June 1998

In this issue

3  Finding DB2 information using the Internet
9  Generation of unload and load utilities
18  Accessing DB2 data over the Internet
22  Collecting accounting information
41  Rebind and convert plans and packages
48  DB2 news

© Xephon plc 1998
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.

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

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

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

Printed in England.
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.

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

*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-l 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.
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 ====>** Specifying subsystem id & database

**SSID ======>** DB2T **DBNAME ====>** STATD200

**OUTPUT LIBRARY ======>** h1vl.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,*
REGION=8M,TIME=1439
//*
///
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                              */
/*                                                                     */
/* ****************************************************************** */

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
endif

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 "// UNIT=SYSDA,"
queue "// DCB=(LRECL=80,BLKSIZE=8000,RECFM=FB),"
queue "// SPACE=(CYL,(1,1),RLSE)"
queue "//"
queue "="/**************************************************
queue "//* Terminate utility (for rerunability)"
queue "="/**************************************************
queue "/STEP3 EXEC PGM=IKJEFT01,DYNAMNBR=20"
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,(50,50)),UNIT=SYSDA"
queue "/SYSUT1 DD SPACE=(CYL,(50,50)),UNIT=SYSDA"
queue "/SYSMAP DD SPACE=(CYL,(50,50)),UNIT=SYSDA"
queue "count = 0"
do while count < sqlca.sqlrows
  count = count + 1
  chrcnt = count - 1
  if chrcnt < 10 then chrcnt = "0"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=IKJEFT01,DYNAMNBR=20"
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)"

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
    THEN
        ZEDSMG = "LMINIT "33rc
        address "ISPEXEC" "SETMSG MSG(ISRZØØ1)"
        EXIT RC
    END

"EDIT DATAID("ID1") MEMBER("jmem")"

IF RC > 4 THEN
    THEN
        ZEDSMG = "EDIT "33rc
        address "ISPEXEC" "SETMSG MSG(ISRZØØ1)"
        EXIT RC
    END

address "ISPEXEC" "VPUT (db2s dbunld) PROFILE"
address "ISPEXEC" "VPUT (jdsn jmem) PROFILE"
PDB2UNLD ISP F PANEL

%-------------------------- DB2 UNLOAD/RELOAD FUNCTION --------------------------
-------

%COMMAN D ===>_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.1,20,30)
         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

Tom Sager (USA) © Xephon 1998
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) © Craig S Mullins 1998
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 information 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=ASMA9Ø,PARM='OBJECT,NODECK'
//SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
// DD DSN=SSID.DB2.DSNMACS,DISP=SHR
// DD DSN=SSID.DB2.DSNAMP,DISP=SHR
//SYSLIN DD DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,
// SPACE=(800,1000,1000)),DCB=(BLKSIZE=800)
//SYSPRINT DD SYSOUT=*
### Table: Sample SMF101P1 Report

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

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

**Note:** actual report is 132 characters in width.
### DB2 Accounting SMF Records - Package/DBRM

<table>
<thead>
<tr>
<th>PLANNAME</th>
<th>PACKAGEID/DBRMNAME</th>
<th>TYPE</th>
<th>COUNT</th>
<th>ELAPSED</th>
<th>TCB-CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMNPLAN</td>
<td>CMNDB2SQ</td>
<td>PK</td>
<td>33</td>
<td>1.393</td>
<td>0.558</td>
</tr>
<tr>
<td>DSNESPAC</td>
<td>DSNEM68</td>
<td>PK</td>
<td>2</td>
<td>1.573</td>
<td>0.119</td>
</tr>
<tr>
<td>DSNESPPR</td>
<td>DSNEM68</td>
<td>PK</td>
<td>19</td>
<td>9.399</td>
<td>3.594</td>
</tr>
<tr>
<td>DSNTEP2</td>
<td>DSNTEP2</td>
<td>RM</td>
<td>41</td>
<td>4.957</td>
<td>0.976</td>
</tr>
<tr>
<td>DSNTIAL</td>
<td>DSNTIAL</td>
<td>PK</td>
<td>21</td>
<td>11.200</td>
<td>5.368</td>
</tr>
<tr>
<td>F2PLN371</td>
<td>ADSQID000</td>
<td>PK</td>
<td>1</td>
<td>0.667</td>
<td>0.018</td>
</tr>
<tr>
<td>F2PLN371</td>
<td>CWSQLPRO</td>
<td>PK</td>
<td>12</td>
<td>7.582</td>
<td>1.045</td>
</tr>
<tr>
<td>F2PLN371</td>
<td>OSSQLCAT</td>
<td>PK</td>
<td>13</td>
<td>0.034</td>
<td>0.007</td>
</tr>
<tr>
<td>F2PLN371</td>
<td>OSSQLC41</td>
<td>PK</td>
<td>10</td>
<td>0.162</td>
<td>0.021</td>
</tr>
</tbody>
</table>

**Run Date:** 98.028  **Time:** 15:40:33

<table>
<thead>
<tr>
<th>I/O-WAIT</th>
<th>L/L-WAIT</th>
<th>ASYNC-READ</th>
<th>SYNCH-READ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.522</td>
<td>0.001</td>
<td>0.046</td>
<td>0.072</td>
</tr>
<tr>
<td>0.112</td>
<td>0.003</td>
<td>1.324</td>
<td>0.001</td>
</tr>
<tr>
<td>0.753</td>
<td>0.005</td>
<td>2.886</td>
<td>0.001</td>
</tr>
<tr>
<td>0.221</td>
<td>0.008</td>
<td>2.125</td>
<td>0.134</td>
</tr>
<tr>
<td>0.910</td>
<td>0.001</td>
<td>1.711</td>
<td>0.100</td>
</tr>
<tr>
<td>0.104</td>
<td>0.000</td>
<td>0.020</td>
<td>0.044</td>
</tr>
<tr>
<td>1.071</td>
<td>0.018</td>
<td>4.597</td>
<td>0.578</td>
</tr>
<tr>
<td>0.009</td>
<td>0.000</td>
<td>0.000</td>
<td>0.004</td>
</tr>
<tr>
<td>0.109</td>
<td>0.000</td>
<td>0.009</td>
<td>0.006</td>
</tr>
</tbody>
</table>
//SYSUDUMP DD SYSOUT=*  
//SYSUT1 DD SPACE=(800,(1000,100),..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:  
* QWAØ - 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, 2Ø48(R11) SET UP SECOND BASE REGISTER 
** LA R12, 2Ø48(R12) SET UP SECOND BASE REGISTER 
** B INITIALZ GO AROUND SAVEAREA 

***************************************** INITIALIZATION *****
** I N I T I A L I Z E 
** INITIALZ DS ØH 
** BAL R14, INITSØØØ INITIALIZE STUFF 
** BAL R14, GETSYØØØ GET PASSED PARM 
** BAL R14, COLHDØØØ PRINT COLUMN HEADERS 

***************************************** MAINLINE ********** 
** M A I N L I N E 
** MAINLINE DS ØH 
** BAL R14, READSØØØ READ SMF RECORD 
** CLI READSFLG, X'FF' IF END OF FILE? 
** BE FINALIZE THEN FINALIZE 
** BAL R14, CHECKØØØ CHECK RECTYPE AND SSID 
** CLI CHECKFLG, X'FF' IF RECORD NOT OK 
** BE MAINLINE THEN READ NEXT RECORD 
** BAL R14, PROCEØØØ PROCESS THIS RECORD 
** B MAINLINE READ NEXT RECORD 

***************************************** FINALIZATION ********** 
** F I N A L I Z E 
** FINALIZE DS ØH 
** BAL R14, FINALØØØ 
** 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=Ø RETURN TO OS WITH RETCODE=Ø 

***************************************** SUBROUTINES ********** 
** S U B - R O U T I N E S 
** INITSØØØ ********** 
** PROCESS THE ACCOUNTING RECORD 
** NAVIGATE TO VARIOUS SECTIONS 
** IF THIS IS AN OVERFLOW RECORD (IFCID=239) 
** THEN PROCESS AS SUCH 
** PROCEØØØ DS ØH 
** 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,QWAØ1PSØ/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 QWACPKGØ,=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,QWAØ1R8Ø/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,QPKGØ

PQPAC239 DS ØH/QPAC PACKAGE/DBRM LEVEL ACCTG(OVERFLOW)
    LR R5,R3/INIT R5 TO START OF RECORD
    A R5,QWA11R20/ADD THE OFFSET
    USING QPAC.R5/ASSIGN QPAC TO R5
    BAL R14,FQPACØØØ/FORMAT PACKAGE/DBRM DATA

    B PROCE999

PROCE999 DS ØH/LR R14,PROCESAV/BR R14

PROCESAV DS F'Ø'

** GET SMF HEADER DATA FROM QWHS SECTION  
** DATE, TIME AND IFCIDE  
FQWHS000 DS 0H  
  ST R14,FQWHSSAV  
  USING QWHS,R5  
  MVC WORKFWD1,SM101TIME  
  MVC WORKFWD2,SM101DTE  
  BAL R14,PDATE000  
  MVC 03DATE,WORKCL07+1  
  MVC 03TIME,WORKCL10+2  
  MVC 01DATE,WORKCL07+1  
  MVC 01TIME,WORKCL10+2  
  MVC WHIFICID,QWHSIID  
FQWHS999 DS 0H  
  L R14,FQWHSSAV  
  BR R14  
FQWHSSAV DS F'0'  

** GET CORRELATION HEADER DATA FROM QWHC SECTION  
** 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  
FQWHC005 DS 0H  
  MVC 01PLAN,QWHPLAN  
  MVC 03PLAN,QWHPLAN  
  CLC 03PLAN,=C' ' IF PLAN BLANK?  
  BNE FQWHC010 N. GO ON  
  MVC 03PLAN,=C'*COMMAND' Y. REPLACE WITH THIS TEXT  
  MVC 04PLAN,03PLAN  
  CLC 03PLAN,=C'DSNUTIL ' IF PLAN DSNUTIL  
  B MAINLINE Y. SKIP IT  
  CLC 03PLAN,=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,03PLAN 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  

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
** 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 @H
ST R14,FQWACSAV
USING QWAC,R5
SR R9,R9
MVC WHPKRECN,QWACPKGN

FQWAC999 DS @H
L R14,FQWACSAV
BR R14

** 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 @H
ST R14,FQPACSAV
USING QPAC,R5
LA R8,10
LA FQPAC011
TM QPACFLGS+1,B'ØØØØØØØ1' IF THIS IS A DBRM
BO FQPAC011
TM QPACFLGS+1,B'ØØØØØØ1Ø' IF THIS IS A PAKG
BO FQPAC012
MVC O3TYPE,=C'RM'
B FQPAC020
MVC O3TYPE,=C'PK'
B FQPAC020

FQPAC011 MVC O3PKID,QPACPKID
MARK AS DBRM
B FQPAC020
FQPAC012 MVC O3LOCN,QPACLOCN
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,QPACP Kidd
FORMAT PACKAGE/DBRM NAME
* MVC O3LOCN,QPACLOCN
FORMAT LOCATION NAME
* MVC O3COLN,QPACCOLN
FORMAT COLLECTION NAME

STCKCONV STCKVAL=QPACSCT,
CONVVAL=WORKCL16,
TIMETYPE=DEC, X
DATETYPE=YYYYMMDD
BAL R14,CONV2SEC
MVC 03ACSCT=X'202020202020214B202020'
ED 03ACSCT,WORKPL8A+2
MVC 04ACSCT,WORKPL8A
STCKCONV STCKVAL=QPACTJST, X
CONVVAL=WORKCL16, X
TIMETYPE=DEC, X
DATETYPE=YYYYMMDD
BAL R14,CONV2SEC
MVC 03ACTJST=X'202020202020214B202020'
ED 03ACTJST,WORKPL8A+2
MVC 04ACTJST,WORKPL8A
STCKCONV STCKVAL=QPACAWTI, X
CONVVAL=WORKCL16, X
TIMETYPE=DEC, X
DATETYPE=YYYYMMDD
BAL R14,CONV2SEC
MVC 03ACAWTI=X'202020202020214B202020'
ED 03ACAWTI,WORKPL8A+2
MVC 04ACAWTI,WORKPL8A
STCKCONV STCKVAL=QPACAWTL, X
CONVVAL=WORKCL16, X
TIMETYPE=DEC, X
DATETYPE=YYYYMMDD
BAL R14,CONV2SEC
MVC 03ACAWTL=X'202020202020214B202020'
ED 03ACAWTL,WORKPL8A+2
MVC 04ACAWTL,WORKPL8A
STCKCONV STCKVAL=QPACAWTR, X
CONVVAL=WORKCL16, X
TIMETYPE=DEC, X
DATETYPE=YYYYMMDD
BAL R14,CONV2SEC
MVC 03ACAWTR=X'202020202020214B202020'
ED 03ACAWTR,WORKPL8A+2
MVC 04ACAWTR,WORKPL8A
STCKCONV STCKVAL=QPACAWTE, X
CONVVAL=WORKCL16, X
TIMETYPE=DEC, X
DATETYPE=YYYYMMDD
BAL R14,CONV2SEC
MVC 03ACAWTE=X'202020202020214B202020'
ED 03ACAWTE,WORKPL8A+2
MVC 04ACAWTE,WORKPL8A
MVC PRNTLINE,O3RECORD
PUT SMFSOUT,PRNTLINE
MVC 04PLAN,O3PLAN
MVC 04PKID,O3PKID
MVC 04TYPE,O3TYPE
PUT SMFDOUT,04RECORD
MVC 03RECORD,=132CL1'
MVC 04RECORD,=80CL1'
CLC QPACRECN,WHPKRECN IF PKGNO => MAX PACKGES
BNL R5,QPACEND Y. GET OUT OF LOOP
LA R5,QPACEND N. BUMP TO NEXT PKG/DBRM
MVC 03PLAN,O1PLAN REMEMBER PREV INFO
MVC 03DATE,O1DATE DITTO
MVC 03TIME,O1TIME DITTO
BCT R8,FQPAC0Ø1 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'**
** : : : : : DAYS**
** OUTPUT IS PLØ8 'SS SS SS SS SS SS TT TC'**
CONV2SEC DS ØH
ST R14,CONVSAYE
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 S TT 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
(continued)

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
CVD R1,WORKDWD1
CONV2SEX

DP WORKDWD1,=P'100'
DP WORKDWD1(6),=P'3000'
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
PDAT999 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 O3ACSC,=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 O3ACSCST,=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'20202020202020202021'
ED COUNTERØ,COUNTER1
MVC PRNTLINE+25(9),COUNTERØ
PUT SYSPRINT,PRNTLINE
MVC PRNTLINE(25),=C' TOTAL ACCOUNTNG RECORDS :'
MVC COUNTERØ,=X'20202020202020202021'
ED COUNTERØ,COUNTER2
MVC PRNTLINE+25(9),COUNTERØ
PUT SYSPRINT,PRNTLINE
MVC PRNTLINE(25),=C' TOTAL RECORDS SELECTED :
MVC COUNTERØ,=X'20202020202020202021'
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

FINALSADV 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'

GETSYØ1Ø BAL R14,READNØØØ READ SYSIN CARD
CLI READNFLG,X'FF' IF EOF
BE GETSYØ00 Y. EXIT
CLC I1KEYWRD,KWRDSSID IF SSID KEYWORD
BE GETSYØ2Ø
CLC I1KEYWRD,KWRDPLAN IF PLAN KEYWORD
BE GETSYØ3Ø
CLC I1KEYWRD,KWRDPAKG IF PAKG KEYWORD
BE GETSYØ4Ø
CLC I1KEYWRD,KWRDDATE IF DATE KEYWORD
BE GETSYØ5Ø
B GETSYØ1Ø
GETSYØ2Ø MVC CRITSSID,I1PARM
B GETSYØ1Ø
GETSYØ3Ø MVC CRITPLAN,I1PARM
B GETSYØ1Ø
GETSYØ4Ø MVC CRITPAKG,I1PARM
B GETSYØ1Ø
GETSYØ5Ø MVC CRITDATE,I1PARM
B GETSYØ1Ø
GETSYØ00 MVC PRNTLINE(25),=C' SELECTION CRITERIA USED:'
PUT SYSPRINT,PRNTLINE
MVC PRNTLINE,=132CL1' '

GETSYØ10 CLC CRITSSID,=C'
BE GETSYØ2Ø
MVC PRNTLINE+1(5),KWRDSSID
MVC PRNTLINE+6(4),CRITSSID
PUT SYSPRINT,PRNTLINE
GETSYØ2Ø CLC CRITPLAN,=C'
BE GETSYØ3Ø
MVC PRNTLINE+1(5),KWRDPLAN
MVC PRNTLINE+6(8),CRITPLAN
PUT SYSPRINT,PRNTLINE
GETSYØ3Ø CLC CRITPAKG(8),=C'
BE GETSYØ4Ø
MVC PRNTLINE+1(5),KWRDPAKG
MVC PRNTLINE+6(8),CRITPAKG
PUT SYSPRINT,PRNTLINE
GETSYØ4Ø 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'0'
********************************************************** READN000 *************
** READ SYSIN CARDS
READN000 DS ØH
ST R14,READNSAV
GET SYSIN,II1RECORD
READN999 DS ØH
L R14,READNSAV
BR R14
READNEOF DS ØH
MVI READNFLG,X'FF'
B READN999
READNSAV DS F'0'
READNFLG DS X'0'
********************************************************** READS000 *************
** READ SMF RECORDS
** R3 POINTS TO THE BUFFER OF THE RECENTLY READ RECORD
READS000 DS ØH
ST R14,READSSAV
GET SMFDATA
LR R3,R1 SAVE BUFFER AFTER READ
USING SM101,R3 ANCHOR R3 TO SMF101 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'0'
READSFLG DS X'0'
********************************************************** CHECK000 ************
** CHECK RECORD TYPE IS 101
** AND SSID
CHECK000 DS ØH
ST R14,CHECKSAV
MVI CHECKFLG,X'00' ASSUME RECORD IS GOOD
SR R15,R15 CLEAR REG
ICM R15,1,SM101RTY INSERT RECORD TYPE
CH R15,=H'101' IF TYPE 101
BNE CHECK010 N. SKIP RECORD
CLC SM101SSI,CRITSSID IF SUBSYSTEM IS CORRECT
BNE CHECK010 N. SKIP RECORD
AP COUNTER2.=P.'1' COUNT RECORDS PROCESSED
B CHECK999
CHECKØ10 MVI CHECKFLG,X.'FF' MARK FOR SKIPPING
CHECK999 DS 0H
L R14,CHECKSAV
BR R14
CHECKSAV DS F.'0'
CHECKFLG DS X.'0'
** WORKING STORAGE **
SYSIN DCB DSORG=PS,MACRF=(GM).
DDNAME=SYSIN,EODAD=READNEOF
SMFDATA DCB DSORG=PS,MACRF=GL.
DDNAME=SMFDATA,EODAD=READSEOF,BFTEK=A
SMFSOUT DCB DSORG=PS,RECFM=F,MACRF=(PM),LRECL=132,BLKSIZE=3036.
DDNAME=SMFSOUT
SMFDOUT DCB DSORG=PS,RECFM=FB,MACRF=(PM),LRECL=80,BLKSIZE=23440.
DDNAME=SMFDOUT
SYSPRINT DCB DSORG=PS,RECFM=F,MACRF=(PM),LRECL=132,BLKSIZE=3036.
DDNAME=SYSPRINT
SNAPDUMP DCB DSORG=PS,RECFM=VBA,MACRF=(W),LRECL=125,BLKSIZE=1632.
DDNAME=SNAPDUMP
SNAPAREA DS 0H
DS 18F.'0'
DS 2F
SNAPAEND DS 0H
*********************************************************************** WORKAREA DSECT ***************
WORKAREA DSECT
SAVEAREA DC 18F.'0'
WHPKRECN DS H
WHIFCID DS H
DS 0D
WORKCL16 DS CL16
WORKZL15 DS ZL15
WORKZL01 DS ZL01
WORKPL8A DS PL08
WORKPL8B DS PL08
WORKPL8C DS PL08
CRITERIA DS OC6L1
KWRRDSID DS CL05
CRITSSID DS CL04
DS CL01
KWRRDPNDS DS CL05
CRTRPLAN DS CL08
DS CL01
KWRRDPKGS DS CL05
CRTRPKG DS CL18
DS CL01
KWRRDPLA DS CL05

WORKLEN EQU *-WORKAREA

***************************************** SM101 DSECT ***************
** THIS DSECT DEFINES THE RECORD LAYOUT OF SMF TYPE 101 RECORD
** WHICH IS THE DB2 ACCOUNTING TRACE RECORD
SMFRECRD DSNDQWAS DSECT=YES,SUBTYPE=ALL

***************************************************************************
REGISTER EQUATES ***********
R0 EQU 0
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
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15

//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=(1024,(50,50)),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=*  
//SMFSDOUT DD  DISP=SHR,DSN=MYTSOID.SMFSUMM
//SYSIN    DD  *
SSID=SSID

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

Larry Prestosa (USA) © Xephon 1998
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 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
• RSQSL – 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
/* 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)"
```
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
      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'
  END
  text=''
  line1="Select substr('REBIND PLAN('concat name"
  line2="concat')                        ',1,45)"
  line3="from sysibm.sysplan                    
  line4=''
  line5=''
  SELECT
    WHEN(ff='1') THEN DO
      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'
  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) © Xephon 1998
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 online 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.