113

DB2 update

March 2002

In this issue

DB2 Update

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

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

Subscriptions and back-issues
A year's subscription to *DB2 Update*, comprising twelve monthly issues, costs £255.00 in the UK; $380.00 in the USA and Canada; £261.00 in Europe; £267.00 in Australasia and Japan; and £265.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1997 issue, are available separately to subscribers for £22.50 ($33.75) 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.

Editor
Trevor Eddolls

Contributions
When Xephon is given copyright, articles published in *DB2 Update* are paid for at the rate of £170 ($260) per 1000 words and £100 ($160) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £50 ($80) per 100 lines. In addition, there is a flat fee of £30 ($50) per article. 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.

Modified DB2 sign-on exit

IMS-DB2 transactions perform a sign-on when they start, invoking the DB2 sign-on exit, DSN3@SGN. The sign-on was costing us too much time, so here's what we did.

BACKGROUND

We were checking the performance of a new application that had a large number of short IMS-DB2 transactions. We noticed that each of them took around 40msec for DB2 sign-on (which was actually longer than the time for a typical dynamic prepare!).

We were using the sample DSN3@SGN exit from IBM, which has not changed significantly since it was introduced in Version 2.1 of DB2. The vast majority of the exit's run time was spent in RACF checking the 'new' user. The only IMS-DB2 transactions running on this system were those for the new application.

SOLUTION

Since the user authority had already been checked in IMS before each transaction used DB2, we reasoned that it was not necessary to check again. Therefore, the sample sign-on exit was modified. Immediately before 'SECTION 1: DETERMINE THE PRIMARY AUTHORIZATION ID' the following was inserted:

```
*******SECTION Ø: IF THIS IS AN IMS TRANSACTION - NO RACF CHECK *******
*                                                                     *
*    THIS SECTION DOES A BRANCH TO FREEMAIN AND EXIT RC(Ø)            *
*                                                                     *
***********************************************************************
         SPACE
SSGNØØ5  DS    ØH
         CLC   EXPLTYPE,IMSTYPE    IS THIS AN IMS TRANSACTION?
         BE    SSGNØ9Ø             BRANCH AROUND RACF PART
         SPACE 2
```

The exit is invoked for all IMS, CICS, or DDF users, but not for TSO or batch. Our modification bypasses the RACF checking only for IMS, so all DDF (or CICS) users are still checked as normal. See the *DB2*

Administration Guide, Appendix B *Writing exit routines*, for more about this exit.

RESULT

With the modified exit, the sign-on was typically taking 2msec instead of 40msec.

This simple change is very effective, if you can use it.

Ron Brown (Germany) © Xephon 2002

Quick EXPLAIN – REXX version

This is a rewrite of a utility that was published in *DB2 Update* April 2001, Issue 102, *Quick EXPLAIN for DBRM SQLs*. The earlier article used Assembler routines to invoke DB2 to get the rows from the Plan table. This is a rewrite of the same using the DB2 REXX interface. The earlier edit macros have been re-used after suitable modifications. More features have also been added.

The utility can be invoked by marking the SQL to be explained in EDIT or VIEW mode using CC begin and end markers on the line numbers, or Cnnn to indicate the number of lines, to be considered as input to Quick EXPLAIN. (This is very similar to the *Cut* macro.) Then, on the command line, enter:

```
QXP SSID TCREATOR,
```

where SSID is the DB2 subsystem ID, and TCREATOR is the name of the creator of the tables.

The EXPLAIN statement is not necessary because the utility automatically substitutes the query number based on the current timestamp. It is assumed that a PLAN_TABLE exists with TCREATOR as the qualifier and the user has *Select* access to the tables in the query being explained.

Only SELECT queries are supported, and any other kinds of query like UPDATE and DELETE are not considered. Any cursor declarations may be explained by beginning at the *Select* clause and omitting the *Declare* clause.

It uses a temporary dataset to strip the query of host variables and replace it with parameter markers. The temporary dataset is deleted on termination of the invocation.

A new feature is the Timeron value that is reported in the REXX stem variable, SQLERRD.4, after preparing the query. This may be useful when comparing access path cost factors, though it does not necessarily indicate the best path.

Appropriate error messages are flagged for improper usage or for SQL errors. Following proper execution of the EXPLAIN SQL, the results are displayed on a scrollable panel to give the user a quick review of the access path. Pressing *Enter* or *End* will exit the display and bring you back to the query.

This utility also works as a great syntax checker. Note that there should not be any terminating semi-colons at the end of the query being explained because this will cause SQLCODE 98.

Components of the utility are:

- qxprex – REXX utility that performs the main processing.
- qxp1_p – the main panel where the results are displayed.
- qxp2_p – the intermediate panel showing work in progress.
- qxph_p – a help panel / tutorial briefly explaining how to use qxp.
- qxp and qxp2 – edit macros used by the utility.

The edit macros may be included in your SYSPROC dataset. The REXX utility may be in your SYSEXEC. The panels should be copied into a panels library.

XESAMP

```
——————————————— DB2 Explain SQL ———— Row 1 to 8 of 8
COMMAND ===>                                                Scroll ===> P
```

```
                       DB2 SUBSYSTEM DBT3
                  REPORT ON FNTEST.PLAN_TABLE
         DATE: 12/13/Ø1    TIME: ØØ:38       TIMERON: 1156387628

QB/PL/MIX M ACC MAT I INDEX    SRTN SRTC TS  PRE COL  TB TABLE
NO/NO/SEQ T TYP COL O USED     UJOG UJOG LCK FTC EVAL NO NAME
-- -- --- - --- --- - -------- ---- ---- --- --- ---- -- ----------
1  Ø   Ø  Ø      Ø  N          NNNN NNNN  IX           1  XYEMDTTB
1  Ø   Ø  Ø      Ø  N          NNNN NNNN  IX           1  XYEMDTTB
1  1   Ø  Ø  M   Ø  N          NNNN NNNN  IS   L       1  XYCWSXTB
1  1   1  Ø  MX  1  Y DMAUSXX1 NNNN NNNN  IS   S       1  XYCWSXTB
1  1   2  Ø  MX  1  Y DMAUSXX1 NNNN NNNN  IS   S       1  XYCWSXTB
1  1   3  Ø  MU  Ø  N          NNNN NNNN               1  XYCWSXTB
1  2   Ø  1  I   1  N DMCNTYX1 NNNN NNNN  IS           2  XYEPTYTB
1  3   Ø  3      Ø  N          NNNN NNYN               Ø
-- -- --- - --- --- - -------- ---- ---- --- --- ---- -- ----------
```

QXP

```
/*********************************************************/
/* Author: Jaiwant K. Jonathan                           */
/*                                                       */
/*********************************************************/
ISREDIT MACRO (SSID,CREAT,KEYZ) NOPROCESS
/* TRACE Ø  */
/* CONTROL SYMLIST CONLIST LIST MSG                             */
   CONTROL NOSYMLIST NOCONLIST NOLIST NOMSG
   ISPEXEC CONTROL ERRORS RETURN
/* ============================================================= */
/*                  EXPLAIN PLAN                                 */
/* ============================================================= */
/* ------------------------------------------------------------- */
/*  SET MACROS NAME                                              */
/* ------------------------------------------------------------- */
IF &STR(&SSID) = &STR() OR &STR(&CREAT) = &STR() THEN +
   DO
      GOTO DISPANEL
      WRITE 'ERROR: QXP USAGE: QXP SSID CREATOR '
      WRITE 'ERROR:            where SSID IS SUBSYSTEM IDENTIFIER '
      WRITE 'ERROR:              and CREATOR IS TABLE QUALIFIER '
      EXIT CODE(8)
   END
SET SSID  = &SSID
SET CREAT = &CREAT
SET KEYZ  = &KEYZ
SET &MACN = &STR(QXP)
/* ------------------------------------------------------------- */
/*  HELP NEEDED ?                                                */
/* ------------------------------------------------------------- */
```

```
IF  &SSID  = &STR('?') THEN +
    DO
    GOTO DISPANEL
    END
IF &STR(&SSID) = &STR() THEN +
   DO
      WRITE 'USAGE IS QXP SSID CREATOR '
      WRITE 'SSID IS SUBSYSTEM IDENTIFIER '
      WRITE 'CREATOR IS TABLE QUALIFIER '
      EXIT CODE(8)
   END
/* ---------------------------------------------------------------- */
/*  HELP NEEDED ?                                                   */
/* ---------------------------------------------------------------- */
IF  &CREAT = ? THEN +
    DO
    GOTO DISPANEL
    END
/* ---------------------------------------------------------------- */
/*  CORRECT PARAMETERS SPECIFIED ?                                  */
/* ---------------------------------------------------------------- */
IF  &KEYZ NE THEN +
    DO
    GOTO DISPANEL
    END
/* ---------------------------------------------------------------- */
/*  CHECK RANGE                                                     */
/* ---------------------------------------------------------------- */
CHKRANGE: +
    ISREDIT PROCESS RANGE C
    SET &RC1 = &LASTCC
    IF  &RC1 = Ø THEN +
        DO
        GOTO OKRANGE
        END
    ELSE +
        DO
        IF  &RC1 = 4 THEN +
            DO
            GOTO ERRFOUND
            END
        ELSE +
            DO
            GOTO ERRFOUND
            END
        END
        END
    END
/* ---------------------------------------------------------------- */
/*  OK RANGE IS FOUND                                               */
/* ---------------------------------------------------------------- */
OKRANGE: +
    IF  &CREAT = THEN +
        DO
        SET &CREAT = &SYSUID
        END
    ISREDIT (LINECMD) = RANGE_CMD
    ISREDIT (BEGLIN)  = LINENUM .ZFRANGE
    ISREDIT (ENDLIN)  = LINENUM .ZLRANGE
    SET &RO           = &EVAL(&ENDLIN - &BEGLIN + 1)
    ISREDIT (LEFTCOL,RIGHTCOL) = BOUNDS
    SET &LE           = &LEFTCOL
    SET &RI           = &RIGHTCOL
    SET &TEMPNUM =   &SUBSTR(1:2,&NRSTR(&SYSTIME))+
                     &SUBSTR(4:5,&NRSTR(&SYSTIME))+
                     &SUBSTR(7:8,&NRSTR(&SYSTIME))
/*  SET &TEMPMEM = XP&TEMPNUM                    */
    SET &TEMPMEM = &STR(XPTEMPDS)
    SET &WKDSN = &SYSPREF..&SYSUID..&TEMPMEM

    DELETE '&WKDSN'
    FREE DDNAME('&WKDSN')
    ALLOC F(WKDD) NEW UNIT(SYSDA) SPACE(5,1Ø) TRACKS +
     DSORG(PS) BLKSIZE(312Ø) LRECL(8Ø) RECFM(F B) DSNAME('&WKDSN')

    FREE DDNAME(WKDD)

/*                                                              */
/*  ISREDIT (WKDSN) = DATASET                                   */
    ISPEXEC LMINIT DATASET('&WKDSN') DATAID(WKFILE) ENQ(EXCLU)

    ISPEXEC LMOPEN DATAID(&WKFILE) OPTION(OUTPUT)

/*  ISPEXEC LMMFIND DATAID(&WKFILE) MEMBER(&TEMPMEM)            */
/*  IF &LASTCC = Ø THEN +                                       */
/*      DO                                                      */
/*      WRITE Member &TEMPMEM already exists                    */
/*      GOTO CLEANUP                                            */
/*      END                                                     */
    SET &COUNTROW = Ø
    DO  WHILE &COUNTROW < &RO
    ISREDIT (WKLINE) = LINE &EVAL(&BEGLIN + &COUNTROW)
    SET &RECORD = &SUBSTR(&LE:&RI,&NRSTR(&WKLINE))
    ISPEXEC LMPUT DATAID(&WKFILE) MODE(INVAR) +
        DATALOC(RECORD) DATALEN(8Ø) NOBSCAN
    SET &COUNTROW = &COUNTROW + 1
    END
/*  ISPEXEC LMMADD DATAID(&WKFILE) MEMBER(&TEMPMEM)          */
    ISPEXEC LMCLOSE DATAID(&WKFILE)
    ISPEXEC LMFREE DATAID(&WKFILE)
    ISPEXEC CONTROL DISPLAY LOCK
    ISPEXEC LIBDEF ISPPLIB        +
    DATASET ID ('HRDBA.PROD.$$PAJKJ.PANELS')
    ISPEXEC LIBDEF ISPTLIB        +
    DATASET ID ('HRDBA.PROD.&SYSUID..TABLES')
    ISPEXEC LIBDEF ISPTABU        +
    DATASET ID ('HRDBA.PROD.&SYSUID..TABLES')
    ISPEXEC LIBDEF ISPTLIB
    ISPEXEC DISPLAY PANEL(QXP2)

/* ------------------------------------------------------------------ */
/*  SQL COPIED TO PARTITIONED DATASET                                 */
/* ------------------------------------------------------------------ */
    ISPEXEC VPUT (CREAT TEMPNUM) PROFILE
    ISPEXEC EDIT DATASET ('&WKDSN') MACRO(QXP2)
    IF &LASTCC ¬=Ø THEN +
    DO
       WRITE 'ERROR RUNNING MACRO QXP2 '
       GOTO ERRFOUND
    END
    SET &DSNM = '&WKDSN'
/*  ISPEXEC BROWSE DATASET ('&WKDSN')                          */

    IF  &RC2 ¬= Ø THEN +
        DO
        SET ZEDSMSG = &STR(Error !. RC=&RC2)
        SET ZEDLMSG = &STR(SQL error or "C" delimiters set )
        ISPEXEC SETMSG MSG(ISRZØØ1)
        GOTO CLEANUP
        END
    ISPEXEC SELECT CMD(QXPREX &DSNM &TEMPNUM &SSID &CREAT) +
        NEWAPPL(TEMP)

/* ------------------------------------------------------------------ */
/*  CLEAN UP                                                          */
/* ------------------------------------------------------------------ */
CLEANUP: +
    ISPEXEC VERASE (CREAT TEMPNUM) PROFILE
    ISPEXEC LMINIT DATASET('&WKDSN') DATAID(DELFILE) ENQ(EXCLU)
    ISPEXEC LMOPEN DATAID(&DELFILE) OPTION(OUTPUT)
    ISPEXEC LMMDEL DATAID(&DELFILE)
    ISPEXEC LMCLOSE DATAID(&DELFILE)
    ISPEXEC LMFREE DATAID(&DELFILE)
    EXIT CODE (&LASTCC)
/* ------------------------------------------------------------------ */
/*   Routines                                                         */
/* ------------------------------------------------------------------ */
DISPANEL: +
    ISPEXEC DISPLAY PANEL (QXPH)
    EXIT CODE(Ø)
```

```
ERRFOUND: +
    SET ZEDSMSG = &STR(Error found !)
    SET ZEDLMSG = &STR(Error detected ! RC = &RC1 +
                       no process possible !)
    ISPEXEC SETMSG MSG(ISRZØØ1)
    EXIT CODE(12)
```

QXP1_P

```
)ATTR
@  TYPE(OUTPUT) INTENS(HIGH) COLOR(RED) HILITE(REVERSE)
%  TYPE(TEXT)   INTENS(HIGH) COLOR(WHITE)
<  TYPE(TEXT)   INTENS(LOW)
$  TYPE(TEXT)   INTENS(HIGH) COLOR(BLUE)   CAPS(ON)
|  TYPE(TEXT)   INTENS(HIGH) COLOR(TURQ)   CAPS(OFF)
!  TYPE(INPUT)  INTENS(HIGH)               CAPS(ON)  JUST(LEFT)
+  TYPE(OUTPUT) INTENS(HIGH) COLOR(WHITE)
#  TYPE(OUTPUT) INTENS(HIGH)               CAPS(OFF)
)BODY EXPAND(\\)
%-\-\- DB2 Explain SQL  -\-\-
%COMMAND ===>!ZCMD                                \ \%Scroll ===>!AMT <
<
                      $ DB2 SUBSYSTEM &SSID <
                  $REPORT ON &CREAT..PLAN&UDS.TABLE     <
       $DATE:+MYDATE     $TIME:+ZTIME      $TIMERON:+TMRON           <
<
|QB/PL/MIX M ACC MAT I INDEX    SRTN SRTC TS  PRE COL  TB TABLE
|NO/NO/SEQ T TYP COL O USED     UJOG UJOG LCK FTC EVAL NO NAME
|— — — - — — - ———— —— —— — — —— — —————————
)MODEL  CLEAR(A1,A2,A3,A4,A5,A6,A7,B8,A8,A9,B1,B2,B3,B4,B6)
#Z #Z  #Z #Z @Z <#Z #Z#Z       #Z   #Z   #Z    #Z #Z  #Z #Z
)INIT
   .ZVARS = '(A1 A2 A3 A4 A5 A6 A7 B8 A8 A9  +
              B1 B2 B3 B4 B6)'
   &MYDATE = '&ZMONTH/&ZDAY/&ZYEAR'
   &UDS    = '_'
)PROC
   &DGIPFKEY = PFKEY
   IF (&CMD=END)
     &COMMAND = CANCEL
)END
```

QXP2

```
ISREDIT MACRO (KEYZ) NOPROCESS
/* TRACE Ø  */
/* CONTROL SYMLIST CONLIST LIST MSG ASIS                            */
   CONTROL NOSYMLIST NOCONLIST NOLIST ASIS NOMSG
   ISPEXEC CONTROL ERRORS RETURN
/* ================================================================ */
/*                  EXPLAIN PLAN (MACRO 2)                          */
/* ================================================================ */
/* ---------------------------------------------------------------- */
/*  MOVE ALL ROWS 1 BYTE TO RIGHT AND REMOVE COMMENTS               */
/* ---------------------------------------------------------------- */
    ISREDIT (LEFTCOL,RIGHTCOL) = BOUNDS
    SET &CTRROW = Ø
    ISREDIT (MAXROW) = LINENUM .ZL
    DO WHILE &CTRROW < &MAXROW
        SET &CTRROW = &CTRROW + 1
        ISREDIT LABEL &CTRROW = .HERE
        ISREDIT SEEK '–' FIRST &LEFTCOL &RIGHTCOL .HERE .HERE
        IF  &LASTCC = Ø THEN +
            DO
            ISREDIT (LINSEL,COLSEL) = CURSOR
            ISREDIT CHANGE P'=' ' ' .HERE .HERE &COLSEL &RIGHTCOL ALL
            END
        ISREDIT SHIFT ) .HERE 1
    END
/* ---------------------------------------------------------------- */
/*  TRANSFORM UNNEEDED EXEC SQL AND END-EXEC INTO BLKS              */
/* ---------------------------------------------------------------- */
    ISREDIT CHANGE ')' ' )' &LEFTCOL &RIGHTCOL ALL
    ISREDIT SEEK 'INSERT ' FIRST &LEFTCOL &RIGHTCOL
    IF  &LASTCC = 4 THEN +
        DO
        ISREDIT SEEK 'SELECT ' FIRST &LEFTCOL &RIGHTCOL
        IF  &LASTCC = 4 THEN +
            DO
            ISREDIT SEEK 'UPDATE ' FIRST &LEFTCOL &RIGHTCOL
            IF  &LASTCC = 4 THEN +
                DO
                ISREDIT SEEK 'DELETE ' FIRST &LEFTCOL &RIGHTCOL
                IF  &LASTCC ¬= Ø THEN +
                    DO
                    EXIT CODE (&LASTCC)
                    END
                ELSE +
                    DO
                    GOTO DOCHANGE
                    END
                END
            ELSE +
                DO
                GOTO DOCHANGE
                END
            END
        ELSE +
            DO
            GOTO DOCHANGE
            END
        END
    ELSE +
        DO
        GOTO DOCHANGE
        END
    END
DOCHANGE: +
    IF  &LASTCC ¬= Ø THEN +
        DO
        EXIT CODE (&LASTCC)
        END
    ISREDIT (LINSEL,COLSEL) = CURSOR
    ISREDIT LABEL &EVAL(&LINSEL - 1) = .LBLSE
    ISREDIT DELETE .ZFIRST .LBLSE ALL
    ISREDIT LABEL 1 = .LBLSF
    ISREDIT CHANGE P'=' ' ' .LBLSF .LBLSF +
        &LEFTCOL &EVAL(&COLSEL - 1) ALL
    ISREDIT CHANGE 'EXEC ' ' ' &LEFTCOL &RIGHTCOL ALL
    ISREDIT CHANGE ' SQL ' '' &LEFTCOL &RIGHTCOL ALL
    ISREDIT CHANGE 'END-EXEC ' '' &LEFTCOL &RIGHTCOL ALL
    ISREDIT CHANGE 'END-EXEC. ' '' &LEFTCOL &RIGHTCOL ALL
/* ----------------------------------------------------------------- */
/*  CREATE A NEW MEMBER TO STORE ORIGINAL SQL                        */
/* ----------------------------------------------------------------- */
    ISPEXEC VGET (CREAT TEMPNUM) PROFILE
/* ----------------------------------------------------------------- */
/*  ADD THE EXPLAIN STATEMENT ON 2nd LINE                            */
/* ----------------------------------------------------------------- */
    SET &REC1 = &STR(EXPLAIN PLAN SET QUERYNO = &TEMPNUM FOR)
    ISREDIT LINE_BEFORE .ZFIRST  ="&REC1"
/* ----------------------------------------------------------------- */
/*  DECLARE THE CURRENT SQLID                                        */
/* ----------------------------------------------------------------- */
    SET &RECØ = &STR(SET CURRENT SQLID = '&CREAT';)
    ISREDIT LINE_BEFORE .ZFIRST  ="&RECØ"
    ISREDIT RESET
/* ----------------------------------------------------------------- */
/*  SEARCH FOR HOST VARIABLES                                        */
/* ----------------------------------------------------------------- */
    ISREDIT CURSOR = 1 1
    ISREDIT SEEK ':' FIRST
    SET &RC = &LASTCC
    DO  WHILE &RC = Ø
        ISREDIT (LINFR,COLFR) = CURSOR
        ISREDIT LABEL &LINFR = .LBLAA
        ISREDIT SEEK ',' NEXT .LBLAA .LBLAA
        IF &LASTCC = Ø THEN +
        DO
           ISREDIT (LINTO,COLTO) = CURSOR
        END
        ELSE +
        DO
           ISREDIT SEEK ' ' NEXT .LBLAA .LBLAA
           ISREDIT (LINTO,COLTO) = CURSOR
        END
        ISREDIT CHANGE ':' '? ' .LBLAA .LBLAA &COLFR &COLTO ALL
        SET &COLFR = &COLFR + 1
        ISREDIT CHANGE P'=' '' .LBLAA .LBLAA &COLFR &COLTO ALL
        ISREDIT SEEK ':' FIRST
        SET &RC = &LASTCC
    END
/* ----------------------------------------------------------------- */
/*  FINAL END                                                        */
/* ----------------------------------------------------------------- */
    ISREDIT SAVE
    ISREDIT END
    EXIT
```

QXP2_P

```
)ATTR
@  TYPE(OUTPUT) INTENS(HIGH) COLOR(RED)    HILITE(BLINK)
?  TYPE(TEXT)   INTENS(HIGH) COLOR(YELLOW) HILITE(REVERSE)
%  TYPE(TEXT)   INTENS(HIGH)
<  TYPE(TEXT)   INTENS(LOW)
$  TYPE(TEXT)   INTENS(HIGH) COLOR(BLUE)
\  TYPE(TEXT)   INTENS(HIGH) COLOR(TURQ)   CAPS(OFF)
!  TYPE(INPUT)  INTENS(HIGH)               CAPS(ON)  JUST(LEFT)
+  TYPE(OUTPUT) INTENS(HIGH) COLOR(WHITE)
#  TYPE(TEXT)   INTENS(HIGH) COLOR(WHITE)   HILITE(BLINK)
)BODY EXPAND(||)
%-|-|- DB2 Explain SQL-|-|-
%COMMAND ===>!ZCMD                              | |%Scroll ===>!AMT <
<
                      $ DB2 SUBSYSTEM &SSID <
                  $REPORT ON &CREAT..PLAN&UDS.TABLE     <
       $DATE:+MYDATE     $TIME:+ZTIME      $TIMERON:+TMRON       <
<
\QB/PL/MIX M ACC MAT I INDEX   SRTN SRTC     PRE COL  TB          TABLE
\NO/NO/SEQ T TYP COL O USED    UJOG UJOG LCK FTC EVAL NO CREATOR  NAME
--------- - --- --- - ----    ---- ---- --- --- ---- -- -------  -----
<
<
<
<                    ?                                  $
<                    ?  #                            ?  $
<                     ?  #    Execution in progress   ?  $
<                     ?  #        please wait         ?  $
<                     ?  #                            ?  $
<                     ?                                  $
)INIT
    &MYDATE = '&ZMONTH./&ZDAY./&ZYEAR'
    &UDS    = '_'
)PROC
   &DGIPFKEY = .PFKEY
   IF (&CMD=END)
     &COMMAND = CANCEL
)END
```

QXPH_P

```
%Quick Aid ————————— QXP MACRO ——————————— TUTORIAL
%COMMAND ===> _ZCMD
+
+
+The%QXP macro+is used to execute the%DB2 EXPLAIN+function while
+editing SQL code in a member of a PDS with the ISPF Editor.
+The format of the command is %QXP SSID CREATOR+ on the command line
+in combination with the %C+ or %CC+ range delimiters on the line
+command. Parameter SSID is the DB2 subsystem identifier and CREATOR is
+the qualifier for non-qualified tables. It assumes that the SQL is not
+qualified.
+ Prerequisite: CREATOR.PLAN&UDS.TABLE must exist in the subsystem
+ SSID. The example below invokes the macro with DBT2 and DMSYST as
+ the values.
+
%COMMAND ===> QXP DBT2 DMSYST
%
%EDIT —— HRXS.$$PAJKJ.COBOL(DMB1110A)
+*************************** TOP OF DATA ******************************
+000023        INITIALIZE ACCOUNT-ABS.
%CC+024        EXEC-SQL
+000025            DECLARE CURSACC CURSOR FOR
+000026              SELECT INTEG1, FIELD1
+000027              FROM   TVOLUME
+000028              WHERE  INTEG2 = :WK-FIELD2   — AND INTEG3 = 4
%CC+029        END-EXEC.
+000030        EXEC-SQL
)INIT
   &UDS    = '_'
)END
```

Jaiwant K Jonathan
DB2 DBA
QSS Inc (USA)

© Xephon 2002

Issuing SQL statements in DB2 utilities

As of Version 7, the EXEC SQL utility control statement can be used to declare cursors and execute dynamic SQL statements during the execution of a DB2 utility. The EXEC SQL control statement produces a result table when you specify a cursor. The EXEC SQL control statement executes entirely in the EXEC phase of the utility. The EXEC phase can be restarted if necessary.

The EXEC SQL statement requires no additional privileges to execute. However, EXEC SQL adheres to the same authorization rules as must be followed for executing dynamic SQL using EXECUTE IMMEDIATE.

SQL statements can be used only in conjunction with DB2 utilities that allow concurrent SQL access on a table space with the utility. No other databases are affected when issuing the EXEC SQL statement.

USING EXEC SQL

To use EXEC SQL as a utility control statement, simply code a permissible SQL statement after the EXEC SQL keyword. That SQL statement will be run during the utility execution as a separate thread. When the SQL statement is executed, the specified statement string is parsed and checked for errors. If the SQL statement is invalid, it is not executed and the error condition is reported. If the SQL statement is valid, but an error occurs during execution, that error condition is reported. When an error occurs, the utility terminates.

There are two options when using EXEC SQL to supply an SQL statement to a utility. The first option is for non-SELECT dynamic SQL statements where the SQL is used as input to an EXECUTE IMMEDIATE statement. The following SQL statements can be specified in an EXEC SQL statement for processing by a DB2 utility:

- ALTER
- COMMENT ON
- COMMIT

- CREATE
- DELETE
- DROP
- EXPLAIN
- GRANT
- INSERT
- LABEL ON
- LOCK TABLE
- RENAME
- REVOKE
- ROLLBACK
- SET CURRENT DEGREE
- SET CURRENT LOCALE LC_CTYPE
- SET CURRENT OPTIMIZATION HINT
- SET CURRENT PATH
- SET CURRENT PRECISION
- SET CURRENT RULES
- SET CURRENT SQLID
- UPDATE.

The second form of SQL permitted within an EXEC SQL utility control statement is a cursor-driven SELECT statement. To use this option, simply declare a cursor that is not already declared and specify the SELECT statement to be used in conjunction with the cursor. For example:

```
EXEC SQL
DECLARE CSR1 CURSOR FOR
SELECT DEPTNO, DEPTNAME, LOCATION FROM DSN871Ø.DEPT
ENDEXEC
```

This statement declares a cursor named CSR1 that selects three columns from all of the rows in the DEPT sample table.

WHY ISSUE SQL DURING A UTILITY?

Once a DBA learns of this new DB2 capability the next logical question usually is, “Why would I want to do that?”. Well, there are several good reasons to run SQL in conjunction with a utility.

One possible use is for general purpose SQL that needs to be run and would otherwise be issued using DSNTEP2, SPUFI, or QMF. For example, consider the (perhaps unlikely) scenario where you wish to give every employee a 10% raise. You could use the EXEC SQL utility control statement to perform this task as you run the utility by including the following statement:

```
EXEC SQL
UPDATE DSN871Ø.EMP
              SET SALARY = SALARY * 1.1Ø
ENDEXEC
```

Perhaps a more likely scenario would be for DBAs to create the tables required for exception processing in CHECK DATA, or the mapping table and index for running a REORG using SHRLEVEL CHANGE. For example, when running CHECK DATA on the ACT sample table you might include the following DDL in the utility job using EXEC SQL:

```
EXEC SQL
CREATE TABLE EXCPT_ACT LIKE DSN871Ø.ACT
ENDEXEC

EXEC SQL
ALTER TABLE EXCPT_ACT
ADD EXCPT_RID CHAR(4)
ENDEXEC

EXEC SQL
ALTER TABLE EXCPT_ACT
ADD EXCPT_TS TIMESTAMP
ENDEXEC
```

This effectively creates the exception table and adds columns to the table as needed.

Similarly, to create the mapping table for a REORG SHRLEVEL CHANGE, the following DDL can be included in the utility job using EXEC SQL:

```
EXEC SQL
CREATE TABLESPACE XMAPØØØ1
 IN DBNAME
  USING STOGROUP MAPSG
     PRIQTY 52
SECQTY 2Ø
  ERASE NO
  LOCKSIZE PAGE
  BUFFERPOOL BP9
  SEGSIZE 8
  CLOSE YES
  COMPRESS NO
ENDEXEC

EXEC SQL
CREATE TABLE MAP_TABLE_ØØØ1
  (TYPE        CHAR(1) NOT NULL,
   SOURCE_RID  CHAR(5) NOT NULL,
   TARGET_XRID CHAR(9) NOT NULL,
   LRSN        CHAR(6) NOT NULL)
IN DBNAME.XMAPØØØ1
ENDEXEC

EXEC SQL
CREATE UNIQUE INDEX XMAPØØØ1
 ON MAP_TABLE_ØØØ1
 (SOURCE_RID ASC,
  TYPE,
  TARGET_XRID,
  LRSN)
ENDEXEC
```

This effectively creates the table space for the mapping table, the mapping table itself, and the unique index required for the mapping table. Please note that, other than the table space needing to be segmented, the exact parameters specified in this example are not etched in stone, and can be changed to suit your site's needs. Additionally, if desired, following the REORG an additional step could be run to DROP the mapping table objects. This way the mapping table exists only when it is needed – during the online reorganization process – and it does not hang around consuming extra disk space when it is not required.

SUMMARY

DB2's ability to execute SQL statements during a utility job delivers a powerful new capability to the DBA. What used to take multiple steps or jobs, might now be accomplished in a single utility step.

Craig S Mullins
Director, Technology Planning
BMC Software (USA)

Data tool for database management – part 2

This month we conclude the code for the DB2 Data Toolkit.

```
sk.1='//COPY'stp' EXEC COPY,LIB='$hiwork'.'DBname'.'nome'.PUNCH,      '
sk.2='//           DISP='MOD'                                         '
          end
       if stp = maxdd then
sk.3='//* ---------------------------------------------- *            '
       sk.Ø=3
       jobw = FIIEB ; "alloc da('"outdsieb"') f("jobw") mod reuse"
       Call WriteRec
   /*–  Step  delete work areas             –*/
sk.1='  DELETE  '$hiwork'.'DBname'.'nome'                             '
sk.2='  DELETE  '$hiwork'.'DBname'.'nome'.PUNCH                       '
       if stp = maxdd & notrat ¬= #d then
sk.3='//* ---------------------------------------------- *            '
       sk.Ø=3
       jobw = FIDEL ; "alloc da('"outdsdel"') f("jobw") mod reuse"
       Call WriteRec
   /*–  Verify max DD for unload job        –*/
       if stp = maxdd then do ; stp = Ø
   /*–  Jobs number                         –*/
          njob  = njob + 1
          if njob = 99 then do
        say '>>>>>>>>                                                 '
        say '>>>>>>>>         !!!!!!!!  W A R N I N G  !!!!!!!!       '
        say '>>>>>>>>                                                 '
      say '>>>>>>>> More than 99 Unload Jobs have been created. During'
      say '>>>>>>>> output, archiving process will have problems with
        duplicated jobnames !!!!!!'
      say '>>>>>>>> Save Output of unload job manually              '
         say '>>>>>>>>' ; say '' ; end
          testata = yes
          if Full = yes then do
sk.1='//LABØEND  ENDIF                                                '
sk.2='//* ---------------------------------------------- *            '
sk.3='//STRRW'njob1'  EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø,COND=EVEN        '
sk.4='//SYSTSPRT  DD  SYSOUT=*                                        '
sk.5='//SYSTSIN   DD  DISP=SHR,                                       '
sk.6='//          DSN='outdsrep'(STRRW'njob1')                        '
sk.7='//* ------------------------------------------- *               '
sk.8='//* ————    Test RC step Unload      ———— *             '
sk.9='//* ------------------------------------------- *               '
sk.1Ø='//LAB1     IF (RC EQ Ø) THEN                                   '
sk.11='//OK       EXEC   DB2REXX1                                     '
sk.12='//REXXØØ.SYSTSIN  DD  *                                        '
sk.13='  ISPSTART CMD(@DB2RCØØ
'subsys','DBname','Jobname||njob1',OK,@DB2UNØØ)'
sk.14='//LAB1END  ENDIF                                               '
sk.15='//* ------------------------------------------- *              '
sk.16='//LAB2     IF (ABEND OR RC GT Ø) THEN                          '
sk.17='//KO       EXEC   DB2REXX1                                     '
sk.18='//REXXØØ.SYSTSIN  DD  *                                        '
sk.19='  ISPSTART CMD(@DB2RCØØ
'subsys','DBname','Jobname||njob1',KO,@DB2UNØØ)'
sk.2Ø='//LAB2END  ENDIF                                               '
sk.21='//* ------------------------------------------- *              '
             sk.Ø=21
             jobw = FIUNL ;"alloc da('"outdsunl"') f("jobw") mod reuse"
             Call WriteRec ; end
   /*—  Release next iebgener Job           —*/
          njob1 = njob1 + 1 ; njob1 = right(njob1,2,'Ø')
          jnarel = iebn || njob1
          if Full = yes then do
sk.1='//*             Test RC step Iebgener             *             '
sk.2='//* ------------------------------------------- *               '
sk.3='//LAB1     IF (RC EQ Ø) THEN                                    '
sk.4='//OK       EXEC   DB2REXX1                                      '
sk.5='//REXXØØ.SYSTSIN  DD  *                                         '
sk.6='  ISPSTART CMD(@DB2RCØØ 'subsys','DBname','jnamod',OK,@DB2UNØØ) '
sk.7='//LAB1END  ENDIF                                                '
sk.8='//* ------------------------------------------- *               '
sk.9='//LAB2     IF (ABEND OR RC GT Ø) THEN                           '
sk.1Ø='//KO       EXEC   DB2REXX1                                     '
sk.11='//REXXØØ.SYSTSIN  DD  *                                        '
sk.12='  ISPSTART CMD(@DB2RCØØ 'subsys','DBname','jnamod',KO,@DB2UNØØ)'
sk.13='//LAB2END  ENDIF                                               '
sk.14='//* ------------------------------------------- *              '
sk.15='//RELEASE EXEC PGM=IEBGENER                                    '
sk.16='//SYSPRINT  DD  SYSOUT=*                                       '
sk.17='//SYSUT2    DD  SYSOUT=(,INTRDR)                               '
sk.18='//SYSUT1    DD  *,DLM=EE                                       '
sk.19='/*$A '''jnarel'''                                              '
sk.2Ø='EE                                                             '
sk.21='//SYSPRINT  DD  SYSOUT=*                                       '
sk.22='//SYSIN     DD  DUMMY                                          '
sk.23='//* ------------------------------------------- *              '
             sk.Ø=23 ; end
          else do
sk.1='//RELEASE EXEC PGM=IEBGENER                                     '
sk.2='//SYSPRINT  DD  SYSOUT=*                                        '
sk.3='//SYSUT2    DD  SYSOUT=(,INTRDR)                                  '
sk.4='//SYSUT1    DD  *,DLM=EE                                          '
sk.5='/*$A '''jnarel'''                                               '
sk.6='EE                                                            '
sk.7='//SYSPRINT  DD  SYSOUT=*                                        '
sk.8='//SYSIN     DD  DUMMY                                           '
sk.9='//* --------------------------------------------- *             '
             sk.Ø=9 ; end
          jobw = FIIEB ; "alloc da('"outdsieb"') f("jobw") mod reuse"
          Call WriteRec ; end ; end /* DO #d End */
    if Full = yes then do
       jobw = FIUNL ; "alloc da('"outdsunl"') f("jobw") mod reuse"
       if (sysprint.Ø < maxdd) | ,
          (stp > Ø & stp < maxdd)  then do
sk.1='//LABØEND  ENDIF                                                '
sk.2='//* --------------------------------------------- *             '
sk.3='//STRRW'njob1'  EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø,COND=EVEN         '
sk.4='//SYSTSPRT  DD  SYSOUT=*                                        '
sk.5='//SYSTSIN   DD  DISP=SHR,                                       '
sk.6='//          DSN='outdsrep'(STRRW'njob1')                        '
sk.7='//* --------------------------------------------- *             '
sk.8='//* --------    Test RC step Unload      -------- *             '
sk.9='//* --------------------------------------------- *             '
sk.1Ø='//LAB1     IF (RC EQ Ø) THEN                                   '
sk.11='//OK       EXEC   DB2REXX1                                     '
sk.12='//REXXØØ.SYSTSIN  DD  *                                        '
sk.13='  ISPSTART CMD(@DB2RCØØ
'subsys','DBname','Jobname||njob1',OK,@DB2UNØØ)'
sk.14='//LAB1END  ENDIF                                               '
sk.15='//* --------------------------------------------- *            '
sk.16='//LAB2     IF (ABEND OR RC GT Ø) THEN                          '
sk.17='//KO       EXEC   DB2REXX1                                     '
sk.18='//REXXØØ.SYSTSIN  DD  *                                        '
sk.19='  ISPSTART CMD(@DB2RCØØ
'subsys','DBname','Jobname||njob1',KO,@DB2UNØØ)'
sk.2Ø='//LAB2END  ENDIF                                               '
sk.21='//* --------------------------------------------- *            '
sk.22='//'jna'.M JOB ('$accn'),''Job Monitor'',CLASS='$class',        '
sk.23='//      MSGCLASS='$msgcla',USER='$user',REGION='$region',      '
sk.24='//      MSGLEVEL=('$msglvl'),NOTIFY='$notif
sk.25='/*JOBPARM BYTES=999999,LINES=9999                              '
sk.26='//* --------------------------------------------- *            '
sk.27='//* -----      Monitor job activity         ----- *            '
sk.28='//* --------------------------------------------- *            '
sk.29='//DB2PROC  JCLLIB ORDER=('$proclib')                           '
sk.3Ø='//REXXØØ   EXEC   DB2REXX1                                     '
sk.31='//REXXØØ.SYSTSIN  DD  *                                        '
sk.32='  ISPSTART CMD(@DB2UNØ4 'outdssou','Jobname')                  '
sk.33='//* --------------------------------------------- *            '
```

```
          sk.Ø=33 ; Call WriteRec
sk.1='//* -------------------------------------------- *           '
sk.2='//* --------    Test RC step Iebgener    -------- *           '
sk.3='//* -------------------------------------------- *           '
sk.4='//LAB1     IF (RC EQ Ø) THEN                                 '
sk.5='//OK       EXEC   DB2REXX1                                   '
sk.6='//REXXØØ.SYSTSIN  DD  *                                      '
sk.7='  ISPSTART CMD(@DB2RCØØ 'subsys','DBname','jnamod',OK,@DB2UNØØ) '
sk.8='//LAB1END  ENDIF                                             '
sk.9='//* -------------------------------------------- *           '
sk.1Ø='//LAB2     IF (ABEND OR RC GT Ø) THEN                       '
sk.11='//KO       EXEC   DB2REXX1                                  '
sk.12='//REXXØØ.SYSTSIN  DD  *                                     '
sk.13='  ISPSTART CMD(@DB2RCØØ 'subsys','DBname','jnamod',KO,@DB2UNØØ)'
sk.14='//LAB2END  ENDIF                                            '
sk.15='//* -------------------------------------------- *          '
          sk.Ø=15
          jobw = FIIEB ; "alloc da('"outdsieb"') f("jobw") mod reuse"
          Call WriteRec ; end
       else do
sk.1='//* -------------------------------------------- *           '
sk.2='//'jna'.M JOB ('$accn'),''Job Monitor'',CLASS='$class',      '
sk.3='//      MSGCLASS='$msgcla',USER='$user',REGION='$region',    '
sk.4='//      MSGLEVEL=('$msglvl'),NOTIFY='$notif
sk.5='/*JOBPARM BYTES=999999,LINES=9999                            '
sk.6='//* -------------------------------------------- *           '
sk.7='//* ------      Monitor job activity        ---- *           '
sk.8='//* -------------------------------------------- *           '
sk.9='//DB2PROC  JCLLIB ORDER=('$proclib')                         '
sk.1Ø='//REXXØØ   EXEC   DB2REXX1                                  '
sk.11='//REXXØØ.SYSTSIN  DD  *                                     '
sk.12='  ISPSTART CMD(@DB2UNØ4 'outdssou','Jobname')               '
sk.13='//* -------------------------------------------- *          '
          sk.Ø=13 ; Call WriteRec ; end
   /*–  Write member  strROyy/strRWyy        –*/
       do #g = 1 to sysrecØØ.Ø
          if #x = Ø then do
             #x = 1
sy.#x ='DSN SYSTEM('Subsys')                                      '
sz.#x ='DSN SYSTEM('Subsys')                                      '
             end ; #x = #x + 1
          $tsname = strip(substr(sysrecØØ.#g,51,8))
sy.#x ='  -START DATABASE('dbname') SPACENAM('$tsname') ACCESS(RO)'
sz.#x ='  -START DATABASE('dbname') SPACENAM('$tsname') ACCESS(RW)'
          #exit = maxdd + 1
          if #x = #exit then do ; call WrstrRO ; call WrstrRW ; end
          end
       if #x > Ø then do ; call WrstrRO ; call WrstrRW ; end
/*implØ1*/
    /*–  Write file header                     –*/
       njobc = njob ; if maxdd = 1 then njobc = njobc - 1
       wappØØ = left(jobname'yy',8,' ')
```

```
sk.1='                   +-----------------------------+       '
sk.2='                   |-     RC jobs verify File     -|       '
sk.3='                   +-----------------------------+       '
sk.4='                                                            '
sk.5=' - Totale job di Unload   ('Jobname'yy) n 'right(njobc,4,'Ø')
        if Autosub = no then
sk.6=' - Totale job di Iebgener ('iebn'yy) n 'right(njobc,4,'Ø')
        else
sk.6='                                                            '
sk.7='                                                            '
       sk.Ø=7;
       jobw = fichk ; "alloc da('"outdschk"') f("jobw") mod reuse"
       Call WriteRec ; end
    if Full = yes then do
sk.1='//* --------------------------------------------- *         '
sk.2='//'$user'M JOB ('$accn'),''Monitor Load Jobs'',CLASS='$class',  '
sk.3='//      MSGCLASS='$msgcla',USER='$user',REGION='$region',      '
sk.4='//      MSGLEVEL=('$msglvl'),NOTIFY='$notif
sk.5='/*JOBPARM BYTES=999999,LINES=9999                           '
sk.6='//* --------------------------------------------- *         '
sk.7='//* ------       Monitor LOAD activity       ---- *         '
sk.8='//* --------------------------------------------- *         '
sk.9='//JOBLIB    DD  DISP=SHR,DSN='$dsnload
sk.1Ø='//DB2PROC   JCLLIB ORDER=('$proclib')                       '
sk.11='//* --------------------------------------------- *        '
sk.12='//REXXØØ   EXEC   DB2REXX1                                 '
sk.13='//REXXØØ.SYSTSIN  DD  *                                    '
sk.14='  ISPSTART CMD(@DB2UNØ6 'Subsys','TJobname','DBname')       '
sk.15='//* --------------------------------------------- *        '
       sk.Ø=15 ; end
    else do
sk.1='//* --------------------------------------------- *         '
       sk.Ø=1 ; end
    jobw = FILOA ; "alloc da('"outdsloa"') f("jobw") mod reuse"
    Call WriteRec
sk.1='//'jna'.O JOB ('$accn'),''Overwrite job'',CLASS='$class',      '
sk.2='//      MSGCLASS='$msgcla',USER='$user',REGION='$region',      '
sk.3='//      MSGLEVEL=('$msglvl'),NOTIFY='$notif
sk.4='/*JOBPARM BYTES=999999,LINES=9999                           '
sk.5='//* --------------------------------------------- *         '
sk.6='//* -------  Overwrite syspunch datasets  ------- *         '
sk.7='//* --------------------------------------------- *         '
sk.8='//DB2PROC  JCLLIB ORDER=('$proclib')                         '
sk.9='//REXXØØ   EXEC   DB2REXX1                                  '
sk.1Ø='//REXXØØ.SYSTSIN  DD  *                                    '
sk.11='  ISPSTART CMD(@DB2UNØ2 'subsys','DBname')                  '
sk.12='//* --------------------------------------------- *        '
    sk.Ø=35;jobw = FIOVR ; "alloc da('"outdsovr"') f("jobw") mod reuse"
    Call WriteRec
    if maxdd = 1 then njob = njob - 1
   /*-  Delete work areas                    -*/
sk.1='  DELETE  'outdsdel
sk.2='  DELETE  'outdsloa
sk.3='  DELETE  'outdsunl
sk.4='  DELETE  'outdsfor
sk.5='  DELETE  'outdsieb
sk.6='  DELETE  'outdsmod
sk.7='  DELETE  'outdsovr
sk.8='  DELETE  'outdsprt
sk.9='  DELETE  'outdsrec
    sk.Ø=9; jobw = FIDEL ; "alloc da('"outdsdel"') f("jobw") mod reuse"
    Call WriteRec
    if Full = yes then do
sk.1='  DELETE  'outdsdmp
sk.2='  DELETE  'outdsrep
sk.3='  DELETE  'outdsddl
sk.4='  DELETE  'outdssou
sk.5='  DELETE  'outdsdrp
sk.6='  DELETE  'outdschk
sk.7='  DELETE  'outdsali
sk.8='  DELETE  '$hiwork'.'subsys'.@DB2SDSF.ISFOUT                  '
sk.9='  DELETE  '$hiwork'.'subsys'.@DB2SDSF.ISFIN                   '
       sk.Ø=9
       if ownerb ¬= '*' then do
sk.1Ø='  DELETE  'outdsbnd
sk.11='  DELETE  'outdsfre
          sk.Ø=11
          if autosub = no then do
sk.12='  DELETE  '$hiwork'.'subsys'.'DBname'.PACKAGE                 '
sk.13='  DELETE  '$hiwork'.'subsys'.'DBname'.PLAN                    '
sk.14='  DELETE  '$hiwork'.'subsys'.'DBname'.PLANPACK                '
sk.15='  DELETE  '$hiwork'.'subsys'.'DBname'.UTILITY                 '
sk.16='//* ---------------------------------------------- *          '
             sk.Ø=16 ; end ; end
       jobw = FIDEL ; "alloc da('"outdsdel"') f("jobw") mod reuse"
       Call WriteRec
sk.1='//* ---------------------------------------------- *           '
       sk.Ø=1
       jobw = FISOU ; "alloc da('"outdssou"') f("jobw") mod reuse"
       Call WriteRec ; end
   /*-  Start Read/Write Tablespace          -*/
sk.1='//'jna'.F JOB ('$accn'),''Reset Copy Pending'',CLASS='$class',  '
sk.2='//      MSGCLASS='$msgcla',USER='$user',REGION='$region',       '
sk.3='//      MSGLEVEL=('$msglvl'),NOTIFY='$notif
sk.4='/*JOBPARM BYTES=999999,LINES=9999                              '
sk.5='//* ---------------------------------------------- *           '
sk.6='//* --------      Reset  COPY/CHKP       -------- *           '
sk.7='//* ---------------------------------------------- *           '
sk.8='//DB2PROC  JCLLIB ORDER=('$proclib')                           '
sk.9='//REXXØØ   EXEC   DB2REXX1                                     '
sk.1Ø='//REXXØØ.SYSTSIN  DD  *                                        '
sk.11='  ISPSTART CMD(@DB2FORØ 'Tsubsys','TDBname')                   '
/*implØ2*/
sk.12='//*ISPSTART CMD(@DB2RUNS 'TSubsys','TDBname',si,*)            '
sk.13='//* ------------------------------------------ *           '
    sk.Ø=13
    jobw = FIFOR ;  "alloc da('"outdsfor"') f("jobw") mod reuse"
    Call WriteRec
    if Full = yes then do
   /*-  Start job Iebcopy                       -*/
       if autosub = no then do
sk.1='//LAB1     IF (RC = Ø) THEN                               '
sk.2='//STEPØØ   EXEC   DB2REXX1                                '
sk.3='//REXXØØ.SYSTSIN  DD  *                                   '
sk.4='  ISPSTART CMD(@DB2UNØ5 'subsys','DBname',jobIebg)         '
sk.5='//LAB1END  ENDIF                                          '
sk.6='//* ------------------------------------------ *          '
          sk.Ø=6
          jobw = FISOU ; "alloc da('"outdssou"') f("jobw") mod reuse"
          Call WriteRec ; "free fi(fisou)" ; end
   /*-  job Dump                                -*/
       idgiul = date(j) ; idor   = space(translate(time(),'',':'),Ø)
       wrdatab = center(DBname,8)
sk.1='//'userid()'D JOB ('$accn'),''Dump Dataset'',CLASS='$class',   '
sk.2='//      MSGCLASS='$msgcla',USER='$user',REGION='$region',      '
sk.3='//      MSGLEVEL=('$msglvl'),NOTIFY='$notif
sk.4='/*JOBPARM BYTES=999999,LINES=9999                          '
sk.5='//* ------------------------------------------ *           '
sk.6='//*      Archive    dataset unload   'wrdatab '   *        '
sk.7='//* ------------------------------------------ *           '
sk.8='//STEP1  EXEC  PGM=ADRDSSU,PARM=''UTILMSG=YES,XABUFF=ABOVE16''  '
sk.9='//SYSPRINT DD SYSOUT=*                                    '
sk.1Ø='//OUTDD1   DD
DSN=SYSG.'subsys'.'KeepTime'.'DBname'.D'idgiul'.T'idor','
sk.11='//         DISP=(NEW,CATLG),UNIT='$unitta',LABEL=(1,SL),   '
sk.12='//         VOL=(,RETAIN,,99)                              '
sk.13='//SYSIN    DD  *                                          '
sk.14='  DUMP DATASET(          -                               '
sk.15='       INCLUDE(          -                               '
sk.16='               '$hiwork'.'subsys'.'DBname'.**  -          '
sk.17='               '$hiwork'.'DBname'.**           -          '
sk.18='             ))          -                               '
sk.19='     OUTDDNAME(          -                               '
sk.2Ø='                   OUTDD1 -                              '
sk.21='              )          -                               '
sk.22='     ALLEXCP             -                               '
sk.23='     CANCELERROR         -                               '
sk.24='     COMPRESS            -                               '
sk.25='     OPTIMIZE(4)         -                               '
sk.26='     WAIT(Ø,Ø)                                           '
sk.27='//* ------------------------------------------ *          '
       sk.Ø=27
       jobw = FIDMP ; "alloc da('"outdsdmp"') f("jobw") mod reuse"
       Call WriteRec ; end
   /*-  Display tables with CARD -1             -*/
    Do #b = 1 to #c ; say jk.#b ; end
    tcyls = trunc(ttracks / 15)
 say ''
 say '>>>>>>>>  +-----------------------------------------------------+'
 say '>>>>>>>>    Tablespace number
'right(fullstp,4,'Ø')
 if ctrsts > Ø then
 say '>>>>>>>>    Tables with Runstats older than 24h
'right(ctrsts,4,'Ø')
 say '>>>>>>>>    Job created
'right(njob,4,'Ø')
 say '>>>>>>>>    Unload Cyls space
'right(tcyls,4,'Ø')
 say '>>>>>>>>  +-----------------------------------------------------+'
 say '' ; say ''
   /*—  Automatic submit procedure           —*/
    if autosub = yes then do
   /*—  Automatic start Dump job             —*/
sk.1 ='//LAB1     IF (STEP1.RC = Ø) THEN                            '
sk.2 ='//SUBMIT  EXEC PGM=IEBGENER                                  '
sk.3 ='//SYSUT1   DD  DISP=SHR,DSN='outdsdel
sk.4 ='//SYSUT2   DD  SYSOUT=(,INTRDR)                              '
sk.5 ='//SYSIN    DD  DUMMY                                         '
sk.6 ='//SYSPRINT DD  SYSOUT=*                                      '
sk.7 ='//LAB1END  ENDIF                                             '
sk.8 ='//* ------------------------------------------ *            '
       sk.Ø=8
       jobw = FIDMP ; "alloc da('"outdsdmp"') f("jobw") mod reuse"
       Call WriteRec
sk.1='//LAB1     IF (RC = Ø) THEN                                   '
sk.2='//STEPØØ   EXEC   DB2REXX1                                    '
sk.3='//REXXØØ.SYSTSIN  DD  *                                       '
sk.4='  ISPSTART CMD(@DB2UNØ5 'subsys','DBname',jobDump)            '
sk.5='//LAB1END  ENDIF                                              '
sk.6='//* ------------------------------------------ *              '
   sk.Ø=6 ; jobw = FISOU ; "alloc da('"outdssou"') f("jobw") mod reuse"
       Call WriteRec
       "free fi(fisou)" ; xx=OUTTRAP(trpsubØ.)
          ADDRESS TSO "SUBMIT '"outdsunl"'" ; xx=OUTTRAP(OFF)
       if rc > Ø then do
          do #a = 1 to trpsubØ.Ø ; say trpsubØ.#a ; end ; exit ; end
       mess  = substr(trpsubØ.1,1,22) ; say '' ; say '' ; say time()
       say time() '          U n l o a d   DATA              '
       say time() '——>> 'mess' has been submitted .....  '
       say time() ; say '' ; say '' ; end ; return
   /*—  Routine DataSet allocation          —*/
 AllocØØ :
   /*—  Reorg/Unload file allocation         —*/
    outdsunl= $hiwork'.'subsys'.'DBname'.@DB2UNØØ.JOBUNLO'
    prmalloc = subsys' 'outdsunl' Ø 45,15 f,b 8Ø 2792Ø fiunl yes'
    call @db2allØ prmalloc ; if word(result,1) = 99 then exit
   /*—  Load file allocation                 —*/
```

```
     outdsloa= $hiwork'.'subsys'.'DBname'.@DB2UN00.JOBLOAD'
     prmalloc = subsys' 'outdsloa' 0 30,15 f,b 80 27920 filoa yes'
     call @db2all0 prmalloc ; if word(result,1) = 99 then exit
   /*- Delete data-set file allocation       -*/
     outdsdel= $hiwork'.'subsys'.'DBname'.@DB2UN00.JOBDELE'
     prmalloc = subsys' 'outdsdel' 0 30,15 f,b 80 27920 fidel yes'
     call @db2all0 prmalloc ; if word(result,1) = 99 then exit
   /*- Start Force file allocation           -*/
     outdsfor= $hiwork'.'subsys'.'DBname'.@DB2UN00.JOBFORC'
     prmalloc = subsys' 'outdsfor' 0 1,15 f,b 80 27920 fifor yes'
     call @db2all0 prmalloc ; if word(result,1) = 99 then exit
   /*- Iebgener file allocation              -*/
     outdsieb= $hiwork'.'subsys'.'DBname'.@DB2UN00.JOBIEBG'
     prmalloc = subsys' 'outdsieb' 0 30,15 f,b 80 27920 fiieb yes'
     call @db2all0 prmalloc ; if word(result,1) = 99 then exit
   /*- Work Syspunch file allocation         -*/
     outdsmod= $hiwork'.'subsys'.'DBname'.@DB2UN00.MOD'
     prmalloc = subsys' 'outdsmod' 0 150,15 f,b 80 27920 fimod yes'
     call @db2all0 prmalloc ; if word(result,1) = 99 then ; exit
   /*- OverWrite file allocation             -*/
     outdsovr= $hiwork'.'subsys'.'DBname'.@DB2UN00.JOBOVER'
     prmalloc = subsys' 'outdsovr' 0 1,15 f,b 80 27920 fiovr yes'
     call @db2all0 prmalloc ; if word(result,1) = 99 then exit
     if Full = yes then do
   /*- Check RC file allocation              -*/
        outdschk= $hiwork'.'subsys'.'DBname'.@DB2UN00.CHECKRC'
        prmalloc = subsys' 'outdschk' 0 1,15 f,b 80 27920 fichk yes'
        call @db2all0 prmalloc ; if word(result,1) = 99 then exit
/*impl03*/
   /*- Backup Data file allocation           -*/
        outdsdmp= $hiwork'.'subsys'.'DBname'.@DB2UN00.JOBDUMP'
        prmalloc = subsys' 'outdsdmp' 0 1,15 f,b 80 27920 fidmp yes'
        call @db2all0 prmalloc ; if word(result,1) = 99 then exit
   /*- Report file allocation                -*/
        outdsrep= $hiwork'.'subsys'.'DBname'.@DB2UN00.REPORT'
        prmalloc = subsys' 'outdsrep' 20 300,90 f,b 133 1330 firep yes'
        call @db2all0 prmalloc ; if word(result,1) = 99 then exit
/*impl04*/
   /*- Archive DDL file allocation           -*/
        outdsddl= $hiwork'.'subsys'.'DBname'.@DB2UN00.DDL'
        prmalloc = subsys' 'outdsddl' 20 150,90 f,b 80 27920 fiddl yes'
        call @db2all0 prmalloc ; if word(result,1) = 99 then exit
   /*- Save Outputs file Report allocation  -*/
        outdssou= $hiwork'.'subsys'.'DBname'.@DB2UN00.JOBSOUT'
        prmalloc = subsys' 'outdssou' 0 1,15 f,b 80 27920 fisou yes'
        call @db2all0 prmalloc ; if word(result,1) = 99 then exit
sk.1='//'jna'.A JOB ('$accn'),''Archiv. Output'',CLASS='$class',      '
sk.2='//      MSGCLASS='$msgcla',USER='$user',REGION='$region',       '
sk.3='//      MSGLEVEL=('$msglvl'),NOTIFY='$notif
sk.4='/*JOBPARM BYTES=999999,LINES=9999                               '
sk.5='//* ------------------------------------------ *                '
sk.6='//*            Archive outputs to Dataset           *           '
sk.7='//* ---------------------------------------------- *           '
sk.8='//DB2PROC  JCLLIB ORDER=('$proclib')                           '
sk.9='//REXXØØ   EXEC   DB2REXX1                                     '
sk.1Ø='//REXXØØ.SYSTSIN  DD  *                                       '
       sk.Ø=1Ø ; jobw = FISOU ; Call WriteRec
/*implØ5*/
   /*-  Bind file allocation                  -*/
       outdsbnd= $hiwork'.'subsys'.'DBname'.@DB2UNØØ.JOBBIND'
       prmalloc = subsys' 'outdsbnd' Ø 1,15 f,b 8Ø 2792Ø fibnd yes'
       call @db2allØ prmalloc ; if word(result,1) = 99 then exit
sk.1='//'jna'.B JOB ('$accn'),''Bind Extractor'',CLASS='$class',      '
sk.2='//      MSGCLASS='$msgcla',USER='$user',REGION='$region',       '
sk.3='//      MSGLEVEL=('$msglvl'),NOTIFY='$notif
sk.4='/*JOBPARM BYTES=999999,LINES=9999                              '
sk.5='//* ---------------------------------------------- *           '
sk.6='//* --     Extract Bind Catalog Statements     -- *            '
sk.7='//* ---------------------------------------------- *           '
sk.8='//DB2PROC   JCLLIB ORDER=('$proclib')                          '
sk.9='//JOBLIB    DD  DISP=SHR,DSN='$dsnload
sk.1Ø='//          DD  DISP=SHR,DSN='$plilink
sk.11='//          DD  DISP=SHR,DSN='$sibmlnk
sk.12='//* ---------------------------------------------- *          '
sk.13='//DELVIEW  EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø                      '
sk.14='//SYSTSPRT  DD  SYSOUT=*                                      '
sk.15='//SYSPRINT  DD  SYSOUT=*                                      '
sk.16='//SYSTSIN   DD  *                                             '
sk.17='DSN SYSTEM('subsys')                                          '
sk.18='RUN PROGRAM('$tep2pgm') PLAN('$tep2pln') LIB('''$runlib''')    '
sk.19='//SYSIN    DD  *                                              '
sk.2Ø='DROP VIEW VBPPACK   ;                                         '
sk.21='DROP VIEW VBPACK    ;                                         '
sk.22='DROP VIEW VBPMEMB   ;                                         '
sk.23='//* ---------------------------------------------- *          '
sk.24='//LABØ     IF (DELVIEW.RC LT 9) THEN                          '
sk.25='//CREVIEW  EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø                      '
sk.26='//SYSTSPRT  DD  SYSOUT=*                                      '
sk.27='//SYSPRINT  DD  SYSOUT=*                                      '
sk.28='//SYSTSIN   DD  *                                             '
sk.29='DSN SYSTEM('subsys')                                          '
sk.3Ø='RUN PROGRAM('$tep2pgm') PLAN('$tep2pln') LIB('''$runlib''')    '
sk.31='//SYSIN    DD  *                                              '
sk.32='                                                              '
sk.33='  CREATE VIEW VBPPACK                                         '
sk.34='  (A,B,C,D,E,F,G,H,I,L,M,N,O,P,Q,R,S) AS                      '
sk.35='  SELECT  B.SEQNO,A.NAME,B.NAME,B.COLLID,A.CREATOR,A.QUALIFIER,'
sk.36='          VALIDATE,A.ACQUIRE,A.ISOLATION,A.RELEASE,A.EXPLAN,   '
sk.37='
A.CACHESIZE,A.EXPREDICATE,A.DEGREE,A.SQLRULES,A.DISCONNECT,'
sk.38='          A.DYNAMICRULES                                      '
sk.39='          FROM 'catnam'.SYSPLAN      A ,                      '
sk.4Ø='               'catnam'.SYSPACKLIST  B                         '
sk.41='          WHERE B.COLLID   LIKE ''%''      AND                 '
sk.42='                A.NAME     LIKE ''%''      AND               '
sk.43='                A.CREATOR  =  '''ownerb''' AND               '
sk.44='                A.NAME     =    B.PLANNAME ;                '
sk.45='                                                            '
sk.46='  CREATE VIEW VBPACK (A,B,C,D,E,F,G,H,I,L,M,N,O) AS          '
sk.47='  SELECT  A.NAME,A.COLLID,A.OWNER,A.QUALIFIER,               '
sk.48='          A.VALIDATE,A.EXPLAIN,A.PDSNAME,A.ISOLATION,A.RELEASE,'
sk.49='          A.DEGREE,A.DYNAMICRULES,A.DEFERPREP,A.SQLERROR     '
sk.50='          FROM 'catnam'.SYSPACKAGE   A                      '
sk.51='          WHERE A.COLLID   LIKE ''%''      AND               '
sk.52='                A.NAME     LIKE ''%''      AND               '
sk.53='                A.OWNER    =  '''ownerb''' ;                 '
sk.54='                                                            '
sk.55='  CREATE VIEW VBPMEMB (A,B,C,D,E,F,G,H,I,L,M,N,O,P,Q,R) AS    '
sk.56='  SELECT
A.NAME,B.NAME,A.CREATOR,A.QUALIFIER,A.VALIDATE,ACQUIRE,'
sk.57='          A.ISOLATION,A.RELEASE,A.EXPLAN,B.PDSNAME,A.CACHESIZE,'
sk.58='          A.EXPREDICATE,A.DEGREE,A.SQLRULES,A.DISCONNECT,     '
sk.59='          A.DYNAMICRULES                                    '
sk.60='          FROM 'catnam'.SYSPLAN     A ,                     '
sk.61='               'catnam'.SYSDBRM     B                      '
sk.62='          WHERE A.NAME     LIKE ''%''      AND               '
sk.63='                A.CREATOR  =  '''ownerb''' AND               '
sk.64='                A.NAME     =     B.PLNAME  ;                '
sk.65='                                                            '
sk.66='//* ------------------------------------------- *           '
sk.67='//LAB1      IF (CREVIEW.RC EQ 0) THEN                       '
sk.68='//REXX00   EXEC   DB2REXX1                                  '
sk.69='//REXX00.SYSTSIN  DD  *                                     '
sk.70='  ISPSTART CMD(@DB2BIND 'subsys',yyy,'DBname')              '
sk.71='//LAB0END  ENDIF                                            '
sk.72='//LAB1END  ENDIF                                            '
sk.73='//* ------------------------------------------- *           '
sk.74='//REXX00   EXEC   DB2REXX1,COND=EVEN                        '
sk.75='//REXX00.SYSTSIN  DD  *                                     '
sk.76='  ISPSTART CMD+                                             '
sk.77='(@DB2OUBK 'Subsys','$user'B,'outdsrep',CREAPACK,'DBname')    '
sk.78='//* ------------------------------------------- *           '
      sk.0=78 ;jobw = fibnd ; Call WriteRec
/*impl06*/
     end
   return
  /*-  Routine Display % elaboration        -*/
 VerElab :
   select
      when #d = welab25 then do
         say '  25%  -$----  Record elab. n 'right(welab25,4,'0')
         end
      when #d = welab50 then do
         say '  50%  --$---  Record elab. n 'right(welab50,4,'0')
         end
      when #d = welab75 then do
```

```
          say '  75%  ————$–  Record elab. n 'right(welab75,4,'Ø')
          end
       when #d = welabØØ then do
          say ' 1ØØ%  —————$  Record elab. n 'right(welabØØ,4,'Ø')
          say ''
          end
       otherwise
          nop ; end ; return
   /*–  Write record routine                   –*/
 WriteRec :
    "EXECIO * DISKW "jobw" (STEM sk. FINIS"
 ClearRec:
    DO #f = 1 to sk.Ø ; sk.#f = blk ; end ; return
/*implØ7*/
   /*–  Write Start Tablespace RO/RW         –*/
 WrstrRO :
    jobw = firep
    "alloc da('"outdsrep"(STRRO"right(njobRO,2,'Ø')")') f("jobw") shr
reuse"
    sy.Ø = #x ; "EXECIO * DISKW "jobw" (STEM sy. FINIS"
    njobRO = njobRO + 1
     DO #f = 1 to sy.Ø ; sy.#f = blk end ; return
 WrstrRW :
    jobw = firep
    "alloc da('"outdsrep"(STRRW"right(njobRW,2,'Ø')")') f("jobw") shr
reuse"
    sz.Ø = #x ; "EXECIO * DISKW "jobw" (STEM sz. FINIS"
    njobRW = njobRW + 1
    DO #f = 1 to sz.Ø ; sz.#f = blk ; end ; #x = Ø ; return
   /*–  Free Work areas                        –*/
 FreeØ :
    xx=outtrap(trpdummy.) ; address tso "delete '"outdsprt"'"
       address tso "delete '"outdsrec"'" ; xx=outtrap(off)
 Free  :
    xx=outtrap(trpdummy.)
       "free fi(systsinp)" ; address tso "delete '"outdstsin"'"
       "free fi(sysprint)" ; "free fi(sysrecØØ)"
       "free fi(syspunch)" ; "free fi(sysin)"
       "free fi(firep)" ; address tso "delete '"outdsin"'"
    xx=outtrap(off) ; return
```

$DB2UN02 REXX EXEC

```
/* REXX */
   /* ------------------------------------------------------------ */
   /*-                 Data Tool for DB management                -*/
   /*-                  Update Syspunch creator                   -*/
   /* ----------------------------------------- -----------------*/
arg parmin ; parm = translate(parmin,' ',',')
nparm  = words(parm) ;Subsys = word(parm,1) ;Datas  = word(parm,2)
Creunl = word(parm,3);Creloa = word(parm,4) ;ReusVS = word(parm,5)
```

```
   /*-  Test input parameters             ---*/
 if nparm < 5 then do
    say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Parameter string is incomplete !!!!'
    say '>>>>>>>>           'parmin
    say '' ; say '' ; exit ;end
   /*-  Parameters assignment                   -*/
 call @db2parØ Subsys ; if word(result,1) = 99 then exit
 $lpar =word(result,1) ; $accn =word(result,2) ; $class =word(result,3)
 $msgcla = word(result,4);$region  = word(result,5) ;$msglvl =
word(result,6)
 $notif  = word(result,7);$user    = word(result,8) ;$unitda =
word(result,9)
 $unitta = word(result,1Ø);$esunit = word(result,11);$prt    =
word(result,12)
 $hiwork = word(result,13);$db2ver = word(result,14);$ctsubs =
word(result,15)
 $librexx= word(result,16);$parmlib= word(result,17);$proclib=
word(result,18)
 $jcllib = word(result,19);$report = word(result,2Ø);$libexec=
word(result,21)
 $isptenu= word(result,22);$isppenu= word(result,23);$ispmenu=
word(result,24)
 $ispslib= word(result,25);$plilink= word(result,26);$sibmlnk=
word(result,27)
 $sortlib= word(result,28);$hilvlDB= word(result,29);$runlib =
word(result,3Ø)
 $dsnload= word(result,31);$tep2pgm= word(result,32);$tep2pln=
word(result,33)
 $unlopgm= word(result,34);$unlopln= word(result,35);$dunlopg=
word(result,36)
 $dunlopl= word(result,37);$dsnproc= word(result,38)
   /*- Change creator in syspunch data-sets --*/
 $vdatas  = $hiwork'.'datas'.PUNCH'
 $vcreunl = Creunl ; $vcreloa = Creloa
 $vesunit = $esunit; $vreusVS = ReusVS
 xx=OUTTRAP(trpØ6.)
    address ispexec 'vput ($vdatas)  profile'
    address ispexec 'vput ($vcreunl) profile'
    address ispexec 'vput ($vcreloa) profile'
    address ispexec 'vput ($vesunit) profile'
    address ispexec 'vput ($vreusVS) profile'
    "ispexec edit dataset('"$hiwork"."datas".PUNCH') macro(@mdb2Ø4Ø)"
 xx=OUTTRAP(OFF)
 if rc = Ø then do
    "alloc da('"$vdatas"') f(punchØØ) shr reuse"
    "execio 3 diskr punchØØ (stem punchØØ. finis"
    say ''
    say '__________*  Start display of contents to SYSPUNCH  *________'
    say ''
    do #a = 1 to punchØØ.Ø ; say punchØØ.#a ;  end
    say ''
    say '_____________*  Fine   display contenuto SYSPUNCH  *_________'
    say '' ; end ; return
```

$DB2UN02 REXX EXEC

```
/* REXX */
   /* -------------------------------------------------------------- */
   /*-                 Data Tool for DB management                  -*/
   /*-                 OverWrite syspunch Dataset                   -*/
   /* -------------------------------------------------------------- */
arg parmin ; parm = translate(parmin,' ',',') ; nparm = words(parm)
subsys  = word(parm,1) ; DBname  = word(parm,2)
   /*— Test input parameters               —*/
 if nparm < 2 then do
    say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Parameter string is incomplete !!!!'
    say '>>>>>>>>          'parmin
    say '>>>>>>>>' ; say '' ; say '' ; exit ; end
   /*— Parameters assignment               —*/
 call @db2parØ subsys ; if word(result,1) = 99 then exit
 $lpar =word(result,1) ; $accn =word(result,2) ; $class =word(result,3)
 $msgcla = word(result,4);$region  = word(result,5) ;$msglvl =
word(result,6)
 $notif  = word(result,7);$user    = word(result,8) ;$unitda =
word(result,9)
 $unitta = word(result,1Ø);$esunit = word(result,11);$prt    =
word(result,12)
 $hiwork = word(result,13);$db2ver = word(result,14);$ctsubs =
word(result,15)
 $librexx= word(result,16);$parmlib= word(result,17);$proclib=
word(result,18)
 $jcllib = word(result,19);$report = word(result,2Ø);$libexec=
word(result,21)
 $isptenu= word(result,22);$isppenu= word(result,23);$ispmenu=
word(result,24)
 $ispslib= word(result,25);$plilink= word(result,26);$sibmlnk=
word(result,27)
 $sortlib= word(result,28);$hilvlDB= word(result,29);$runlib =
word(result,3Ø)
 $dsnload= word(result,31);$tep2pgm= word(result,32);$tep2pln=
word(result,33)
 $unlopgm= word(result,34);$unlopln= word(result,35);$dunlopg=
word(result,36)
 $dunlopl= word(result,37);$dsnproc= word(result,38)
   /*—  Work areas initialization          —*/
 blk   = ;#c = Ø ;wtsname = ;punt = 1 ;#f = Ø ;#g  = Ø ;totrc = Ø
   /*— SYSRECØØ file allocation            —*/
 outdsrec= $hiwork'.'subsys'.'DBname'.@DB2UNØØ.SYSRECØØ'
 "alloc da('"outdsrec"') f(sysrecØØ) shr reuse"
   /*— Read extracted Records              —*/
 xx=outtrap(trpreadØ1.);"execio * diskr sysrecØØ (stem sysrecØØ. finis"
xx=outtrap(off)
if rc > Ø then do
   do #a = 1 to trpreadØ1.Ø ; say trpreadØ1.#a ; end
   say '' ; say '' ; say '>>>>>>>>'
   say '>>>>>>>>  Lettura file "'outdsrec'"    '
   say '>>>>>>>>  ha dato RC='rc'. Controllare.'
   say '>>>>>>>>' ; say '' ; say '' ; exit ; end
  /*—  SYSPUNCH input file allocation       —*/
outdsmod= $hiwork'.'subsys'.'DBname'.@DB2UNØØ.MOD'
"alloc da('"outdsmod"') f(fimod) shr reuse"
xx=outtrap(trpreadØ2.) ; "execio * diskr fimod (stem fimod. finis"
 "free fi(fimod)" ; xx=outtrap(off)
if rc > Ø then do
   do #a = 1 to trpreadØ2.Ø ; say trpreadØ2.#a ; end
   say '' ; say '' ; say '>>>>>>>>'
   say '>>>>>>>>  Lettura file "'outdsmod'"    '
   say '>>>>>>>>  ha dato RC='rc'. Controllare.'
   say '>>>>>>>>' ; say '' ; say '' ; exit ; end
  /*—  Extract TSname                       —*/
do #a = 1 to sysrecØØ.Ø
   wtsname = $tsname ; $tsname = strip(substr(sysrecØØ.#a,51,8))
   if wtsname ¬= $tsname then do
      #c = #c + 1 ; jk.#c=$tsname ; end
   wtsname = $tsname ; end
  /*—  OverWrite routine                    —*/
do #b = 1 to #c
   nome = blk ; call @db2tbna jk.#b
      if word(result,1) = 99 then  exit
   nome = word(result,1) ; #f = 1
   do #e = punt to fimod.Ø
      #g = #e + 1 ; fiout.#f = fimod.#e
      if (substr(fimod.#e,2,1) = ')' & ,
         substr(fimod.#g,1,9) = 'LOAD DATA') | ,
         (substr(fimod.#e,2,1) = ')' & #e = fimod.Ø) then do
         outds1 = $hiwork'.'DBname'.'nome'.PUNCH'
         "alloc da('"outds1"') f(fiout) shr reuse"
         "newstack"
         "execio " #f "diskw fiout (stem fiout. finis"
         if rc = Ø then totrc = totrc + 1
         "delstack"
         "free fi(fiout)"
         punt = punt + #f ; leave ; end
      #f = #f + 1 ; end ; end
  /*—  Check overvritten data-set           —*/
if totrc ¬= #c  then do
   say '' ; say '' ; say '' ; say '>>>>>>>>'
   say '>>>>>>>>            W  A  R  N  I  N  G  !!!! '
   say '>>>>>>>>  Not all syspunch datasets have been overwritten'
   say '>>>>>>>>' ; say '' ; end
say '' ; say ''
say '>>>>>>>>  +-------------------------------------+'
say '>>>>>>>>    Total SYSPUNCH dataset overwritten N> ' totrc
```

```
 say '>>>>>>>>    Total SYSPUNCH dataset              N> ' #c
 say '>>>>>>>>  +------------------------------------+'
 say '' ; say '' ; say '' ; exit
```

$DB2UN03 REXX EXEC

```
/* REXX */
   /* ------------------------------------------------------------- */
   /*-                  Data Tool for DB management                 -*/
   /*-                  Archive output on dataset                   -*/
   /* ------------------------------------------------------------- */
arg parmin
parm = translate(parmin,' ',',') ;nparm = words(parm) ;
datas = word(parm,1) ; mem = word(parm,2) ;outna = word(parm,3)
   /*—  Test input parameters             ———*/
 if nparm < 3 then do
    say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Parameter string is incomplete !!!!'
    say '>>>>>>>>           'parmin
    say '>>>>>>>>' ; say '' ; say '' ; exit ; end
 xx=outtrap(trp1.)
    address tso "output "outna" print('"datas"("mem")')
    keep hold"
 xx=outtrap(off)
 if trp1.Ø > Ø then do
    say ''
    do a = 1 to trp1.Ø ; say trp1.a ; end
    say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Problem during the archive of job "'outna'" '
    say '>>>>>>>>' ; say '' ; say '' ; exit ; end
 else do
    say '' ; say '>>>>>>>>  Archive of "'outna'"  O.K.'
    say '' ; end ; exit
```

$DB2UN04 REXX EXEC

```
/* REXX */
   /* ------------------------------------------------------------- */
   /*-                  Data Tool for DB management                 -*/
   /*-     Monitor Unload job & start archive sysout to dataset     -*/
   /* ------------------------------------------------------------- */
 arg parmin ; parm   = translate(parmin,' ',',')
 nparm  = words(parm) ; datas  = word(parm,1) ;jobnam = word(parm,2)
   /*—  Test input parameters             ———*/
 if nparm < 2 then do
    say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Parameter string is incomplete !!!!'
    say '>>>>>>>>           'parmin
    say '>>>>>>>>' ; say '' ; say '' ; exit ; end
   /*—   Work areas initialization           —*/
 blk  =  ; ctr = 1
   /*–    Verify Start/Stop activity          –*/
 do while ctr ¬= Ø
    address tso "delayØØ 45"
    call verifyØØ
   /*–    Unload in progress.....             –*/
    if ctr > Ø then do
       say '——> ' time() ' N> 'ctr'  job 'jobnam'* running'
       say '' ; end  ; end
   /*–    Activity ended                       –*/
 xx=outtrap(trpØ3.) ; address tso "submit '"datas"'" ;xx=outtrap(off)
 if rc > Ø then do
    do a = 1 to trpØ3.Ø ; say trpØ3.a ; end ; exit ;end
 mess  = substr(trpØ3.1,1,22)
 say '' ; say time()
 say time() '——>> il 'mess' e" stato sottomesso.... '
 say time() ; say '' ; say '' ;exit
   /*–   Routine verify active job             –*/
 verifyØØ:
    say  ''  ; ctr = Ø
    cvtmser=d2x(c2d(storage(1Ø,4))+6Ø)
    base=d2x(c2d(storage(cvtmser,4)))
    cscb=d2x(c2d(storage(base,4)))
    do while cscb ¬= 'ØØØØØØØØ'
       nameaddr=d2x(x2d(cscb)+8)
       jobname=storage(nameaddr,6)
       jobstop = jobnam
       if left(jobname,1) ¬=' ' & ,
          left(jobname,6) = jobstop  then do
          ctr = ctr + 1 ;end
       cscb=d2x(c2d(storage(cscb,4))) ; end
    if ctr = Ø then do
       say '>>>>>>>>' ; say '>>>>>>>>  job 'jobnam'* ended.    '
       say '>>>>>>>>  Start archive sysout to data-set '
       say '>>>>>>>>' ; say '' ; end ; return
```

$DB2UN06 REXX EXEC

```
/* REXX */
   /* ------------------------------------------------------------- */
   /*-                  Data Tool for DB management                -*/
   /*-        Check Return Code and start Iebgener/Dump job        -*/
   /* ------------------------------------------------------------- */
arg parmin ;parm = translate(parmin,' ',',') ; nparm = words(parm)
subsys = word(parm,1) ;datab = word(parm,2) ; nextjob  = word(parm,3)
   /*–  Test input parameters                 –*/
 if nparm < 3 then do
    say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Parameter string is incomplete !!!!'
    say '>>>>>>>>           'parmin
    say '>>>>>>>>' ; say '' ; say '' ; exit ; end
   /*-  Parameters assignment                 -*/
 call @db2parØ subsys ; if word(result,1) = 99 then exit
 $lpar =word(result,1) ; $accn =word(result,2) ; $class =word(result,3)
 $msgcla = word(result,4);$region  = word(result,5) ;$msglvl =
word(result,6)
 $notif  = word(result,7);$user    = word(result,8) ;$unitda =
word(result,9)
 $unitta = word(result,1Ø);$esunit = word(result,11);$prt    =
word(result,12)
 $hiwork = word(result,13);$db2ver = word(result,14);$ctsubs =
word(result,15)
 $librexx= word(result,16);$parmlib= word(result,17);$proclib=
word(result,18)
 $jcllib = word(result,19);$report = word(result,2Ø);$libexec=
word(result,21)
 $isptenu= word(result,22);$isppenu= word(result,23);$ispmenu=
word(result,24)
 $ispslib= word(result,25);$plilink= word(result,26);$sibmlnk=
word(result,27)
 $sortlib= word(result,28);$hilvlDB= word(result,29);$runlib =
word(result,3Ø)
 $dsnload= word(result,31);$tep2pgm= word(result,32);$tep2pln=
word(result,33)
 $unlopgm= word(result,34);$unlopln= word(result,35);$dunlopg=
word(result,36)
 $dunlopl= word(result,37);$dsnproc= word(result,38)
   /*-   Work areas initialization           -*/
 blk  =  ; koswitch =  ;sidump = on ;siiebg = on
   /*-  CheckList file allocation            -*/
 outdschk= $hiwork'.'subsys'.'datab'.@DB2UNØØ.CHECKRC'
 jobw = fichk ; "alloc da('"outdschk"') f("jobw") shr reuse"
   /*-  Analize CheckList file               -*/
 "execio * diskr fichk (stem fichk. finis" ; "free fi(fichk)"
 do #d = 1 to fichk.Ø
    koswitch = word(fichk.#d,7)
    if koswitch = 'Verificare' then do
       sidump = off ; siiebg = off ; end ; end
 select
    when sidump = on & nextjob = jobDump then do
       outdsdmp= $hiwork'.'subsys'.'datab'.@DB2UNØØ.JOBDUMP'
       xx=outtrap(trpsubØ.) ; address tso "submit '"outdsdmp"'"
       xx=outtrap(off)
       if rc > Ø then do
          do a = 1 to trpsubØ.Ø ; say trpsubØ.a ; end ; exit ; end
       mess  = substr(trpsubØ.1,1,22)
       say '' ; say '' ; say time()
       say time() '      Starting Data Backup on Tape       '
       say time() '-->> 'mess' has been submitted ...... '
       say time() ; say '' ;  say '' ; end
    when siiebg = on & nextjob = jobiebg then do
       outdsieb= $hiwork'.'subsys'.'datab'.@DB2UNØØ.JOBIEBG'
       xx=outtrap(trpsubØ.) ; address tso "submit '"outdsieb"'"
```

```
        xx=outtrap(off)
        if rc > Ø then do
           do a = 1 to trpsubØ.Ø ; say trpsubØ.a ; end ; exit ; end
        mess  = substr(trpsubØ.1,1,22)
        say '' ; say '' ; say time()
        say time() '      Starting Iebgener job              '
        say time() '——>> 'mess' has been submitted ...... '
        say time() ; say '' ; say '' ; end
     otherwise
        say '' ; say '' ; say '>>>>>>>>>'
        say '>>>>>>>>>  ————>>>  W A R N I N G <<<———— '
        say '>>>>>>>>>  ————>>>  W A R N I N G <<<———— '
        say '>>>>>>>>>'
        say '>>>>>>>>>  Unload Data is ended with RC > Ø           '
        say '>>>>>>>>>  Verify the reason of the problem           '
        say '>>>>>>>>>  All automatic job scheduling is stopped  '
        say '>>>>>>>>>' ; say '' ; say '' ; exit ; end ; exit
```

$DB2UN06 REXX EXEC

```
/* REXX */
   /* ------------------------------------------------------------- */
   /*-                  Data Tool for DB management                -*/
   /*-             Monitor Load & Reset COPY pending State         -*/
   /* ------------------------------------------------------------- */
 arg parmin ; parm  = translate(parmin,' ',',')
 nparm  = words(parm) ; Subsys = word(parm,1)
 Jobnam = word(parm,2); DBname = word(parm,3)
   /*—  Test input parameters               —*/
 if nparm < 3 then do
    say '' ; say ''  ; say '>>>>>>>>>'
    say '>>>>>>>>>  Parameter string is incomplete !!!!'
    say '>>>>>>>>>           'parmin
    say '>>>>>>>>>' ; say '' ; say '' ; exit ; end
   /*—  Parameters assignment               —*/
 call @db2parØ Subsys ; if word(result,1) = 99 then exit
 $lpar =word(result,1) ; $accn =word(result,2) ; $class =word(result,3)
 $msgcla = word(result,4);$region  = word(result,5) ;$msglvl =
word(result,6)
 $notif  = word(result,7);$user    = word(result,8) ;$unitda =
word(result,9)
 $unitta = word(result,1Ø);$esunit = word(result,11);$prt    =
word(result,12)
 $hiwork = word(result,13);$db2ver = word(result,14);$ctsubs =
word(result,15)
 $librexx= word(result,16);$parmlib= word(result,17);$proclib=
word(result,18)
 $jcllib = word(result,19);$report = word(result,2Ø);$libexec=
word(result,21)
 $isptenu= word(result,22);$isppenu= word(result,23);$ispmenu=
word(result,24)
```

```
 $ispslib= word(result,25);$plilink= word(result,26);$sibmlnk=
word(result,27)
 $sortlib= word(result,28);$hilvlDB= word(result,29);$runlib =
word(result,30)
 $dsnload= word(result,31);$tep2pgm= word(result,32);$tep2pln=
word(result,33)
 $unlopgm= word(result,34);$unlopln= word(result,35);$dunlopg=
word(result,36)
 $dunlopl= word(result,37);$dsnproc= word(result,38)
   /*–   Work areas initialization          –*/
 blk   = ;outdsfor = $hiwork'.'subsys'.'DBname'.@DB2UN00.JOBFORC'
   /*–   Verify Load in progress            –*/
 say '' ; parmsdsf = subsys || ','jobnam'*,240,75'
 Call @db2sdsf parmsdsf
 if word(result,1) = 99 then  do
   say '' ; say '>>>>>>>>'
   say '>>>>>>>>  Error during Load activity monitor. Verify manually '
   say '>>>>>>>>' ; say '' ; exit  ; end
 if word(result,1) = 98 then  do
    ctr = right(word(result,2),3,'0')
    say ''
    say '>>>>>>>>'
    say '>>>>>>>>  5 hours are spent from monitor Load start activity '
    say '>>>>>>>> 'ctr' job are already running. Verify the reasons of
problem'
    say '>>>>>>>>' ; say '' ; exit ; end
 if word(result,1) = 00 then  do
    say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Laod activity is ended.               '
    say '>>>>>>>>  Start DataBase (rw) and Start Runstats.  '
    say '>>>>>>>>' ; say ''
    xx=outtrap(trpsub0.) ; address tso "submit '"outdsfor"'"
    xx=outtrap(off)
    if rc > 0 then do
       do a = 1 to trpsub0.0 ; say trpsub0.a ; end ; exit ; end
    mess  = substr(trpsub0.1,1,22)
    say '' ;say '' ; say time()
    say time() '           Start Force & Runstats DataBase   '
    say time() '——>> 'mess' has been submitted ....       '
    say time()  ; say '' ; say '' ; end  ; exit
```

$DB2TBNA REXX EXEC

```
/* REXX */
   /*-------------------------------------------------------------*/
   /*-                  Data Tool for DB management               -*/
   /*-                   Dsname Routine creation                  -*/
   /*-------------------------------------------------------------*/
arg parmin ; parm = translate(parmin,' ',',')
nparm = words(parm) ; tbname = word(parm,1)
   /*–  Test input parameters               –*/
```

```
 if nparm < 1 then do
    say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Parameter string is incomplete !!!!'
    say '>>>>>>>>           'parmin
    say '>>>>>>>>' ; say '' ; say '' ; $exitc = 99
    return $exitc
    exit ; end
   /*-   Work areas initialization           -*/
 backname =  ; nstrin1  = Ø ;lstrinok = si ;nstrinok = si
   /*-   Dsname Routine creation             -*/
 wtbnameØ = strip(translate(tbname,' ','_'))
 wtbname1 = length(space(wtbnameØ,Ø))
 wtbname2 = length(tbname) - wtbname1
 if wtbname2  >= words(wtbnameØ) then
    tbname = strip(translate(tbname,'$','_'))
 tabel   = tbname ;tablew1 = strip(translate(tabel,'  ','_'))
 nstrin  = words(tablew1)
 do #a = 1 to nstrin ; #vett.#a =  word(tablew1,#a) ; end
 do while lstrinok = si | nstrinok = si
    call checknum ;call checklen ;end
 do #e = 1 to nstrin
    if #e = 1 then backname = #vett.#e
    else backname = backname ||'.'|| #vett.#e
    end ; return backname
 exit
 checknum :
    ctr1 = Ø
    do #b = 1 to nstrin
       if datatype(substr(#vett.#b,1,1)) = num then do
          tablew4.#b = $ || #vett.#b ; ctr1 = ctr1 + 1 ; end
       else tablew4.#b = #vett.#b ; end
    if ctr1 = Ø then nstrinok = no ; return
 checklen :
    #d = Ø ; nstrin1 = Ø
    do #c = 1 to nstrin
       #d = #d + 1 ; lstrin = length(tablew4.#c)
       cstrin = tablew4.#c
       if lstrin > 8 then do
          nstrin1  = nstrin1 + 1 ; #vett.#d = substr(cstrin,1,8)
          #d = #d + 1 ; #vett.#d = substr(cstrin,9,lstrin-8) ; end
       else #vett.#d = substr(cstrin,1,lstrin) ; end
       nstrin = nstrin + nstrin1
    if nstrin1 = Ø then lstrinok = no ; return
```

$DB2RC00 REXX EXEC

```
/* REXX */
   /*-------------------------------------------------------------------*/
   /*-                  Data Tool for DB management                     -*/
   /*-                   Return Code report file                        -*/
   /*-------------------------------------------------------------------*/
```

```
arg parmin ; parm = translate(parmin,' ',',')
nparm = words(parm)  ;subsys= word(parm,1) ;datab = word(parm,2)
jobnam= word(parm,3) ;esito = word(parm,4) ;rexxfrom = word(parm,5)
   /*-  Test input parameters                -*/
 if nparm < 5 then do
    say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Parameter string is incomplete !!!!'
    say '>>>>>>>>           'parmin
    say '>>>>>>>>' ;say '' ; say '' ; exit ; end
   /*-  Parameters assignment                -*/
 call @db2parØ subsys ;if word(result,1) = 99 then exit
 $lpar =word(result,1) ; $accn =word(result,2) ; $class =word(result,3)
 $msgcla = word(result,4);$region  = word(result,5) ;$msglvl =
word(result,6)
 $notif  = word(result,7);$user    = word(result,8) ;$unitda =
word(result,9)
 $unitta = word(result,1Ø);$esunit = word(result,11);$prt    =
word(result,12)
 $hiwork = word(result,13);$db2ver = word(result,14);$ctsubs =
word(result,15)
 $librexx= word(result,16);$parmlib= word(result,17);$proclib=
word(result,18)
 $jcllib = word(result,19);$report = word(result,2Ø);$libexec=
word(result,21)
 $isptenu= word(result,22);$isppenu= word(result,23);$ispmenu=
word(result,24)
 $ispslib= word(result,25);$plilink= word(result,26);$sibmlnk=
word(result,27)
 $sortlib= word(result,28);$hilvlDB= word(result,29);$runlib =
word(result,3Ø)
 $dsnload= word(result,31);$tep2pgm= word(result,32);$tep2pln=
word(result,33)
 $unlopgm= word(result,34);$unlopln= word(result,35);$dunlopg=
word(result,36)
 $dunlopl= word(result,37);$dsnproc= word(result,38)
   /*-   Work areas initialization           -*/
 blk = ;ctr = Ø
   /*-  CheckList file allocation            -*/
 say '' ; outdschk= $hiwork'.'subsys'.'datab'.'rexxfrom'.CHECKRC'
 jobw = fichk  ; "alloc da('"outdschk"') f("jobw") mod reuse"
 if esito = 'OK'  then
sk.1=' Job  'jobnam'   'esito'   RC = Ø '
 if esito = 'KO'  then
sk.1=' Job  'jobnam'   'esito'   RC > Ø  Verify output !!!!!! '
 if esito = 'BLK' then
sk.1='                                                             '
 sk.Ø=1 ; Call WriteRec ;Call Free ;exit
   /*-  Write record routine                 -*/
 WriteRec :
    "EXECIO * DISKW "jobw" (STEM sk. FINIS"
    if rc = 2Ø then do
```

```
       do while rc ¬= 'ØØ'
          if ctr > 24 then do
            say '' ; say '>>>>>>>>'
     say '>>>>>>>> 2 minutes are spent waiting Check file. End process'
            say '>>>>>>>>' ; say '' ; exit ; end
          ctr = ctr + 1 ; address tso "delayØØ ØØ5"
          "free fi(fichk)" ;jobw = fichk
          "alloc da('"outdschk"') f("jobw") mod reuse"
          "EXECIO * DISKW "jobw" (STEM sk. FINIS"
          end ; end
 ClearRec:
    DO f = 1 to sk.Ø
       sk.f = blk ; end ;return
   /*–  Free Work areas                      –*/
 Free    :
    xx=outtrap(trpdummy.) ;"free fi(fichk)" ;xx=outtrap(off);return
```

$DB2SDSF REXX EXEC

```
/* REXX */
   /*-------------------------------------------------------------*/
   /*-                 Data Tool for DB management               -*/
   /*-                     Display job Status                    -*/
   /*-------------------------------------------------------------*/
arg prmin ; parm = translate(prmin,' ',',') ; nparm  = words(parm)
subsys   = word(parm,1) ;jobcheck = word(parm,2)
delaytim = word(parm,3) ;timeexit = word(parm,4)
   /*–  Test input parameters                –*/
 if nparm < 4 then do
    say '' ;say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Parameter string is incomplete !!!!'
    say '>>>>>>>>           'parmin
    say '>>>>>>>>' ;say '' ;say '' ; $exitc = 99 ; return $exitc
    exit ; end
   /*–  Parameters assignment                –*/
 call @db2parØ subsys
 if word(result,1) = 99 then do
    $exitc = 99 ; return $exitc ; exit ; end
 $lpar =word(result,1) ; $accn =word(result,2) ; $class =word(result,3)
 $msgcla = word(result,4);$region  = word(result,5) ;$msglvl =
word(result,6)
 $notif  = word(result,7);$user    = word(result,8) ;$unitda =
word(result,9)
 $unitta = word(result,1Ø);$esunit = word(result,11);$prt    =
word(result,12)
 $hiwork = word(result,13);$db2ver = word(result,14);$ctsubs =
word(result,15)
 $librexx= word(result,16);$parmlib= word(result,17);$proclib=
word(result,18)
 $jcllib = word(result,19);$report = word(result,2Ø);$libexec=
word(result,21)
 $isptenu= word(result,22);$isppenu= word(result,23);$ispmenu=
word(result,24)
 $ispslib= word(result,25);$plilink= word(result,26);$sibmlnk=
word(result,27)
 $sortlib= word(result,28);$hilvlDB= word(result,29);$runlib =
word(result,3Ø)
 $dsnload= word(result,31);$tep2pgm= word(result,32);$tep2pln=
word(result,33)
 $unlopgm= word(result,34);$unlopln= word(result,35);$dunlopg=
word(result,36)
 $dunlopl= word(result,37);$dsnproc= word(result,38)
   /*-   Work areas initialization          -*/
 blk  =  ; wtimeexit = Ø
   /*-  IsfOUT file allocation              -*/
 outds1= $hiwork'.'subsys'.@DB2SDSF.ISFOUT'
 dsn = sysdsn(''''outds1'''')
 if dsn ¬= OK then do
    prmalloc = subsys' 'outds1' Ø 6Ø,3Ø f,b,a 133 Ø isfout si'
    call @db2allØ prmalloc
    if word(result,1) = 99 then do
       $exitc = 99 ; return $exitc ; exit ; end ; end
 else do
    jobw = isfout ; "alloc da('"outds1"') f("jobw") shr reuse" ; end
   /*-  IsfIN  file allocation              -*/
 outds2 = $hiwork'.'subsys'.@DB2SDSF.ISFIN'
 dsn = sysdsn(''''outds2'''')
 if dsn ¬= OK then do
    prmalloc = subsys' 'outds2' Ø 1,1 f,b 8Ø 2792Ø isfin si'
    call @db2allØ prmalloc
    if word(result,1) = 99 then do
       $exitc = 99 ;return $exitc ;exit ;end ; end
 else do
    jobw = isfin ; "alloc da('"outds2"') f("jobw") shr reuse" ; end
 jobw = isfin
 sk.1=' ST 'jobcheck
 sk.Ø = 1 ; call WriteRec ;call versdsf
 do while isfout.Ø ¬=  Ø
    wtimeexit = wtimeexit + 1
    say ''
   say '--> 'time()' N> 'right(isfout.Ø,3,'Ø')' job 'jobcheck' running'
    address tso "delayØØ "delaytim"" ; call versdsf
    if wtimeexit = timeexit then do
       $jobnum = isfout.Ø ;$exitc = 98 ;return $exitc $jobnum ;exit
       end ; end  ; call Free ; $exitc = ØØ ; return $exitc ; exit
   /*-  Write record routine                -*/
 WriteRec :
    "EXECIO * DISKW "jobw" (STEM sk. FINIS"
 ClearRec:
    DO #f = 1 to sk.Ø
       sk.#f = blk ; end ; return
   /*-  SDSF routine                          -*/
Versdsf :
    xx=outtrap(trpØØ.)
       "ispexec select pgm(SDSF) "
       if rc > Ø then do
          say '' ;say '>>>>>>'; say '>>>>>> Call SDSF Failed RC = 'rc
          say '>>>>>> Verify output. End elaboration' ;
          say '>>>>>>' ;say ''
          address tso "printoff ('"outds1"') class(X)"
          "free fi(isfin)" ;"free fi(isfout)" ; $exitc = 99
          return $exitc ; exit ;end ;xx=outtrap(off)
    xx=OUTTRAP(trpØ1.)
       "ispexec edit dataset('"outds1"') macro(@MDB2Ø39)"
    xx=OUTTRAP(OFF)
    if rc > Ø then do
       do #a = 1 to trpØ1.Ø
          say trpØ1.#a ; end ;$exitc = 99;return $exitc;exit ;end
    xx=OUTTRAP(trpØ2.)
       "alloc da('"outds1"') f(isfout) shr reuse"
       "execio * diskr isfout (stem isfout. finis"
    xx=OUTTRAP(OFF)
    if rc > Ø then do
       do #a = 1 to trpØ2.Ø
          say trpØ2.#a ; end
       say '' ; say ''
       say '>>>>>>>>  Error during allocation "'outds1'"'
       say '>>>>>>>>  RC='rc'. Verify.'
       say '' ;say '' ;$exitc = 99;return $exitc ;exit; end ;return
   /*-  Free Work areas                       -*/
 Free :
    xx=outtrap(trpdummy.)
       "free fi(isfin)" ;"free fi(isfout)"
    xx=outtrap(off) ;return
```

$MDB2025 EDIT MACRO

```
/* REXX */
trace ?o
/*-----------------------------------------*/
/*--    Used in REXX @DB2FREE           --*/
/*-----------------------------------------*/
isredit macro
isredit exclude all
isredit find '''_|''' all
isredit find ''' Ø ROW'''
isredit delete x all
isredit change P'''====='''  '''      ''' 59 all
isredit change '''|'''  ''' '''  all
isredit save
isredit end
```

$MDB2026 EDIT MACRO

```
/* REXX */
trace ?o
/*----------------------------------------*/
/*——   Used in Rexx @DB2FREE           ——*/
/*----------------------------------------*/
isredit macro
isredit exclude all
isredit find '''_|''' all
isredit find ''' Ø ROW'''
isredit delete x all
isredit change P'''====''' '''      ''' 15 all
isredit change '''|'''  ''' '''  all
isredit save
isredit end
```

$MDB2039 EDIT MACRO

```
/* REXX */
trace ?o
/*-----------------------------------------*/
/*——    Used in REXX @DB2SDSF            ——*/
/*-----------------------------------------*/
isredit macro
isredit exclude all
isredit find EXECUTION  all
isredit delete x all
isredit save
isredit exclude HOLD    all
isredit delete x all
isredit save
isredit end
```

$MDB2040 EDIT MACRO

```
/* REXX */
trace ?o
address ispexec 'VGET ($vdatas)  PROFILE'
address ispexec 'VGET ($vcreunl) PROFILE'
address ispexec 'VGET ($vcreloa) PROFILE'
address ispexec 'VGET ($vesunit) PROFILE'
address ispexec 'VGET ($vreusVS) PROFILE'
/*-----------------------------------------*/
/*——    Used in REXX @DB2UNØ1            ——*/
/*-----------------------------------------*/
isredit macro
if $vreusVS = no then
isredit change '"RESUME YES"' '"REPLACE SORTDEVT '$vesunit' SORTNUM 4"'
else
   isredit change '"RESUME YES"' '"REPLACE REUSE SORTDEVT '$vesunit'
```

```
SORTNUM 4"'
call VerRc 'Chage RESUME YES'
if $vcreunl ¬= $vcreloa then do
   isredit change ''$vcreunl'' ''$vcreloa'' all
   call VerRc 'Change CREUNL/CRELOA'
   end
isredit save
isredit end
exit
VerRc :
   arg parmRc
   if rc > 4 then do
   say ''
   say '    +--------------------------------------------------------+'
   say '    |    !!!!!!!!!!!!!!   W A R N I N G   !!!!!!!!!!!!!!    |'
   say '    |               Cange Function Error                    |'
   say '    |           Return Code 'rc' in @MDB2Ø4Ø Macro          |'
   say '    |               Error during       'left(Parmrc,2Ø,' ')'|'
   say '    |                         'center($vdatas,4Ø)'          |'
   say '    |       Load procedure will be stopped.                 |'
   say '    +--------------------------------------------------------+'
   say ''
      exit
      end
   if rc = 4 then do
  say ''
  say '    +---------------------------------------------------------+'
  say '    |  !!!!!!!!!!!!!!     W A R N I N G     !!!!!!!!!!!!!!   |'
  say '    |     Il 'center(Parmrc,2Ø)' do not run                   |'
  say '    |        'center($vdatas,4Ø)'                             |'
  say '    +---------------------------------------------------------+'
  say ''
      end
   return
```

$DB2UN00 SAMPLE JOB

```
//&SYSUID.$ JOB (ØØØØØØØØ),'DB2-Management',CLASS=S,MSGCLASS=X,
//       REGION=3M,MSGLEVEL=(1,1),NOTIFY=&SYSUID
/*JOBPARM BYTES=999999,LINES=999999
//* ________________________________________________________ *
//DB2PROC   JCLLIB ORDER=(user.library)
//JOBLIB    DD  DSN=SYS1.DSN51Ø.SDSNLOAD,DISP=SHR
//* ________________________________________________________ *
//DELVIEW  EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø
//SYSTSPRT  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SYSTSIN   DD  *
 DSN SYSTEM(DSNZ)
 RUN PROGRAM(DSNTEP2) PLAN(DSNTEP51) LIB('DSN51Ø.RUNLIB.LOAD')
//SYSIN     DD  *
```

```
  DROP VIEW VPRD1ØZ_UNLOAD ;
//* _______________________________________________________ *
//LABØ     IF (DELVIEW.RC LT 9) THEN
//CREVIEW  EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø
//SYSTSPRT  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SYSTSIN   DD  *
 DSN SYSTEM(DSNZ)
 RUN PROGRAM(DSNTEP2) PLAN(DSNTEP51) LIB('DSN51Ø.RUNLIB.LOAD')
//SYSIN     DD  *
CREATE VIEW VPRD1ØZ_UNLOAD
  AS SELECT A.CREATOR,A.NAME,A.CARD,A.RECLENGTH,A.STATSTIME,
            C.NTABLES,A.TSNAME,D.VCATNAME,C.PARTITIONS,
            A.ENCODING_SCHEME,C.DBID,C.PSID,A.OBID,A.CREATEDTS,
            A.ALTEREDTS
      FROM  SYSALT.SYSTABLES     A ,
            SYSALT.SYSDATABASE   B ,
            SYSALT.SYSTABLESPACE C ,
            SYSALT.SYSSTOGROUP   D
            WHERE  B.NAME    = 'PRD1ØZ'   AND
                   A.CREATOR = 'PRA1ØZ'   AND
                   A.TYPE    = 'T'        AND
                   A.TSNAME  =  C.NAME    AND
                   A.CREATOR =  C.CREATOR AND
                   B.STGROUP =  D.NAME    AND
                   A.DBID    =  B.DBID      ;
//* _______________________________________________________ *
//*       (A.TSNAME LIKE 'OF%'    OR
//*        A.TSNAME LIKE 'XC%' ) AND
//* _______________________________________________________ *
//*       (A.TSNAME LIKE 'OF%'    OR
//*        A.TSNAME LIKE 'XC%' ) AND
//* _______________________________________________________ *
//*        A.TSNAME NOT IN ('                  ',
//*                        '                  ')  AND
//* _______________________________________________________ *
//LAB1      IF (CREVIEW.RC EQ Ø) THEN
//REXXØØ   EXEC   DB2REXX1
//* _______________________________________________________ *
//* ___   Variable Description    Value     Default     ___ *
//* _______________________________________________________ *
//* ___   DB2 Subsystem _______:  DSNZ      no/default  ___ *
//* ___   Database ____________:  PRD1ØZ    no/default  ___ *
//* ___   Creator unlo table___:  PRA1ØZ    *           ___ *
//* ___   DB2 Load subsystem __:  *         *=subsys    ___ *
//* ___   DB2 load Database ___:  PRD11Z    *=Database  ___ *
//* ___   DB2 load creator  ___:  PRA11Z    *=Creunl    ___ *
//* ___   Reuse DB2 Vsam ______:  *         *=yes       ___ *
//* ___   Bind owner __________:  *         *=no Bind   ___ *
//* ___   Full unload__________:  yes       *=no        ___ *
//* ___   Maxdd _______________:  8Ø        *=8Ø        ___ *
//* ___   Ritenzione Dump _____:  m12       *=mØ1       ___ *
//* ___   Submit job __________:  *         *=no        ___ *
//* ___   Nome job ____________:  *         *=dbnameyy  ___ *
//* ___   Catalog Creator _____:  sysalt    *=sysibm    ___ *
//* _________________________________________________________ *
//REXXØØ.SYSTSIN  DD  *
 ISPSTART CMD(@DB2UNØØ +
 DSNZ,PRD1ØZ,PRA1ØZ,*,PRD11Z,PRA11Z,*,*,*,8Ø,m12,*,*,sysalt)
//LABØEND  ENDIF
//LAB1END  ENDIF
//* _________________________________________________________ *
```

The code for the following is contained in *DB2 Update*, Issue 106, August 2001, *Imagecopy generator procedure*:

- $DB2PAR0 REXX EXEC
- $DB2ALL0 REXX EXEC
- $DB2FOR0 REXX EXEC
- $DB2OUBK REXX EXEC
- $MDB2018 EDIT MACRO
- $MDB2031 EDIT MACRO
- DB2REXX1 PROC
- IEBDDIN DD for SYSIN

The sample Assembler program, DELAY00, for the TSO delay time has been taken from *MVS Update*, Issue 115, April 1996, which can be downloaded from http://www.xephon.com/archives/m115a03.txt.

The code for the following is contained in *DB2 Update*, Issues 73, 74, and 75, November and December 1998 and January 1999, *Plan and package management*:

- $DB2BIND REXX EXEC
- $MDB2021 EDIT MACRO
- $MDB2023 EDIT MACRO
- $MDB2024 EDIT MACRO
- $MDB2027 EDIT MACRO.

Giuseppe Rendano
DB2 Systems Programmer (Italy)

DB2 news

Rogue Wave Software has increased support for new platform and database clients with the introduction of the third version of Rogue Wave SourcePro C++.

The new edition is now available on Windows XP Professional, Red Hat Linux 7.1, AIX 5L Version 5.1, and HP-UX 11i.

SourcePro C++ Edition 3 now provides support to recent database releases, including DB2 7.2, Oracle9*i*, Sybase 12.5, and Informix 2.7.

It includes support for the latest XML schema specification and for the SOAP 1.2 specification.

For further information contact:
Rogue Wave Software, 5500 Flatiron Pkwy, Boulder, CO 80301, USA.
Tel: (303) 473 9118.
URL: http://www.roguewave.com/products/sourcepro/.

* * *

IBM's WebSphere Application Server and DB2 database have been certified compliant with J2EE 1.3, which incorporates Java Messaging Service, integrates with XML, and supports standardized integration of CORBA and Java applications.

IBM has also announced that DB2 UDB Workgroup and Enterprise Editions for Linux (Intel) V7.2 are available now only through its Passport Advantage programme.

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

* * *

Computer Associates has announced the latest releases of its Advantage InfoRefiner and Advantage InfoTransport data transformation and migration tools, which help transform, integrate, and migrate large volumes of server data to leading mainframe and client/server relational databases without, it's claimed, the need for programming.

Included within Advantage InfoRefiner, Advantage InfoTransport 3.3 supports data migration and change propagation to operating systems and relational databases including DB2 UDB 7.1, Windows 2000, Oracle 8.17 and 9*i*, and Sybase 11.*x*.

In addition, CA has begun shipping Advantage CA-SymDump Batch 2.0 and Advantage CA-Optimizer/II 3.0, its z/OS and OS/390 fault management solutions for helping to find and correct application problems.

Both get expanded batch abend diagnostic capabilities, including a central repository that stores abend reports in a central location, to help zoom in on relevant diagnostic details and simplify analysis of information about application failures.

Batch abend diagnostic capabilities include detailed diagnostic information for applications that access DB2, IMS, and Advantage CA-IDMS/DB databases.

For further information contact:
Computer Associates International, One Computer Associates Plaza, Islandia, NY 11749, USA.
Tel: (631) 342 6000.
URL: http://www3.ca.com/Solutions/Collateral.asp?ID=1229&PID=1011.