



150

DB2

April 2005

In this issue

- [3 Providing a portable and customized Control Center in DB2 Universal Database Version 8.2](#)
 - [9 REXX/ISPF program to help programmers to resolve sqlcode -805 – part 2](#)
 - [24 DSNWSPM – a useful but undocumented stored procedure](#)
 - [28 Move data across DB2 subsystems to another LPAR](#)
 - [46 DB2 news](#)
-

update

© Xephon Inc 2005

DB2 Update

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

Phone: 214-340-5690

Fax: 214-341-7081

Editor

Trevor Eddolls

E-mail: trevore@xephon.com

Publisher

Colin Smith

E-mail: info@xephon.com

Subscriptions and back-issues

A year's subscription to *DB2 Update*, comprising twelve monthly issues, costs \$380.00 in the USA and Canada; £255.00 in the UK; £261.00 in Europe; £267.00 in Australasia and Japan; and £265.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 2000 issue, are available separately to subscribers for \$33.75 (£22.50) each including postage.

***DB2 Update* on-line**

Code from *DB2 Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at <http://www.xephon.com/db2>; you will need to supply a word from the printed issue.

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.

Contributions

When Xephon is given copyright, articles published in *DB2 Update* are paid for at the rate of \$160 (£100 outside North America) per 1000 words and \$80 (£50) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of \$32 (£20) per 100 lines. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

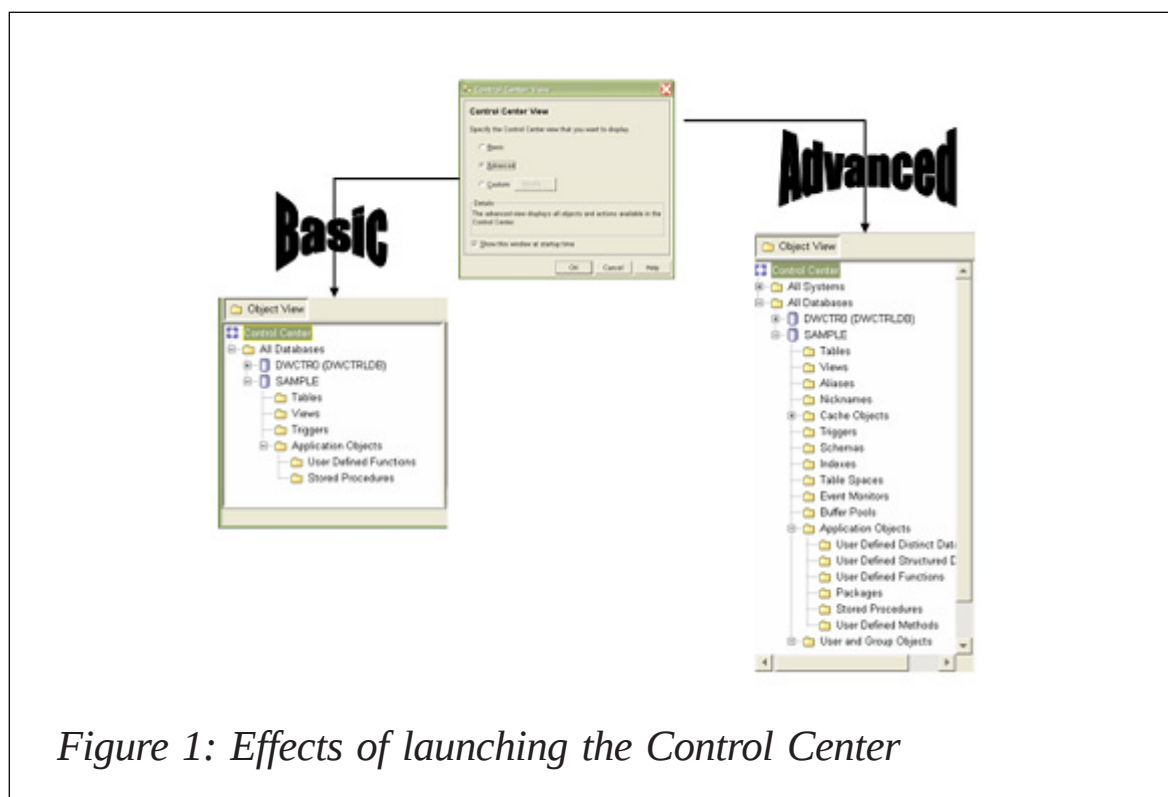
© Xephon Inc 2005. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in 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.

Printed in England.

Providing a portable and customized Control Center in DB2 Universal Database Version 8.2

The IBM DB2 Universal Database for Linux, Unix, and Windows Version 8.2 product (DB2 UDB v8.2) delivers countless new features for database administrators (DBAs) and application developers alike. Before the DB2 UDB v8.2 release, the Control Center provided a complete range of functionality and features for all users, regardless of their skill sets. For inexperienced DBAs or power users, the complete range of the Control Center's functionality presented all at once could complicate the navigation of the Control Center and make task execution less efficient.

The DB2 UDB v8.2 release adds 'personas' to the Control Center to simplify navigation and complexity for those users who don't need an all-encompassing total management tool. Using a persona, users can personally customize the Control Center capabilities so they can accomplish their tasks more



easily and navigate the Control Center more efficiently.

Figure 1 shows the effects of launching the Control Center in DB2 UDB v8.2 in both the Basic and Advanced personas.

You can see the differences between the two personas in Figure 1. The Basic persona lists the databases that are available for administration and applies the 80/20 rule with respect to the objects that you can work with. On the other hand, the Advanced persona lists both system servers and every object of a database that you can work with. (Note that to view a list of all the objects that you can work with in a database requires the use of a scroll bar.)

When you start the Control Center in DB2 UDB v8.2, you can set the views for Basic, Advanced, and Custom functionality. (You can disable this option so that personas can be set using only the **Tools/Customize Control Center** function in the Control Center's action menu bar.)

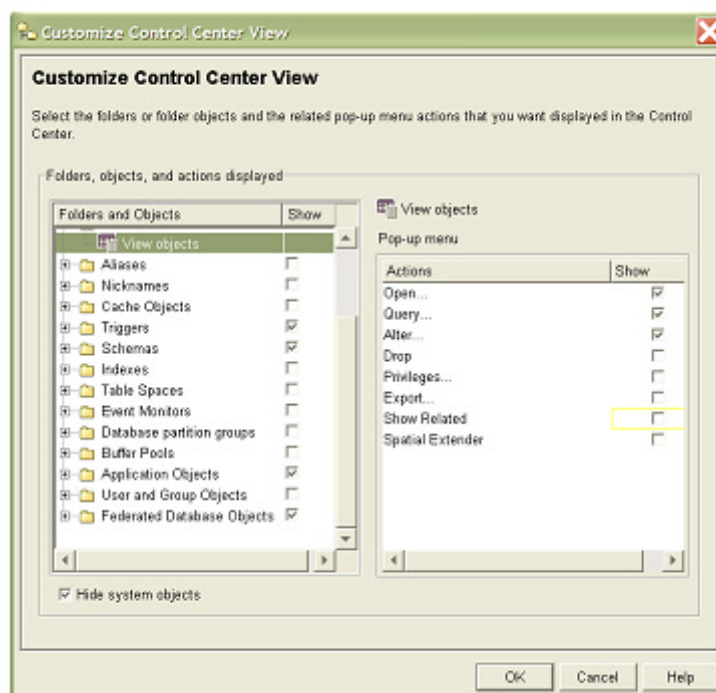


Figure 2: Customized Control Center

The Custom functionality allows you to tailor the view of the Control Center to match your needs. For example, let's assume that you are a DBA who wants the simplicity of the Basic persona, but you also use WebSphere Information Integrator to work with a Microsoft SQL Server database that resides in your network. To satisfy this requirement, you can customize the Control Center so that it shows all of the objects in the Basic persona and any federated database objects you may have defined on your system.

Using a custom approach, you can customize not only the objects that appear in the Control Center but also the actions that you can perform on them. In Figure 2, I've configured the Control Center to match the Basic persona, but customized it so that I can work with federated database objects, and limited some of the actions I can perform on views that are not part of the Basic persona configuration.

In short, the persona feature of the Control Center allows users to:

- 1 Assume different personas (Basic, Advanced, and Custom)

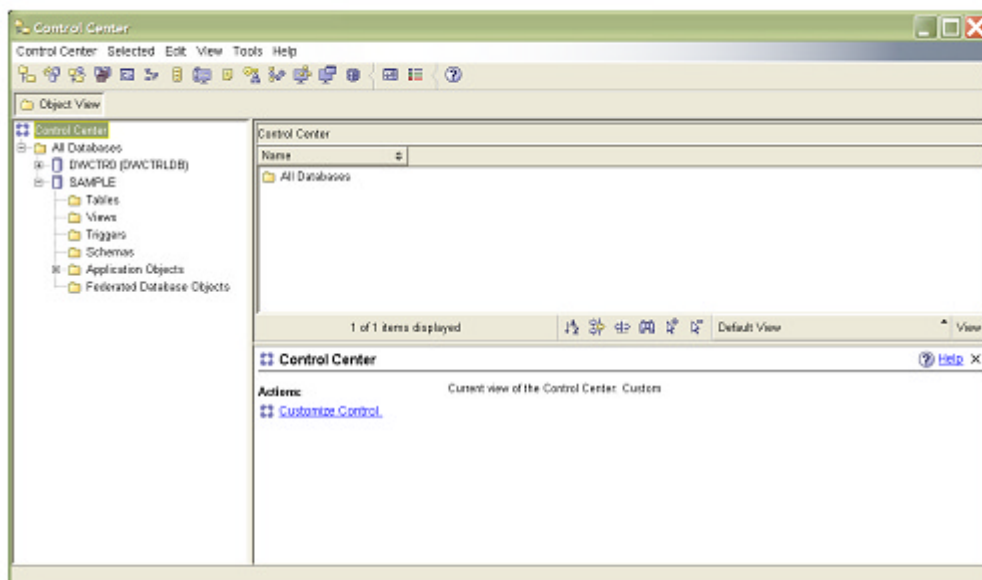


Figure 3: Additional Federated Database Objects folder

and pick the one that best suits their needs.

- 2 Enjoy simplified and improved Control Center navigation.
- 3 Choose to create a hybrid persona using the Custom persona.

After you customize the Control Center view and actions that can be performed there, it will display the objects you defined in this persona until you select a different persona or change your customization. In Figure 3, you can see the additional Federated Database Objects folder.

While this functionality is powerful indeed, there is no documented way to replicate your customizations across different servers. You may be deploying the Control Center as an administration point for a group of junior DBAs and want to restrict the actions that they can perform from this interface, or perhaps you are an Independent Software Vendor (ISV) who builds applications that run on DB2 UDB and you want to

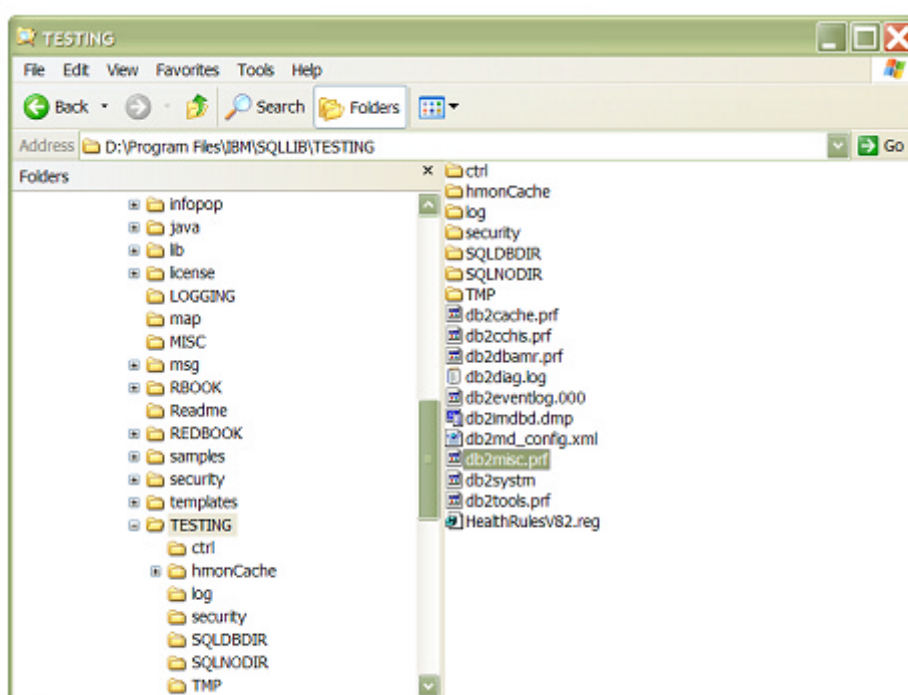
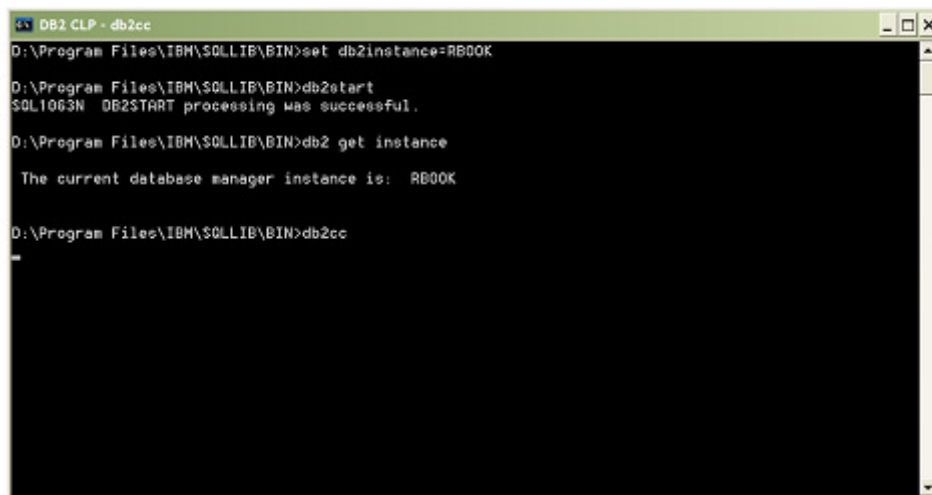


Figure 4: db2misc.prf file location



```
DB2 CLP - db2cc
D:\Program Files\IBM\SQLLIB\BIN>set db2instance=RB00K

D:\Program Files\IBM\SQLLIB\BIN>db2start
SQL063N DB2START processing was successful.

D:\Program Files\IBM\SQLLIB\BIN>db2 get instance

The current database manager instance is: RB00K

D:\Program Files\IBM\SQLLIB\BIN>db2cc
```

Figure 5: Starting the new instance

customize the Control Center so that the DBAs would be exposed only to objects that you want them to work with. In neither case is your customization officially reproducible.

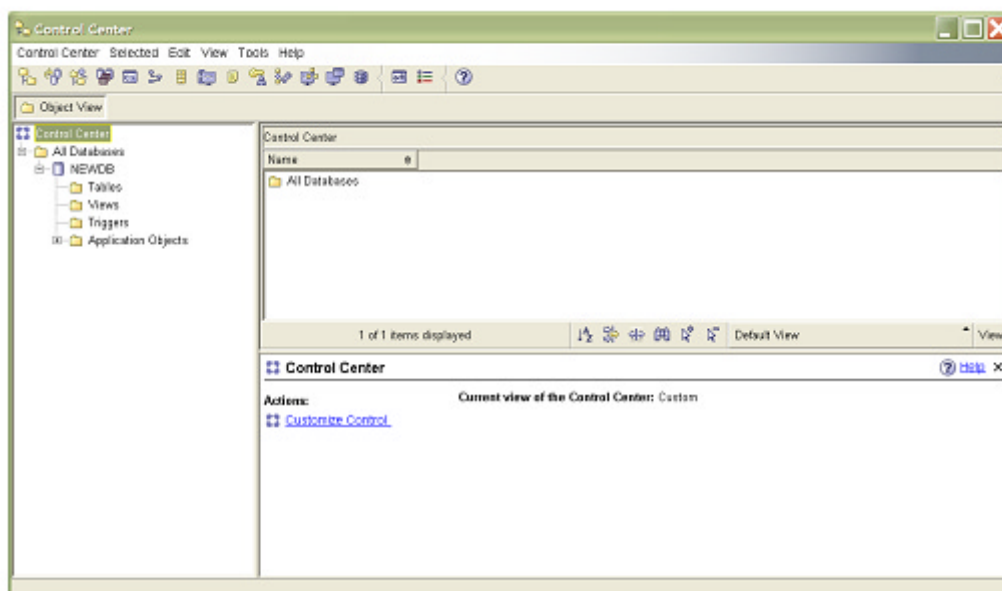


Figure 6: Duplicating customization

DB2 UDB v8.2 has no support for Custom profile importing and exporting; however, most of the Control Center's persona information is stored in the db2misc.prf file. You can find this file in your instance's directory under the <database_path>\sqllib\ directory. For example, in Figure 3, I customized the Control Center for the TESTING instance, and the db2misc.prf file therefore resides in the sqllib\TESTING\ directory, as shown in Figure 4.

If you take the db2misc.prf file and copy it to another instance directory (on the same server or a different one) you can take with it the customized persona that you've defined.

For example, on a separate server in an instance called RBOOK, I overwrote the db2misc.prf file in the appropriate directory and started the Control Center. As you can see in Figures 5 and 6, the customizations I made for the TESTING instance were duplicated in the RBOOK instance, even though it has a completely different database structure. In fact, I didn't even have to take down the instance for the changes to take effect.

CUSTOMIZE ME

While the movement of customized personas for the Control Center is not a documented or supported feature (which is why I used the word 'most' when describing the content of the db2misc.prf file), you may find the information in this article is just what you've been looking for if you're taking advantage of this new feature in DB2 UDB v8.2.

Because the solution that I suggest you try simply involves the movement of a preference file, it can be automated by a script file that you can PUSH or PULL using distributed software or applied as part of a post-installation script with your software. It may even be the case that you want to have your profile for use on whatever systems you might be administering if you're a consultant. Whatever way you choose to leverage this functionality, I'm sure you will find this undocumented trick helpful.

One more thing, just in case... Before you copy in your version of the db2misc.prf file, rename the old one to db2misc.prf.old – that's just good practice.

Paul C Zikopoulos (paulz_ibm@msn.com)
IBM Database Competitive Technology team
IBM (Canada)

© IBM 2005

REXX/ISPF program to help programmers to resolve sqlcode -805 – part 2

This month we conclude the code to help programmers avoid sqlcode -805 errors.

```
/*-----*/
ALLOCSYS:
ALLOCSYSRC = 0
DSNIN      = USERID() || '.PLN' || '.IN'  || TIME('S')
DSNREC     = USERID() || '.PLN' || '.REC' || TIME('S')
DSNPRINT   = USERID() || '.PLN' || '.PRT' || TIME('S')
ADDRESS TSO "ALLOC FILE(SYSIN)  DATASET('"DSNIN"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(80) BLKSIZE(27920) RECFM(F B) SPACE(1,1) CYL"
            IF RC <> 0 THEN
                DO
                    ADDRESS ISPEXEC "SETMSG MSG(DBC001)"
                    ALLOCSYSRC = 1
                    RETURN
                END
            END

/*-----*/
ADDRESS TSO "ALLOC FILE(SYSREC00) DATASET('"DSNREC"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
            "LRECL(100) BLKSIZE(27900) RECFM(F B) SPACE(1,1) CYL"
            IF RC <> 0 THEN
                DO
                    ADDRESS ISPEXEC "SETMSG MSG(DBC002)"
                    ALLOCSYSRC = 1
                    RETURN
                END
            END

/*-----*/
ADDRESS TSO "ALLOC FILE(SYSPRINT) DATASET('"DSNPRINT"') " ,
            "NEW CAT REUSE UNIT(SYSDA)" ,
```

```

        "LRECL(133) BLKSIZE(27930) RECFM(F B) SPACE(1,1) CYL"
        IF RC <> 0 THEN
            DO
                ADDRESS ISPEXEC "SETMSG MSG(DBC003)"
                ALLOCSYSRC = 1
                RETURN
            END
        RETURN
    /*-----*/
LIBDEFWIND0:
SELECT
    WHEN SSID = 'DB2P' THEN
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX1.PUBLIC.WINDOWS')"
    WHEN SSID = 'DB2T' THEN
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX2.PUBLIC.WINDOWS')"
    WHEN SSID = 'DB2D' THEN
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX3.PUBLIC.WINDOWS')"
    OTHERWISE
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX1.PUBLIC.WINDOWS')"
END
RETURN
/*-----*/
LIBDEFWINDJ:
SELECT
    WHEN SSID = 'DB2P' THEN
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX1.PUBLIC.WINDOWSJ')"
    WHEN SSID = 'DB2T' THEN
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX2.PUBLIC.WINDOWSJ')"
    WHEN SSID = 'DB2D' THEN
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX3.PUBLIC.WINDOWSJ')"
    OTHERWISE
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX1.PUBLIC.WINDOWSJ')"
END
RETURN
/*-----*/
LIBDEFPANEL:
SELECT
    WHEN SSID = 'DB2P' THEN
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX1.PUBLIC.PANELLIB')"
    WHEN SSID = 'DB2T' THEN
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX2.PUBLIC.PANELLIB')"
    WHEN SSID = 'DB2D' THEN
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX3.PUBLIC.PANELLIB')"
    OTHERWISE
        "ISPEXEC LIBDEF ISPPLIB DATASET ID('PREFIX1.PUBLIC.PANELLIB')"
END
RETURN
/*-----*/
DELTMP:
W = OUTTRAP(DELTMP.)

```

```

ADDRESS TSO "DELETE ('"DSNIN"')"
ADDRESS TSO "DELETE ('"DSNREC"')"
ADDRESS TSO "DELETE ('"DSNPRINT"')"
W = OUTTRAP('OFF')
RETURN
/*-----*/
FINAL:
ADDRESS ISPEXEC 'LIBDEF ISPPLIB'
ADDRESS ISPEXEC 'LIBDEF ISPMLIB'
EXIT 0;
/*----- end of code ----- carlos-osorio@excite.com --*/

```

PREFIX0.PUBLIC.PANELI(DB2PLNSS)

Irecl 80, blksize 27920.

```

)ATTR
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
  color(turquoise)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
_ TYPE(INPUT)   INTENS(HIGH) CAPS(ON)          HILITE(USCORE)
] TYPE(OUTPUT)  INTENS(LOW)  CAPS(ON) JUST(LEFT) HILITE(REVERSE)
  color(turquoise)
@ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)          HILITE(REVERSE)
  color(turquoise)
$ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)          HILITE(REVERSE)
  color(blue)
)BODY EXPAND(\\)
@                                     $DB2 Subsystems@
@
%COMMAND ==>_ZCMD                      +%SCROLL ==>_SCRO+
%
%          Option  DB2 Subsystem
+          %-----  -----
)MODEL
+          _0+    ]SDB2+
)INIT
  .help    = DB2PLNSH
  .cursor  = 0
  &SCRO    = PAGE
)REINIT
  .cursor  = 0
)PROC
if (.PFKEY = 'PF01')
  &pfkeyin = 's'
VER(&0,LIST,S)
)END

```

PREFIX0.PUBLIC.PANELI(DB2PLNSH)

lrecl 80, blksize 27920.

```
)attr
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
  color(turquoise)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
- TYPE(INPUT)   INTENS(HIGH) CAPS(ON)          HILITE(USCORE)
] TYPE(OUTPUT)  INTENS(LOW)  CAPS(ON) JUST(LEFT) HILITE(REVERSE)
  color(turquoise)
@ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)          HILITE(REVERSE)
  color(turquoise)
$ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)          HILITE(REVERSE)
  color(blue)
)BODY EXPAND(\\)
@                                     $DB2 Subsystems@
@
%   Options:+
+
+   %S+          - To choice the DB2 subsystem
+
%   %Description of the field:+
+
+   %DB2 Subsystem+ Name/Identification of a DB2 Subsystem inscribed
+                   in the system operative (SYS1.PARMLIB).
+
)END
```

PREFIXX.PUBLIC.PANELIB(DB2PLNIN)

lrecl 80, blksize 27920.

```
)attr
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
  color(blue)
- TYPE(INPUT)   INTENS(LOW)  CAPS(ON) JUST(LEFT) HILITE(REVERSE)
  color(blue)
@ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)          HILITE(REVERSE)
  color(turquoise)
$ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)          HILITE(REVERSE)
  color(blue)
)BODY EXPAND(\\)
@                                     $HANDLE OF DB2 PLANS/COLLECTIONS/PACKAGES@
@
%
%   Usuario+&ZUSER
%   Provide the Plan name                                     Fecha   +&Fecha
```

```

%      Press Enter.                                Hora      +&Hora
+
+
%      1. Plan+_PLAN      +
+
+
)INIT
  .help      = DB2PLNiH
  .cursor    = Plan
  &Fecha     = '&ZDAY..&ZMONTH..&ZYEAR'
  &Hora      = &ZTIME
)REINIT
  REFRESH (*)
  &Hora      = &ZTIME
)PROC
  REFRESH (*)
  VER(&PLAN, NONBLANK, NAME)
)END

```

PREFIXX.PUBLIC.PANELIB(DB2PLNIH)

lrecl 80, blksize 27920.

```

)attr
% TYPE(TEXT)  INTENS(HIGH) SKIP(ON)
+ TYPE(TEXT)  INTENS(LOW)  SKIP(ON)
  color(blue)
_ TYPE(INPUT) INTENS(LOW)  CAPS(ON) JUST(LEFT) HILITE(REVERSE)
  color(blue)
@ TYPE(TEXT)  INTENS(HIGH) CAPS(ON)                HILITE(REVERSE)
  color(turquoise)
$ TYPE(TEXT)  INTENS(HIGH) CAPS(ON)                HILITE(REVERSE)
  color(blue)
)BODY EXPAND(\\)
@                                $HANDLE OF DB2 PLANS/COLLECTIONS/PACKAGES@
@
%
Usuario+&ZUSER
%      Provide the Plan name                                Fecha  +&Fecha
%      Press Enter.                                         Hora    +&Hora
+
+
%      Plan:+      Is the DB2 plan involved in the sqlcode -805
+
+
)INIT
  &Fecha     = '&ZDAY..&ZMONTH..&ZYEAR'
  &Hora      = &ZTIME
)REINIT

```

```

REFRESH (*)
&Hora      = &ZTIME
)END

```

PREFIXX.PUBLIC.WINDOWSJ(DB2TPLNS)

lrecl 60, blksize 6000.

```

)ATTR
$ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)
] TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(WHITE)
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
_ TYPE(INPUT)   INTENS(HIGH) CAPS(ON)
@ TYPE(OUTPUT) INTENS(LOW)   CAPS(ON) JUST(RIGHT)
  HILITE(REVERSE) COLOR(TURQUOISE)
? TYPE(OUTPUT) INTENS(LOW)   CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(TURQUOISE)
)BODY WINDOW(34,19) ASIS
+
$COLLECTIONS OF THE PLAN:]PLAN
$-----
%COMMAND=>_ZCMD    +%SCROLL =>_SCRO
%
%OPTION  COLLECTION          SEQNO
+-----+-----+-----+
)MODEL
+ _O+   ?COLLID              +@Z
)INIT
.HELP   = DB2TPLNH
.ZVARS  ='(SEQNO) '
&FECHA  = '&ZDAY..&ZMONTH..&ZYEAR'
&HORA   = &ZTIME
&SCRO   = PAGE
&O      = ' '
)REINIT
&HORA   = &ZTIME
&O      = ' '
REFRESH(0)
)PROC
VER (&O, NONBLANK, LIST, A, I, D, P)
)END

```

PREFIXX.PUBLIC.WINDOWSJ(DB2TPLNH)

lrecl 60, blksize 6000.

```

)ATTR
$ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)
] TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(WHITE)
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
  COLOR(TURQ)
/ TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
  COLOR(RED)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
)BODY WINDOW(34,19) ASIS
+
$COLLECTIONS OF THE PLAN:]PLAN
$-----
+
+%OPTION+- CHOICE%A,I,D,P+
+   %A+  ADD COLLECTION
+   %I+  INSERT COLLECTION(SAME)
+   %D+  DELETE COLLECTION
+   %P+  PACKAGES IN COLLECTION
+
+%COLLECTION+- COLLECTIONS BELONG
+           THE PLAN
+
+%SEQNO+ - SEQUENCE NUMBER OF THE
+           COLLECTION IN THE PLAN
)END

```

PREFIXX.PUBLIC.WINDOWSJ(DB2TPLUS)

lrecl 60, blksize 6000.

```

)ATTR
$ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)
] TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(WHITE)
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
_ TYPE(INPUT)   INTENS(HIGH) CAPS(ON)
? TYPE(OUTPUT) INTENS(LOW)  CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(TURQUOISE)
)BODY WINDOW(35,13) ASIS
$  THE PLAN:]PLAN    $ IS IN USE
$  ADD/DELETE COLLECTION NOT PROCEED
$-----
%COMMAND =>_ZCMD    +%SCROLL =>_SCRO
%
%      ID      CONECTION      AUTHID

```

```

+ -----
)MODEL
+ ?ID      + ?NAME      + ?AUTHID +
)INIT
  .HELP    = DB2TPLUH
  &FECHA   = '&ZDAY..&ZMONTH..&ZYEAR'
  &HORA     = &ZTIME
  &SCRO     = PAGE
)REINIT
  &HORA     = &ZTIME
)PROC
)END

```

PREFIXX.PUBLIC.WINDOWSJ(DB2TPLUH)

lrecl 60, blksize 6000.

```

)ATTR
$ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)
] TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(WHITE)
% TYPE(TEXT)   INTENS(HIGH) SKIP(ON)
  COLOR(TURQ)
+ TYPE(TEXT)   INTENS(LOW)  SKIP(ON)
)BODY WINDOW(35,13) ASIS
$  THE PLAN:]PLAN    $ IS IN USE
$ ADD/DELETE COLLECTION NOT PROCEED
$-----
+
+%ID      +- ID OF THE PROCESS USING
+          THE PLAN TO MODIFY (JOB,
+          TSO, CICS, STARTED TASK)
+
+%CONNECTION+- CONNECTION ID OF THE
+          PROCESS USING THE PLAN
+
+%AUTHID+- AUTHORIZATION ID OF THE
+          PROCESS USING THE PLAN
)END

```

PREFIXX.PUBLIC.WINDOWSJ(DB2TPKGS)

lrecl 60, blksize 6000.

```

)ATTR
$ TYPE(TEXT)   INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)

```

```

] TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(WHITE)
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
_ TYPE(INPUT)   INTENS(HIGH) CAPS(ON)
@ TYPE(OUTPUT) INTENS(LOW)   CAPS(ON) JUST(RIGHT)
  HILITE(REVERSE) COLOR(TURQUOISE)
? TYPE(OUTPUT)                CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR1)
; TYPE(OUTPUT)                CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR1)
)BODY WINDOW(60,16) ASIS
$  PACKAGES OF THE COLLECTION]COLLID          $
$-----
%COMMAND =>_ZCMD                                +%SCROLL =>_SCRO+
%
%0 PACKAGE      BINDTIME          PCTIME          CONTOKEN      OPE
+- -----
)MODEL          CLEAR(OPE)
_P?PACKAGE ?BINDTIME      ?PCTIME          ?CONTOKEN      +;Z+
)INIT
  .HELP    = DB2TPKGH
  .ZVARS   ='(OPE) '
  &FECHA   = '&ZDAY..&ZMONTH..&ZYEAR'
  &HORA    = &ZTIME
  &SCRO    = PAGE
  &P       = ' '
  IF (&OPE = 'N') &COLOR1 = 'RED'
  ELSE                &COLOR1 = 'TURQUOISE'
)REINIT
  &HORA    = &ZTIME
  &P       = ' '
  REFRESH(P)
)PROC
  VER (&P, NONBLANK, LIST, C, V)
)END

```

PREFIXX.PUBLIC.WINDOWSJ(DB2TPLGH)

lrecl 60, blksize 6000.

```

)ATTR
$ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)
] TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(WHITE)
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
                  COLOR(TURQ)
/ TYPE(TEXT)    INTENS(HIGH) SKIP(ON)

```

```

                                COLOR(RED)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
)BODY WINDOW(60,16) ASIS
$  PACKAGES OF THE COLLECTION]COLLID          $
$-----
+
+ %O OPTION+- CHOICE%R,V+
+           %R+ REBIND PACKAGE
+           %V+ VIEW LOAD LIBRARIES WITH SAME MODULE-CONTOKEN
+
+ %PACKAGE +- NAME OF PACKAGE
+
+ %BINDTIME+- TIME OF THE LAST BIND
+
+ %PCTIME  +- TIME OF THE PRE-COMPILE
+
+ %CONTOKEN+- CONTOKEN OF THE PACKAGE
+
+ %OPERATIVE+-IF THE PACKAGE IS OPERATIVE OR NOT
)END

```

PREFIXX.PUBLIC.WINDOWSJ(DB2TCOLS)

lrecl 60, blksize 6000.

```

)ATTR
$ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)
] TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(WHITE)
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
_ TYPE(INPUT)   INTENS(HIGH) CAPS(ON)
@ TYPE(OUTPUT) INTENS(LOW)   CAPS(ON) JUST(RIGHT)
  HILITE(REVERSE) COLOR(TURQUOISE)
? TYPE(OUTPUT)           CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR1)
{ TYPE(OUTPUT)           CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR1)
)BODY WINDOW(60,16) ASIS
$  COLLECTIONS WHERE IS PRESENT THE PACKAGE]PACKAGE $
$-----
%COMMAND =>_ZCMD                      +%SCROLL =>_SCRO+
%
%      COLLECTION                PCTIME                CONTOKEN
+  -----
)MODEL
+ ?COLLIDC          +   ?PCTIMEC          +   ?CONTOKEC
)INIT

```

```

.HELP      = DB2TCOLH
&FECHA     = '&ZDAY..&ZMONTH..&ZYEAR'
&HORA      = &ZTIME
&SCRO      = PAGE
)REINIT
&HORA      = &ZTIME
)PROC
)END

```

PREFIXX.PUBLIC.WINDOWSJ(DB2TCOLH)

lrecl 60, blksize 6000.

```

)ATTR
$ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)
] TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(WHITE)
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
  COLOR(TURQ)
/ TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
  COLOR(RED)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
)BODY WINDOW(60,16) ASIS
$      COLLECTIONS WHERE IS PRESENT THE PACKAGE]PACKAGE$
$-----
+
+ %COLLECTION +- ONE (OR MORE) COLLECTIONS WHERE THE PACKAGE
+                IS FOUNDED
+
+ %PCTIME      +- TIME OF THE PRE-COMPILE OF THE PACKAGE
+
+ %CONTOKEN    +- CONTOKEN OF THE PACKAGE
+
)END

```

PREFIXX.PUBLIC.WINDOWSJ(DB2TLOAS)

lrecl 60, blksize 6000.

```

)ATTR
$ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
_ TYPE(INPUT)   INTENS(HIGH) CAPS(ON)
] TYPE(OUTPUT)  CAPS(ON)
  HILITE(REVERSE) COLOR(TURQ)

```

```

? TYPE(OUTPUT)                CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR1)
¿ TYPE(OUTPUT)                CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR2)
@ TYPE(OUTPUT)                CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR3)
\ TYPE(OUTPUT)                CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR4)
/ TYPE(OUTPUT)                CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR5)
Ñ TYPE(OUTPUT)                CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR6)
¡ TYPE(OUTPUT)                CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR7)
¢ TYPE(OUTPUT)                CAPS(ON) JUST(LEFT)
  HILITE(REVERSE) COLOR(&COLOR8)
)BODY WINDOW(59,16) ASIS
$LOAD LIBRARIES TO SEARCH CORRELATION TO PACKAGE ]PACKAGE
$IN COLLID]COLLID              $AND CONTOKEN]CONTOKEN
$-----
%
%   LOAD LIBRARY                                RESULT
+   -----
%1._LIB1                                ?RESULT1
%2._LIB2                                ¿RESULT2
%3._LIB3                                @RESULT3
%4._LIB4                                \RESULT4
%5._LIB5                                /RESULT5
%6._LIB6                                ÑRESULT6
%7._LIB7                                ¡RESULT7
%8._LIB8                                ¢RESULT8
+   -----
)INIT
  .HELP    = DB2TLOAH
  &FECHA    = '&ZDAY..&ZMONTH..&ZYEAR'
  &HORA     = &ZTIME
)PROC
  VER (&LIB1,NONBLANK,DSNAME)
  IF (&LIB2 ^= ' ')
    VER (&LIB2,DSNAME)
  IF (&LIB3 ^= ' ')
    VER (&LIB3,DSNAME)
  IF (&LIB4 ^= ' ')
    VER (&LIB4,DSNAME)
  IF (&LIB5 ^= ' ')
    VER (&LIB5,DSNAME)
  IF (&LIB6 ^= ' ')
    VER (&LIB6,DSNAME)
  IF (&LIB7 ^= ' ')
    VER (&LIB7,DSNAME)

```

```

IF (&LIB8 ^= ' ')
  VER (&LIB8,DSNAME)
)END

```

PREFIXX.PUBLIC.WINDOWSJ(DB2TLOAH)

lrecl 60, blksize 6000.

```

)ATTR
$ TYPE(TEXT)    INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)
] TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(WHITE)
% TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
  COLOR(TURQ)
/ TYPE(TEXT)    INTENS(HIGH) SKIP(ON)
  COLOR(RED)
+ TYPE(TEXT)    INTENS(LOW)  SKIP(ON)
)BODY WINDOW(59,20) ASIS
$LOAD LIBRARIES TO SEARCH CORRELATION TO PACKAGE ]PACKAGE
$IN COLLID]COLLID          $AND CONTOKEN]CONTOKEN
$-----
+
+%LOAD LIBRARY+- 1 TO 8 DSNAMES OF THE LOAD LIBRARIES TO
+                SEARCH A MODULE WITH THE SAME NAME AND
+                CONTOKEN BELONG THE PACKAGE.
+
+%RESULT          +- RESULTS OF THE SEARCH:
+                %CORRECT  +THE LOAD LIBRARY CONTAINS A
+                MODULE SINCRONIZED WITH THE PACKAGE
+
+                %INCORRECT +THE LOAD LIBRARY CONTAINS A
+                MODULE%NOT+SINCRONIZED WITH THE PACKAGE
+
+                %DSET ERROR+DATASET IS NOT FOUND IN THE
+                CATALOG
+
+                %OPEN ERROR+ERROR IN OPEN LIBRARY
+
+                %NO LOADLIB+THE DATASET IS NOT A LOAD
+                LIBRARY
+
+                %NOT FOUND +MODULE NOT FOUND IN LOAD LIBRARY
)END

```

PREFIXX.PUBLIC.WINDOWS(DB2PLNAD)

lrecl 40, blksize 4000.

```

)attr
_ TYPE(INPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) color(white)
] TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) color(white)
@ TYPE(TEXT) INTENS(HIGH) SKIP(OFF)
  HILITE(REVERSE) color(turquoise)
$ TYPE(TEXT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)
% TYPE(TEXT) INTENS(HIGH) SKIP(ON)
+ TYPE(TEXT) INTENS(LOW) SKIP(ON)
)BODY WINDOW(40,10) ASIS
$ ADD COLLECTION TO PLAN]PLAN %
$-----
%
%
% @COLLECTION NAME =>_COLLIDTG %
%
%
)INIT
  .CURSOR = COLLIDTG
)PROC
VER(&COLLIDTG, NONBLANK)
)END

```

PREFIXX.PUBLIC.WINDOWS(DB2PLNDE)

lrecl 40, blksize 4000.

```

)attr
_ TYPE(INPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) color(RED)
] TYPE(OUTPUT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) color(white)
  JUST(LEFT)
@ TYPE(TEXT) INTENS(HIGH) SKIP(OFF)
  HILITE(REVERSE) color(turquoise)
$ TYPE(TEXT) INTENS(HIGH) CAPS(ON)
  HILITE(REVERSE) COLOR(BLUE)
% TYPE(TEXT) INTENS(HIGH) SKIP(ON)
  COLOR(turquoise)
+ TYPE(TEXT) INTENS(LOW) SKIP(ON)
)BODY WINDOW(40,12) ASIS
$ CONFIRM DELETE OF THE COLLECTION
$-----
%
% Confirm delete of the collection
%
% ]COLLIDTG +

```

```

%   from Plan:
%           ]PLAN      %?  ==>_C%(Y|N)
%
%
%
%
)INIT
  .CURSOR = C
)PROC
VER(&C,NONBLANK,LIST,Y,N)
)END

```

PREFIXX.PUBLIC.MSGLIB(DBC00)

lrecl 80, blksize 27920.

```

DBC001  'FAILURE OF SYSIN ALLOC'      .ALARM=YES
'CHECK FOR OUT OF SPACE IN SYSDA/RACF PERMISSIONS'
DBC002  'FAILURE OF SYSREC ALLOC'     .ALARM=YES
'CHECK FOR OUT OF SPACE IN SYSDA/RACF PERMISSIONS'
DBC003  'FAILURE OF SYSPRINT ALLOC'   .ALARM=YES
'CHECK FOR OUT OF SPACE IN SYSDA/RACF PERMISSIONS'

```

PREFIXX.PUBLIC.MSGLIB(DBC04)

lrecl 80, blksize 27920.

```

DBC041  'REXX LOG NOT ACCESSIBLE' .ALARM=YES
'FAILURE IN DEFINE THE REXX LOG. CHECK RACF ACCESS'

```

PREFIXX.PUBLIC.MSGLIB(DBC70)

lrecl 80, blksize 27920.

```

DBC701  'DB2 PLAN NOT EXIST'          .ALARM=YES .WINDOW=LNORESP
'THE PLAN NOT EXIST. NOT FOUND IN DB2 CATALOG'
DBC702  'COLLECTION ALREADY EXIST'    .ALARM=YES .WINDOW=LNORESP
'THE COLLECCTION TO ADD ALREADY BELONG TO DB2 PLAN'
DBC703  'COLLECTION DELETED'          .ALARM=YES .WINDOW=LNORESP
'THE COLLECCTION HAS BEEN DELETED FROM PLAN SUCCESSFULLY'
DBC704  'COLLECTION ADDED'            .ALARM=YES
'THE COLLECCTION HAS BEEN ADDED TO PLAN SUCCESSFULLY'
DBC705  'COLLECTION NOT ADDED'        .ALARM=YES .WINDOW=LNORESP
'A PROBLEM WAS DETECTED ON THE BIND PLAN. TRY LATER'
DBC706  'COLLECTION NOT DELETED'      .ALARM=YES .WINDOW=LNORESP
'A PROBLEM WAS DETECTED ON THE BIND PLAN. TRY LATER'
DBC709  'COLLECTION NOT EXIST'        .ALARM=YES .WINDOW=LNORESP
'THE COLLECCTION TO DELETE IS NOT PRESENT IN PLAN/PACKLIST'

```

```
DBC710      'PLAN WHITOUT COLLECTIONS'  .ALARM=YES .WINDOW=LNORESP
'THE PLAN/PACKLIST NOT HAVE ANY COLLECTIONS TO DELETE'
```

Carlos Osorio Montoya
Database and MQSeries Administrator (Peru)

© Xephon 2005

DSNWSPM – a useful but undocumented stored procedure

DSNWSPM is a nice, useful, but undocumented stored procedure—at least for DB2 V7. DSNWSPM returns information about resources that are used by a thread. You have to call DSNWSPM twice. The first call sets the initial values. The second call returns performance information. Between the two calls you have to place your SQL that you want to measure.

DSNWSPM returns CPU time, lock wait time, number of getpages, and number of I/Os. You have to consider the second call's values as net cumulative changes. When you run the REXX you will see the performance information at the end of the output:

```
-----
PERFORMANCE INFORMATION (Second Call)
-----
CPU time (Ext.format)           : 00:00:00.005536
Lock/latch cont.wait time (Ext.format) : 00:00:00.000053
Time as integer in hundredths of a sec : 0
Lock/latch cont.wait time (Integer)   : 0
Number of getpages (Integer)          : 63
Number of read I/O operations (Integer) : 1
Number of write I/O operations (Integer) : 0
-----
```

To run DSNWSPM you have to:

1 Create and grant the DSNWSPM stored procedure:

```
CREATE PROCEDURE SYSPROC.DSNWSPM
(
```

```

INOUT  INIT_CPU_TIME      CHAR(8),
INOUT  INIT_LOCK_TIME    CHAR(8),
OUT    CPU_TIME_EXT      CHAR(15),
OUT    LOCK_WAIT_TIME_EXT CHAR(15),
OUT    CPU_TIME          INTEGER,
OUT    LOCK_WAIT_TIME    INTEGER,
OUT    GETPAGE_COUNT     INTEGER,
OUT    READ_IO_COUNT     INTEGER,
OUT    WRITE_IO_COUNT    INTEGER
)
EXTERNAL NAME DSNWSPM
LANGUAGE ASSEMBLE
NO WLM ENVIRONMENT
COLLID DSNWSPM
;
GRANT EXECUTE ON PROCEDURE SYSPROC.DSNWSPM TO PUBLIC;

```

2 Transfer the REXX to your TSO environment:

```

/* REXX ***** */
/* This rexx calls DSNWSPM stored procedure. */
/* */
/* DSNWSPM is called twice. */
/* First call is for setting the initial values. */
/* Second call returns performance information. */
/* Between the two calls you have to place your SQLs which you */
/* want to measure. */
/* SSID must be changed to your own DB2 member name. */
/* Serdar Sabri OZKUBULAY */
/* ***** */
/* Change the SSID as your own DB2 member name ***** */
      SSID='DBØP'
/* Change the SSID as your own DB2 member name ***** */
£SUBCOM DSNREXX£
IF RC THEN
  S_RC = RXSUBCOM('ADD','DSNREXX','DSNREXX')
ADDRESS DSNREXX £CONNECT£ SSID
IF SQLCODE <> Ø THEN CALL SQLCA
/* Initial Values ***** */
INIT_CPU_TIME      = 'FFFFFFFFFFFFFFFF'X
INIT_LOCK_TIME     = 'ØØØØØØØØØØØØØØØØ'X
CPU_TIME_EXT       = ' '
LOCK_WAIT_TIME_EXT = ' '
CPU_TIME           = Ø
LOCK_WAIT_TIME     = Ø
GETPAGE_COUNT      = Ø
READ_IO_COUNT      = Ø
WRITE_IO_COUNT     = Ø
/* Initial Values ***** */
/* First CALL of DSNWSPM ***** */

```

```

ADDRESS DSNREXX £EXECSQL£ ,
£CALL£ DSNWSPM £(:INIT_CPU_TIME      ,£,
                  £ :INIT_LOCK_TIME    ,£,
                  £ :CPU_TIME_EXT      ,£,
                  £ :LOCK_WAIT_TIME_EXT,£,
                  £ :CPU_TIME          ,£,
                  £ :LOCK_WAIT_TIME    ,£,
                  £ :GETPAGE_COUNT     ,£,
                  £ :READ_IO_COUNT     ,£,
                  £ :WRITE_IO_COUNT    )£

IF SQLCODE <> 0 THEN CALL SQLCA
SAY '-----'
SAY ' PERFORMANCE INFORMATION (First Call) '
SAY '-----'
SAY 'Call type or begin CPU Time           : ' INIT_CPU_TIME
SAY 'Begin lock/latch time                 : ' INIT_LOCK_TIME
SAY 'CPU time (Ext.format)                 : ' CPU_TIME_EXT
SAY 'Lock/latch cont.wait time (Ext.format) : ' LOCK_WAIT_TIME_EXT
SAY 'Time as integer in hundredths of a sec : ' CPU_TIME
SAY 'Lock/latch cont.wait time (Integer)    : ' LOCK_WAIT_TIME
SAY 'Number of getpages (Integer)           : ' GETPAGE_COUNT
SAY 'Number of read I/O operations (Integer) : ' READ_IO_COUNT
SAY 'Number of write I/O operations (Integer) : ' WRITE_IO_COUNT
SAY '-----'
/* First CALL of DSNWSPM ***** */
/* SQLs which you want to measure ***** */
ADDRESS DSNREXX £EXECSQL SET :TODAY = CURRENT DATE£
SAY 'TODAY IS' TODAY
SQL_STMT = ,
£ SELECT COUNT(*) FROM SYSIBM.SYSTABLESPACE £
ADDRESS DSNREXX £ EXECSQL DECLARE C1 CURSOR FOR S1 £
ADDRESS DSNREXX £ EXECSQL PREPARE S1 FROM :SQL_STMT £
ADDRESS DSNREXX £ EXECSQL OPEN C1 £
DO FOREVER
ADDRESS DSNREXX £ EXECSQL FETCH C1 INTO :NUMTS £
IF SQLCODE <> 0 THEN LEAVE
SAY 'I HAVE' NUMTS 'TABLESPACE'
END
/* SQLs which you want to measure ***** */

/* Second CALL of DSNWSPM ***** */
ADDRESS DSNREXX £EXECSQL£ ,
£CALL£ DSNWSPM £(:INIT_CPU_TIME      ,£,
                  £ :INIT_LOCK_TIME    ,£,
                  £ :CPU_TIME_EXT      ,£,
                  £ :LOCK_WAIT_TIME_EXT,£,
                  £ :CPU_TIME          ,£,
                  £ :LOCK_WAIT_TIME    ,£,
                  £ :GETPAGE_COUNT     ,£,
                  £ :READ_IO_COUNT     ,£,

```

```

£ :WRITE_IO_COUNT      )£

IF SQLCODE <> 0 THEN CALL SQLCA
/* Second CALL of DSNWSPM ***** */
SAY '-----'
SAY '  PERFORMANCE INFORMATION (Second Call)  '
SAY '-----'
SAY 'CPU time (Ext.format)                   : ' CPU_TIME_EXT
SAY 'Lock/latch cont.wait time (Ext.format)  : ' LOCK_WAIT_TIME_EXT
SAY 'Time as integer in hundredths of a sec  : ' CPU_TIME
SAY 'Lock/latch cont.wait time (Integer)     : ' LOCK_WAIT_TIME
SAY 'Number of getpages (Integer)            : ' GETPAGE_COUNT
SAY 'Number of read I/O operations (Integer) : ' READ_IO_COUNT
SAY 'Number of write I/O operations (Integer): ' WRITE_IO_COUNT
SAY '-----'
EXIT
SQLCA:
  SAY 'SQLCODE =' SQLCODE
  SAY 'SQLERRMC =' SQLERRMC
  EXIT

```

DSNWSPM DDL

```

CREATE PROCEDURE SYSPROC.DSNWSPM
(
  INOUT  INIT_CPU_TIME      CHAR(8),
  INOUT  INIT_LOCK_TIME     CHAR(8),
  OUT    CPU_TIME_EXT       CHAR(15),
  OUT    LOCK_WAIT_TIME_EXT CHAR(15),
  OUT    CPU_TIME           INTEGER,
  OUT    LOCK_WAIT_TIME     INTEGER,
  OUT    GETPAGE_COUNT      INTEGER,
  OUT    READ_IO_COUNT      INTEGER,
  OUT    WRITE_IO_COUNT     INTEGER
)
EXTERNAL NAME DSNWSPM
LANGUAGE ASSEMBLE
NO WLM ENVIRONMENT
COLLID DSNWSPM
;
GRANT EXECUTE ON PROCEDURE SYSPROC.DSNWSPM TO PUBLIC;

```

Serdar Sabri Özkubulay
DB2 Systems Programmer
Akbank (Turkey)

© Xephon 2005

Move data across DB2 subsystems to another LPAR

Moving data across different databases and different platforms has been a common task in IT environments. By moving data we mainly refer to the task of copying data from a source system to a target system. There are many useful utilities for moving data with DB2 UDB for z/OS:

- DSN1COPY is a stand-alone utility for copying table spaces from the DB2 VSAM datasets or from the DB2 image copy datasets.
- DSNTIAUL is a dynamic sample SQL unload program. This program unloads the rows in a form that is compatible with and generates control statements for the Load utility.
- The DB2 Reorg utility with the REORG UNLOAD EXTERNAL option unloads data in a format usable by the Load utility.
- The Unload utility unloads DB2 data to sequential datasets. The source can be DB2 table spaces or DB2 Image Copy datasets. With Unload, you can also select columns by using the field specification list. The output records written by the Unload utility are compatible as input to the Load utility; as a result, you can reload the original table or different tables.
- The DB2 Load utility populates DB2 tables with data from a sequential dataset.
- The DB2 Cross Loader option allows you to transfer data from one location for loading a DB2 table to another location within a single utility job. The data is read from the source location with a dynamic SQL statement (EXEC SQL utility) and loaded into a table at the target location by the Load utility.

In my scenario, I move data from one table residing in one DB2 subsystem in LPAR MVSLJ to one target table residing in another DB2 subsystem in LPAR MVSMB. The prerequisite is that you need to have at least a select privilege on the source table and an update privilege on the target table on the other subsystem. This uses REXX procedures ULLR and ULLMR. The ULLMR procedure uses the Unload and Load utility and generates a single JCL job. The job should be submitted at the source site, LPAR MVSLJ.

The single JCL job has four steps at the source site, LPAR MVSLJ, and three steps at the target site, LPAR MVSMB.

Step1 deletes working datasets for output data and control statements for the Load utility (SYSREC and SYSPUNCH).

Step 2 unloads the rows and generates control statements for the Load utility.

Step 3 edits the control statements (PUNCH dataset) with additional options for the Load utility.

Step 4 generates a new JCL job, and submits it (option SYSOUT=(A,INTRDR)), using PL/I program PXMIT.

This new JCL job is submitted automatically at source site LPAR MVSLJ, but statement /*XMIT MVSMB transmits the input stream records from LPAR MVSLJ to LPAR MVSMB. The JCL job in LPAR MVSMD is also submitted automatically.

The first step (LOAD1) in LPAR MVSMB loads data in the target table. The SYSREC and SYSIN datasets should be accessible from both LPARs. The next steps (CHECKP and REPAIR) are optional. The step CHECKP searches, using PL/I program PTSCP, the table spaces in CHECK PENDING, and step REPAIR forces these table spaces to have READ/WRITE status.

The example JCL is for a single job:

```
/*-----  
/* STEP 1: DELETE WORKING DATASETS  
/*-----
```

```

//STEP1      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DELETE DB2MB.UNLOAD.TABLE.*
  DELETE DB2MB.PUNCH.TABLE.*
  SET MAXCC=0
//*-
//* STEP 2: UNLOAD NADI.TAB01 LPAR: MVSLJ
//*-
//UNL01      EXEC DSNUPROC,SYSTEM=DSNY,
//            UID='NADI.UNLOAD1',UTPROC='',COND=(4,LT)
//SYSREC DD DSN=DB2MB.UNLOAD.TABLE.TB1,
//            STORCLAS=NONSMS,
//            UNIT=3390,VOL=SER=MVSD1,
//            DISP=(NEW,CATLG,CATLG),
//            SPACE=(TRK,(169,25),RLSE)
//SYSPUNCH DD DSN=DB2MB.PUNCH.TABLE.TB1,
//            STORCLAS=NONSMS,
//            UNIT=3390,VOL=SER=MVSD1,
//            DISP=(NEW,CATLG,CATLG),
//            SPACE=(TRK,(1,1),RLSE)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  UNLOAD TABLESPACE DSNDB04.TSTAB01
  EBCDIC NOSUBS NOPAD S390
  SHRLEVEL REFERENCE
  FROM TABLE NADI.TAB01
//*
//*-
//* STEP 3: EDIT PUNCH DATASET
//*-
//EDIT      EXEC PGM=IKJEFT01,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
  EDIT 'DB2MB.PUNCH.TABLE.TB1' DATA NONUM
  TOP
  CHANGE * 99999 'RESUME YES' 'REPLACE' ALL
  TOP
  CHANGE * 99999 'LOG NO' 'LOG NO NOCOPYPEND ENFORCE NO' ALL
  TOP
  FIND 'LOAD DATA'
  INSERT STATISTICS TABLE(ALL) INDEX(ALL)
  INSERT REPORT YES UPDATE(ALL)
  END SAVE
//*
//*-
//* STEP 4: TRANSMIT JCL IN ANOTHER LPAR
//*-
//JCLGEN      EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(7,LT),REGION=4096K
//STEPLIB DD DSN=CEE.SCEERUN,DISP=SHR

```

```

//          DD      DSN=DSN710.SDSNLOAD,DISP=SHR
//          DD      DSN=SKUPNI.BATCH.LOADLIB.Y,DISP=SHR
//SYSTSPRT DD      SYSOUT=X
//SYSPRINT DD      SYSOUT=X
//LISTA    DD      SYSOUT=(A,INTRDR),DCB=(RECFM=F,BLKSIZE=80)
//SYSUDUMP DD      DUMMY
//SYSTSIN   DD      *
DSN SYSTEM(DSNY)
RUN PROGRAM(PXMIT) PLAN(PXMIT)
END
/*-----
/*---- MOVE ON LPAR: MVSMB
/*---- LOAD NADI.TAB01
/*-----
//IN          DD      *
#//NADIX JOB (1200-1205-00),CLASS=A,
#//          MSGCLASS=X,NOTIFY=NADI,
#//          MSGLEVEL=(1,1),USER=,REGION=4M
#/*XMIT MVSMB
#//NADIX JOB (1200-1205-00),CLASS=A,
#//          MSGCLASS=X,NOTIFY=NADI,
#//          MSGLEVEL=(1,1),USER=SYSADM,REGION=4M
#//LOAD1     EXEC DSNUPROC,SYSTEM=DSNN,
#//          UID='NADI.LOAD1',UTPROC='',COND=(4,LT)
#//SYSREC    DD      DSN=DB2MB.UNLOAD.TABLE.TB1,DISP=SHR
#//SYSUT1    DD      DSN=SYSADM.SYSUT1.TABLE.TB1,
#//          DISP=(MOD,DELETE,CATLG),
#//          UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE)
#//SYSERR    DD      DSN=SYSADM.SYSERR.TABLE.TB1,
#//          DISP=(MOD,DELETE,CATLG),
#//          UNIT=SYSDA,SPACE=(CYL,(10,10),RLSE)
#//SYSMAP    DD      DSN=SYSADM.SYSMAP.TABLE.TB1,
#//          DISP=(MOD,DELETE,CATLG),
#//          UNIT=SYSDA,SPACE=(CYL,(10,10),RLSE)
#//SORTWK01 DD      UNIT=SYSDA,SPACE=(CYL,(100,10),,,ROUND)
#//SORTWK02 DD      UNIT=SYSDA,SPACE=(CYL,(100,10),,,ROUND)
#//SORTOUT   DD      UNIT=SYSDA,SPACE=(CYL,(100,10),,,ROUND)
#//SYSPRINT DD      SYSOUT=*
#//SYSIN     DD      DSN=DB2MB.PUNCH.TABLE.TB1,DISP=SHR
#//*
#//CHECKP    EXEC    PGM=IKJEFT01,DYNAMNBR=20,COND=(7,LT),REGION=4096K
#//STEPLIB   DD      DSN=CEE.SCEERUN.ZOS,DISP=SHR
#//          DD      DSN=DSN710.SDSNLOAD,DISP=SHR
#//          DD      DSN=SKUPNI.BATCH.LOADLIB,DISP=SHR
#//SYSTSPRT DD      SYSOUT=X
#//SYSPRINT DD      SYSOUT=X
#//LISTA     DD      DSN=DB2MB.CHECK.TS,DISP=OLD
#//SYSUDUMP DD      DUMMY
#//SYSTSIN   DD      *
# DSN SYSTEM(DSNN)

```

```
# RUN PROGRAM(PTSCP)  PLAN(PTSCP)
# END
#/*
#/*----- CHECK PENDING -----
#//REPAIR  EXEC DSNUPROC,SYSTEM=DSNN,COND=(4,LT)
#//STEPLIB  DD DSN=DSN710.SDSNLOAD,DISP=SHR
#//SYSIN    DD DSN=DB2MB.CHECK.TS,DISP=SHR
#/*
#/*
```

ULLR: REXX DRIVER

```
/* REXX */
/* trace r */
zpfctl = 'OFF'
address ispeexec 'vput (zpfctl) profile'
CUR='F1'
address ispeexec "display panel(pulr01) cursor("CUR")"
do while rc=0
    if kurs='F1' | kurs='FIELD1' then do
        Call ullmr 'C'
        CUR='F1'
    end
    if kurs='F2' | kurs='FIELD2' then do
        Call ullmr 'T'
        CUR='F2'
    end
    if kurs='F3' | kurs='FIELD3' then do
        Call ullmr 'L'
        CUR='F3'
    end
    address ispeexec "display panel(pulr01) cursor("CUR")"
end
exit
```

ULLMR: REXX PROCEDURE

```
/* REXX */
/* Move data across DB2 subsystems in another LPAR */
/* trace r */
arg poz
if poz='C' then title='_Unload - Load Utility - Classic Method_'
if poz='T' then title='_Unload - Load Utility - Template Method_'
if poz='L' then title='_Unload - Load Utility - Listdef Method_'
zpfctl = 'OFF'
Y=MSG("OFF")
address ispeexec 'vput (zpfctl) profile'
Call Create_messg
```

```

sysr='DSNN'
TOP:
address ispeexec "display panel(PULR02)"
if rc=8 then exit
/* Check input parameters */
if crecm=' ' & tabcm=' ' & tsncm=' ' & dbncm=' ' then do
    zedsmsg = "Enter Catalog field"
    zedlmsg = "At least one Catalog search field must be entered"
    address ispeexec "setmsg msg(isrz001)"
    signal top
end
HCR='';HTB='';HTS='';HDB='';HCO='';HRU='';HCC='';HVC=''
messg = "Accessing db2 system "db2""
messg = time() || " " || messg
Call Send_messg
messg = 'Select systables information'
messg = time() || " " || messg
Call Send_messg
messg = 'Select systablespace information'
messg = time() || " " || messg
Call Send_messg
messg = 'Select systablepart information'
messg = time() || " " || messg
Call Send_messg
address ispeexec 'tbcreate "tlist",
                names(NTB,HDB,HTS,HCR,HTB,HCO,HRU,HCC,DETAIL,pri,sec,icdsn)'
/* DSNREXX Language Support */
Address TSO "SUBCOM DSNREXX"
IF RC THEN
S_RC = RXSUBCOM(ADD,DSNREXX,DSNREXX)
SSID = db2
ADDRESS DSNREXX "CONNECT" SSID
SQLSTMT= "SELECT STRIP(T.DBNAME),STRIP(T.TSNAME),STRIP(T.CREATOR), ",
"          STRIP(T.NAME),T.CARD,DATE(T.STATSTIME),T.COLCOUNT, ",
"          STRIP(P.VCATNAME) CONCAT '.DSNDBD.' CONCAT ",
"          STRIP(P.DBNAME) CONCAT '.' CONCAT STRIP(P.TSNAME) ",
"          CONCAT '.I0001.A' CONCAT ",
"          CASE P.PARTITION ",
"              WHEN 0 THEN '001' ",
"              ELSE SUBSTR(DIGITS(P.PARTITION),3) ",
"          END ",
"FROM SYSIBM.SYSTABLES T, ",
"     SYSIBM.SYSTABLESPACE S, ",
"     SYSIBM.SYSTABLEPART P ",
"WHERE T.CREATOR LIKE '"crecm"%' ",
" AND T.NAME like '"tabcm"%' ",
" AND T.TSNAME LIKE '"tsncm"%' ",
" AND T.DBNAME LIKE '"dbncm"%' ",
" AND T.TYPE = 'T' ",
" AND S.DBNAME = T.DBNAME ",

```

```

" AND S.NAME = T.TSNAME
" AND P.DBNAME = S.DBNAME
" AND P.TSNAME = S.NAME
" ORDER BY T.DBNAME, T.TSNAME ,T.CREATOR, T.NAME
" WITH UR
Address DSNREXX "EXECSQL DECLARE C1 CURSOR FOR S1"
Address DSNREXX 'EXECSQL PREPARE S1 FROM :SQLSTMT'
Address DSNREXX "EXECSQL OPEN C1"
Address DSNREXX "EXECSQL FETCH C1 INTO ",
":HDB,:HTS,:HCR,:HTB,:HCO,:HRU,:HCC,:HVC"
tb=''; cr=''
tb=htb; cr=hcr
ntb=1; pri=0; sec=0
indtb=0
if si='IC' & sqlcode=0 then Call Select_image
do while(sqlcode=0)
    indtb=1
    DETAIL=right(ntb,4)||' '||left(HDB,8)||' '||left(HTS,8),
        left(HCR,8)||' '||HTB
    Call Pri_sec
    pri=pri+prip
    sec=sec+secp
    address ispexec 'tbadd "tlist"'
    Address DSNREXX,
    "EXECSQL FETCH C1 INTO :HDB,:HTS,:HCR,:HTB,:HCO,:HRU,:HCC,:HVC"
    if si='IC' then do
        if cr=hcr & tb=htb
            then NOP
            else Call Select_image
        end
    if sqlcode=0 then do
        if cr=hcr & tb=htb then do
            address ispexec 'tbbottom "tlist"'
            address ispexec 'tbdelete "tlist"'
        end
        else do
            tb=htb; cr=hcr
            ntb=ntb+1
            pri=0; sec=0
        end
    end
end
end
Address DSNREXX "EXECSQL CLOSE C1"
/* Data not found */
if indtb=0 & si='TB' then Call Check_data
/* Multiple rows result */
if ntb > 1 & wse = 'YES' then do
    address ispexec 'tbend "tlist"'
    address ispexec "tbend "messdb"
    Call Create_messg

```

```

zedsmg='Multiple rows'
zedlmsg='Selection YES supports only one table. Change criteria'
address ispexec "setmsg msg(isrz001)"
Signal Top
end
/* Data not found */
if indtb=0 then Call Check_data
address ispexec 'tbttop "tlist"'
address ispexec 'tbdispl "tlist" panel(PULR04)'
Call Start
/* Load data */
if loa='YES' then do
    address ispexec "display panel(PULR03)"
    Call Start
end
/* Confirmation to load data in the same table */
do while zcmd<>'Y' & loa='YES' & db2=sysr
    cur='ZCMD'
    zedsmg='Warning! See PF1 message'
    zedl1 ='Confirm load data in the same table '
    zedl2 ='with Y-YES or N-NO in Command line.'
    zedlmsg=zedl1||zedl2
    address ispexec "setmsg msg(isrz001)"
    address ispexec "display panel(PULR03) cursor("cur")"
    if rc=8 then zcmd='N'
    if zcmd='N' then do
        address ispexec 'tbend "tlist"'
        address ispexec "tbend "messdb""
        Call Create_messg
        zcmd=''
        Signal Top
    end
end
/* Selection condition for unload utility */
if wse='YES' & loa='NO' then do
    address ispexec "display panel(PULR03)"
    Call Start
end
if wse='YES' then do
    if colu1>' ' then c1=1
    if colu2>' ' then c2=1
    if item1>' ' then i1=1
    if item2>' ' then i1=2
    if item3>' ' then i1=3
    if item4>' ' then i1=4
    if item5>' ' then i1=5
end
spec1=0; spec2=0
unloads=ccsid||' '
if nos='YES' then unloads=unloads||'NOSUBS '

```

```

if npd='YES' then unloads=unloads||'NOPAD '
unloads=unloads||fio||' '
if mer>=0 then unloads=unloads||'MAXERR '||mer
unloads=strip(unloads)
if unloads>' ' then spec1=1
if shr>' ' then do
    if shr='REFERENCE' then shrlevel='SHRLEVEL REFERENCE'
    else shrlevel='SHRLEVEL CHANGE ISOLATION '||ur
    spec2=1
end
messg = 'Building the JCL statements'
messg = time() || " " || messg
Call Send_messg
/* JCL Skeleton Unload - Load Utility      */
date=date()
time=time(c)
user=userid()
tempfile=userid()||'.UNLOAD.LOAD.JCL'
address tso
"delete '"tempfile'"
"free dsname('"tempfile'"
"free ddname(ispfile)"
"free atttlist(formfile)"
"attrib formfile blksize(800) lrecl(80) recfm(f b) dsorg(ps)"
"alloc ddname(ispfile) dsname('"tempfile'"
    "new using (formfile) unit(3390) space(1 1) cylinders"
address ispexec
"ftopen"
"ftincl ULUSR"
"ftclose"
zedsmgs = "JCL shown"
zedlmsg = "JCL Unload Utility shown"
"setmsg msg(isrz001)"
"edit dataset('"tempfile'"
address ispexec 'tbend "tlist"
exit
/* Primary and secondary allocation      */
Pri_sec:
prip=0; secp=0
dsn = "('hvc'"
X=OUTTRAP('var.')
address tso "listc" entries dsn allocation
X=OUTTRAP('OFF')
hurba = word(translate(var.9,' ','-'),7)
if hurba < trunc(737280/15,0) then do
    prip=1
    secp=1
end
else do
    prip=trunc((hurba/(737280/15)+1),0)

```

```

        secp=max(trunc(prip*0.15,0),1)
    end
Return
/* select full image copies                                */
Select_image:
    indc4=0
    address ispexec 'tbcreate "iclist",
        names(sel,icdb,icts,icd,ict,ity,idsn,medi,dsname)'
    SQLSTMT= "SELECT DISTINCT STRIP(DBNAME),STRIP(TSNAME),DSNUM,
    "          ICTYPE,DEVTYPE,DATE(TIMESTAMP),TIME(TIMESTAMP),DSNAME
    "          ",
    "FROM SYSIBM.SYSCOPY
    "WHERE DBNAME = '"hdb"'
    "  AND TSNAME = '"hts"'
    "  AND ICTYPE = 'F'
    "ORDER BY 1, 2, DSNUM, 6 DESC, 7
    "WITH UR
    Address DSNREXX "EXECSQL DECLARE C4 CURSOR FOR S4"
    Address DSNREXX "EXECSQL PREPARE S4 FROM :SQLSTMT"
    Address DSNREXX "EXECSQL OPEN C4"
    Address DSNREXX "EXECSQL FETCH C4 INTO ",
        ":icdb,:icts,:ity,:idsn,:medi,:icd,:ict,:dsname"
    dbts=icdb||'.'||icts||' '||hcr||'.'||htb
    sel=''
    indic=0
    do while(sqlcode=0)
        indic=1
        address ispexec 'tbadd "iclist"'
        Address DSNREXX "EXECSQL FETCH C4 INTO ",
            ":icdb,:icts,:ity,:idsn,:medi,:icd,:ict,:dsname"
    end
    address ispexec 'tbtop "iclist"'
    if indic=1
    then address ispexec 'tbdispl "iclist" panel(PULRIC)'
    else do
        zedsmg='No Image copy found'
        zedlmsg='No Image copy found for '||dbts
        address ispexec "setmsg msg(isrz001)"
    end
    if sel='' | rc=8 then do
        indc4=1
        address ispexec 'tbend "iclist"'
        rc=8
        Call start
    end
    icdsn=dsname
    address ispexec 'tbend "iclist"'
    Address DSNREXX "EXECSQL CLOSE C4"
Return
/* Check control on F3                                    */
Start:

```

```

if rc=8 then do
  if indc2=1 then do
    Address DSNREXX "EXECSQL CLOSE C2"
    indc2=0
  end
  if indc3=1 then do
    Address DSNREXX "EXECSQL CLOSE C3"
    indc3=0
  end
  if indc4=1 then do
    Address DSNREXX "EXECSQL CLOSE C1"
    Address DSNREXX "EXECSQL CLOSE C4"
    indc4=0
  end
  address ispexec 'tbend "tlist"'
  address ispexec "tbend "messdb""
  Call Create_messg
  Signal Top
end
Return
/* Check control on input data */
Check_data:
  address ispexec 'tbend "tlist"'
  address ispexec "tbend "messdb""
  Call Create_messg
  zedsmsg='Data not found'
  zedlmsg='Data not found. Change DB2 catalog search criteria'
  address ispexec "setmsg msg(isrz001)"
  Signal Top
Return
Create_messg:
  messg = "s"||userid()
  address ispexec "tbcreate "messdb" names(messg) write replace"
Return
Send_messg:
  address ispexec "tbadd " messdb
  address ispexec "control display lock "
  address ispexec "addpop row(13) column(6)"
  address ispexec "tbdispl "messdb" panel(punmes)"
  address ispexec rempop
Return

```

PULR01: PANEL

```

)attr default(%+_)
[ type (output) intens(low) color(green) caps(off)
# type (output) intens(low) color(white) caps(off)
_ type (input) intens(low) color(yellow) caps(off) pad('_')
+ type (text) intens(low) color(green)

```



```

# type(text)    intens(high) caps(off) hilite(reverse)
@ type(output) intens(high) caps(off) color(yellow) hilite(reverse)
[ type( input) intens(high) caps(on ) color(green) pad(_)
)Body Expand(//)
!-/-/- @title                                     +!-/-/-
%Command ==>_zcmd
+
+
#PARAMETER #PARAMETER VALUE                                #PROMPT
+
+
+SSID      =>[db2 +                                         DB2 Subsystem
Identifier
+Source    =>[si+ {sid +
!TS+Tsname!TB+Table!IC+Copy
+Creator   =>[crecm +                                       Table Creator
+Name      =>[tabcm +                                       Table Name
+Tsname    =>[tsncm +                                       Tablespace Name
+Dbname    =>[dbncm +                                       Database Name
+CCSID     =>[ccsid +
!E+Ebcdic!A+Ascii!U+Unicode
+Nosubs    =>[nos+                                           !Yes+or!No
+Nopad     =>[npd+                                           !Yes+or!No
+Float     =>[flo +                                           !S390+or!IEEE
+Maxerr    =>[mer +                                           Maximum rows error
number
+Shrlevel  =>[shr +
!C+Change!R+Reference
+Isolation=>[ur+                                           !C+CS!U+UR
+Load      =>[loa+                                           !Yes+or!No
+Tosystem  =>[sysr+                                           To Subsystem Id
+Selection=>[wse+                                           !Yes+or!No
+
$msg
+
# PF3 Return +
)Init
if (&db2 ^= ' ')
    .attr (db2) = 'pad(nulls)'
if (&si ^= ' ')
    .attr (si) = 'pad(nulls)'
if (&crecm ^= ' ')
    .attr (crecm) = 'pad(nulls)'
if (&tabcm ^= ' ')
    .attr (tabcm) = 'pad(nulls)'
if (&tsncm ^= ' ')
    .attr (tsncm) = 'pad(nulls)'
if (&dbncm ^= ' ')
    .attr (dbncm) = 'pad(nulls)'
if (&ccsid ^= ' ')

```

```

        .attr (ccsid) = 'pad(nulls)'
if (&nos ^= ' ')
        .attr (nos) = 'pad(nulls)'
if (&npd ^= ' ')
        .attr (npd) = 'pad(nulls)'
if (&flo ^= ' ')
        .attr (flo) = 'pad(nulls)'
if (&mer ^= ' ')
        .attr (mer) = 'pad(nulls)'
if (&shr ^= ' ')
        .attr (shr) = 'pad(nulls)'
if (&ur ^= ' ')
        .attr (ur) = 'pad(nulls)'
if (&loa ^= ' ')
        .attr (loa) = 'pad(nulls)'
if (&sysr ^= ' ')
        .attr (sysr) = 'pad(nulls)'
if (&wse ^= ' ')
        .attr (wse) = 'pad(nulls)'
&msg='Enter values for the Unload-Load service !'
)Reinit
)Proc
VER(&db2,NONBLANK)
VER(&si LIST TS,TB,IC)
VER(&si,NONBLANK)
&si = TRUNC(&si,' ')
if (&si='TS') &sid = 'Table space'
if (&si='TB') &sid = 'Table'
if (&si='IC') &sid = 'Image copy'
if (&poz = L)
    if (&si = IC)
        &zedsmsg='Image Copy not allowed'
        &zedlmsg='Image Copy is not allowed with LISTDEF option'
        .msg = isrz001
&ccsid = TRANS(TRUNC(&ccsid,1) E,EBCDIC A,ASCII U,UNICODE)
VER(&ccsid LIST EBCDIC,ASCII,UNICODE)
&nos = TRANS(TRUNC(&nos,1) Y,YES N,NO)
VER(&nos LIST YES,NO)
&npd = TRANS(TRUNC(&npd,1) Y,YES N,NO)
VER(&npd LIST YES,NO)
&flo = TRANS(TRUNC(&flo,1) S,S390 I,IEEE )
VER(&flo LIST S390,IEEE)
VER(&mer,NUM)
&shr = TRANS(TRUNC(&shr,1) C,CHANGE R,REFERENCE)
VER(&shr LIST CHANGE,REFERENCE)
IF (&shr = CHANGE)
    &ur = TRANS(TRUNC(&ur,1) C,CS U,UR)
    VER(&ur LIST CS,UR)
    VER(&ur,NONBLANK)
IF (&shr = REFERENCE)

```

```

      &ur = &z
IF (&shr = &z)
      &ur = &z
&loa = TRANS(TRUNC(&loa,1) Y,YES N,NO)
VER(&loa LIST YES,NO)
VER(&loa,NONBLANK)
IF (&loa = YES)
      IF (&sysr = &z)
          &zedsmsg='Enter target Subsystem'
          &zedlmsg='Enter the DB2 subsystem to which you want to load data'
          .msg = isrz001
IF (&loa = NO)
      &sysr = &z
IF (&si = IC ! &si = TS)
      &wse=NO
&wse = TRANS(TRUNC(&wse,1) Y,YES N,NO)
VER(&wse LIST YES,NO)
VER(&wse,NONBLANK)
if (&poz = L)
      if (&wse = YES)
          &zedsmsg='Selection not allowed'
          &zedlmsg='Selection option is not allowed with LISTDEF option'
          .msg = isrz001
VPUT (db2 si crecm tabcm tsncm dbncm ccid) PROFILE
VPUT (nos npd flo mer shr ur loa sysr wse) PROFILE
)End

```

PULR03: PANEL

```

)Attr Default(%+_)
! type(text)      intens(high) caps(on ) color(yellow)
$ type(text)      intens(high) caps(on ) color(red)
$ type(output)    intens(high) caps(off) color(yellow)
] type(output)    intens(high) caps(off) color(green) hilite(reverse)
@ type(output)    intens(high) caps(off) color(yellow) hilite(reverse)
{ type(output)    intens(high) caps(off) color(blue)
# type(text)      intens(high) caps(off) hilite(reverse)
} type(text)      intens(high) caps(off) color(yellow) hilite(reverse)
[ type( input)    intens(high) caps(on ) color(green) pad(_)
)Body Expand(//)
!-/-/- @title      +!-/-/-
%Command ==>_zcmd
+
+
#PARAMETER #PARAMETER VALUE                                #PROMPT
+
+
+Loadmeth =>[meth    +
!REP+REPLACE!RES+RESUME

```

```

+Log      =>[log+      !Yes+or!No
+Runstats =>[rus+      !Yes+or!No
+Selection=>[wse+      !Yes+or!No
$Columns
    $([colu1
        [colu2
$)
$When
    $([item1
        [item2
        [item3
        [item4
        [item5
$)
+
    $msg
+
+
    } PF3 Return +
)Init
    if (&loa='YES')
        &title='_Load - Parameter Entry_'
    if (&loa='NO' & &wse='YES')
        &title='_Search Criteria Panel_'
    if (&loa='NO' & &wse='YES')
        .attr (meth) = 'type(output) color(white)'
        .attr (log)  = 'type(output) color(white)'
        .attr (rus)  = 'type(output) color(white)'
    if (&si='TS' ! &si='IC')
        .attr (wse) = 'type(output) color(white)'
    if (&si='TB' ! &ntb>1)
        .attr (wse) = 'type(output) color(white)'
    if (&meth^= ' ')
        .attr (meth) = 'pad(nulls)'
    if (&log^= ' ')
        .attr (log) = 'pad(nulls)'
    if (&rus ^= ' ')
        .attr (rus) = 'pad(nulls)'
    if (&wse^= ' ')
        .attr (wse) = 'pad(nulls)'
    if (&wse = NO)
        .attr (colu1) = 'type(output) color(white)'
        &colu1=''
        .attr (colu2) = 'type(output) color(white)'
        &colu2=''
        .attr (item1) = 'type(output) color(white)'
        &item1=''
        .attr (item2) = 'type(output) color(white)'
        &item2=''

```

```

        .attr (item3) = 'type(output) color(white)'
        &item3=''
        .attr (item4) = 'type(output) color(white)'
        &item4=''
        .attr (item5) = 'type(output) color(white)'
        &item5=''
if (&colu1^= ' ')
    .attr (colu1) = 'pad(nulls)'
if (&colu2^= ' ')
    .attr (colu2) = 'pad(nulls)'
if (&item1^= ' ')
    .attr (item1) = 'pad(nulls)'
if (&item2^= ' ')
    .attr (item2) = 'pad(nulls)'
if (&item3^= ' ')
    .attr (item3) = 'pad(nulls)'
if (&item4^= ' ')
    .attr (item4) = 'pad(nulls)'
if (&item5^= ' ')
    .attr (item5) = 'pad(nulls)'
    &msg='Enter values for the Load service !'
)Reinit
)Proc
    &meth = TRANS(TRUNC(&meth,3) REP,REPLACE RES,RESUME)
    VER(&meth LIST REPLACE,RESUME)
    VER(&meth,NONBLANK)
    &log = TRANS(TRUNC(&log,1) Y,YES N,NO)
    VER(&log LIST YES,NO)
    VER(&log,NONBLANK)
    &rus = TRANS(TRUNC(&rus,1) Y,YES N,NO)
    VER(&rus LIST YES,NO)
    VER(&rus,NONBLANK)
    &wse = TRANS(TRUNC(&wse,1) Y,YES N,NO)
    VER(&wse LIST YES,NO)
    VER(&wse,NONBLANK)
    IF (&wse = YES)
        .attr (colu1) = 'type(input) color(green)'
        .attr (colu2) = 'type(input) color(green)'
        .attr (item1) = 'type(input) color(green)'
        .attr (item2) = 'type(input) color(green)'
        .attr (item3) = 'type(input) color(green)'
        .attr (item4) = 'type(input) color(green)'
        .attr (item5) = 'type(input) color(green)'
        IF (&item5=&z & &item4=&z & &item3=&z & &item2=&z & &item1=&z &
            &colu2=&z & &colu1=&z)
            &zedsmsg='Enter selection'
            &zedlmsg='Enter selection condition'
            .msg = isrz001
    VPUT (meth log rus wse) PROFILE

```

VPUT (colu1 colu2 item1 item2 item3 item4 item5) PROFILE
)End

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

Bernard Zver (bernard.zver@informatika.si)

DBA

Informatica (Slovenia)

© Xephon 2005

AquaFold has announced Version 4.0 of its Aqua Data Studio software, which allows administrators to visually manage the security of any relational database.

Version 4.0 can manage the security of users, roles, and object permissions. The product contains a number of different customization features that enable users to configure the tool more precisely so as to better fit their application needs, they claim. The updated version also has customization capabilities for auto completion, syntax colouring, and data retrieval.

The company has added a wizard to the product's export tool that can extract and export data from any relational base, including DB2, SQL Server 2005, and MySQL 4.1.

For further information contact:
URL: www.aquafold.com.

* * *

Daffodil Software has announced Version 1.6 of Daffodil Replicator, its open source data replication tool equipped with DB2 support. This version also provides for increased performance and high data availability.

Replicator is an enterprise-wide data replication solution that eliminates the complexity of sharing information across heterogeneous operating systems and databases.

It mirrors business-critical data and generates multiple copies for high data accessibility. Its bi-directional replication capabilities allow remote clients to synchronize with data changes in the corporate database.

The enhancements in V1.6 are the result of mutual efforts by Daffodil Software and the open source community. It is now possible to merge data between DB2 and heterogeneous remote database servers (Oracle, SQL Server,

PostgreSQL, Daffodil DB, and Derby) running in distributed environments.

For further information contact:
URL: www.daffodildb.com/replicator.html.

* * *

Ingrian Networks has announced Release 4.0 of the Ingrian DataSecure Platform, its way of encrypting and securing critical data in applications and databases. DataSecure is a network-available device that can interface at any one of three layers: the Web layer, the application layer, or the database layer. The new release boosts scalability, improves performance, and offers increased administrative controls.

It supports Java, Microsoft and C/C++ applications, and comes complete with enhanced database integration tools for DB2, Oracle, and MS SQL.

For further information contact:
URL: www.ingrian.com/products/4-0.html.

* * *

Several flaws have been discovered in DB2 UDB that can be exploited to cause DoS attacks, to reveal sensitive information, to read and manipulate file content, or to compromise vulnerable systems.

IBM advises DB2 UDB administrators to upgrade all DB2 client, server and Connect gateway instances on all supported platforms to DB2 UDB Version 8.1 FixPak 8 as soon as possible. The only exceptions are DB2 UDB client instances on Version 8.1 FixPak 6a, 6b or 7a, which don't need to move up to the FixPak 8 level.

For further information contact your local IBM representative.

