * *

IBM has begun shipping its DB2 Universal Database Enterprise Extended Edition on NT and Solaris. Features include parallel operation of all utilities function, including data and index scan, index creation, and back-up and restore. Other features include parallel database operations over all available processors of one or more servers, plus centralized administration of nodes.

Also, IBM has announced that it intends to provide Enterprise JavaBeans support, over time, for DB2, CICS/390, and IMS host-based transaction software, plus MQSeries and the Domino Web application server platform. VisualAge tools will also embrace Enterprise JavaBeans in the future.

For further information contact your local IBM representative.