148

DB2

update

February 2005

In this issue

A xephon PUBLICATION

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

Phone: 214-340-5690
Fax: 214-341-7081

Editor

Trevor Eddolls
E-mail: trevore@xephon.com

Publisher

Colin Smith
E-mail: info@xephon.com

Subscriptions and back-issues

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

DB2 Update on-line

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

Contributions

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

Printed in England.

DB2 UDB for LUW 8.2 – what's new in the db2diag.log

With DB2 UDB for LUW FP7 (V8.2) there are many improvements to the db2diag.log file. These improvements are in the contents of the file and how information in the file can be viewed. I will explain both of these. I ran all the commands/ SQL in this article on a Windows 2000 machine running DB2 UDB 8.2 using the db2admin userid.

Let's first look at the contents of the db2diag.log file. A really nice new feature is that DB2 now logs changes to the database (DB) and database manager (DBM) configuration files. This is really useful when someone asks, "What has changed since last week?". The command is fully documented in the *Command Reference* manual (SC09-4828-01). Help is available in the normal manner (>db2diag –h).The Help file shows:

- filename – one or more space-separated path names of diagnostic logs.
- -filter, -g – case-sensitive search for a list of field-pattern pairs.
- -gi – case-insensitive search for a list of field-pattern pairs.
- -gv – case-sensitive invert matching.
- -gvi, -giv – case-insensitive invert matching.
- -invert , -v – invert the sense of matching for all filtering options.
- -exist – record field must exist in order to be processed.
- -pid – find all records for a list of process IDs.
- -tid – find all records for a list of thread IDs.
- -node, -n – find all records for a list of nodes.
- -error, -e – find all records for a list of errors.

- -history, -H – display the history of logged records for a time interval.
- -level, -l – find all records for a list of severity levels.
- -time, -t – display all the records within a particular time interval.
- -count, -c – display a count of matching records.
- -verbose, -V – display all record fields whether they contain data or not.
- -strict – display records using one ‘field: value’ pair per line.
- -cbe – display records in the Common Base Event (CBE) format.
- -fmt – format tool’s output using a format string.
- -output, -o – save output into a file.
- -follow, -f – continuously display appended records as the file grows.
- -archive, -A – archive a diagnostic log file.
- -rc – display descriptions of DB2 error return codes, ZRC or ECF.

Say you wanted to get some more information on the **gi** option – you then issue:

```
>db2diag -h gi
```

And if you wanted to see all the help in the most detailed form you would issue:

```
>db2diag -h all
```

So let’s see how it works. Let’s update a DB CFG entry for the SAMPLE database and one for the DB2 instance. Updating the DB CFG for the SAMPLE database:

```
>db2 update db cfg for sample using vendoropt fred
```

If we now look in db2diag.log we see:

```
FUNCTION: DB2 UDB, config/install, sqlfLogUpdateCfgParam, probe:20
CHANGE  : CFG DB SAMPLE: "Vendoropt" From: "" To: "fred"
```

And if we update a database manager value for the DB2 instance:

```
>db2 update dbm cfg using maxagents 401
```

Looking in db2diag.log we see:

```
FUNCTION: DB2 UDB, config/install, sqlfLogUpdateCfgParam, probe:30
CHANGE  : CFG DBM: "Maxagents" From: "400" To: "401"
```

Rather than trawling through the db2diag.log file we can use a command (**db2diag -level event**) to list any changes. This command lists all events:

```
>db2diag -level event

.....
FUNCTION: DB2 UDB, config/install, sqlfLogUpdateCfgParam, probe:20
CHANGE  : CFG DB SAMPLE: "Vendoropt" From: "" To: "fred"
.....
FUNCTION: DB2 UDB, config/install, sqlfLogUpdateCfgParam, probe:30
CHANGE  : CFG DBM: "Maxagents" From: "400" To: "401"
```

(There is some header information that I do not show.)

To list all the times that the Database Manager has started you could issue:

```
>db2diag -level event | find /i "ADM7513W"

MESSAGE : ADM7513W  Database manager has started.
```

Because the message block we are interested in is:

```
yyyy-mm-dd-hh.mm.ss.tttttt+060 xxxxxxxxxxx        LEVEL: Event
PID     : 1368                 TID  : 1392        PROC : db2syscs.exe
INSTANCE: DB2                  NODE : 000
FUNCTION: DB2 UDB, base sys utilities, DB2StartMain, probe:911
MESSAGE : ADM7513W  Database manager has started.
START   : DB2 DBM
DATA #1 : Build Level, 124 bytes
Instance "DB2" uses "32" bits and DB2 code release "SQL08020"
with level identifier "03010106".
Informational tokens are "DB2 v8.1.7.445", "s040812", "WR21342", FixPak
"7".
```

Unfortunately, the timestamp for the ADM7513W message is at the top of the block, so isn't shown when you use the **find** command shown above.

Let's look at some other examples.

If we want to see what has changed for database SAMPLE, we would issue:

```
>db2diag -gi changeev= -g db=SAMPLE

yyyy-mm-dd-hh.mm.ss.tttttt+060 xxxxxxxxxxx        LEVEL: Event
PID     : 1540                 TID  : 1712        PROC : db2syscs.exe
INSTANCE: DB2                  NODE : 000         DB   : SAMPLE
APPHDL  : 0-25                 APPID: *LOCAL.DB2.041009180714
FUNCTION: DB2 UDB, config/install, sqlfLogUpdateCfgParam, probe:20
CHANGE  : CFG DB SAMPLE: "Vendoropt" From: "" To: "fred"
```

Note that this output is different from the **>db2diag -level event** command because this one shows only the changes for the SAMPLE database.

If we wanted to see everything that has happened to instance DB2 we would issue:

```
>db2diag -gi changeev= -g inst=DB2
```

This will show us the changes to the instance DB2 as well as the databases in the instance. Remember for a full list of what I can search for issue **>db2diag -h gi**.

A useful option is to list the db2diag.log entries for a particular time period. For full help on this, issue **>db2diag -h t**. But here is an example that I found particularly useful. To display all the messages for one day (say 15 October 2004) to the screen issue:

```
>db2diag -t 2004-10-15-00.00.00:2004-10-15-23.59.59
```

The format for the timestamp is yyyy-mm-dd-hh.mm.ss.nnnnn and the timestamps are separated by a colon. You have to enter the whole string – this is especially true (and annoying!) for the second timestamp. You cannot enter **>db2diag -t 2004-10-15-00.00.00:23.59.59** – if you do you will get a 'db2diag: Invalid year specified in argument string' message back.

You can of course direct this information to a file using the **-o** parameter as shown in the command below:

```
>db2diag -t 2004-10-15-00.00.00:2004-10-15-23.59.59 -o
c:\temp\log20041015.txt
```

We can combine two of the above searches – those for change event and time – to give a command that will list all the changes to the SAMPLE database since a certain timestamp (15 October 2004 at 17:06):

```
>db2diag -gi changeev= -g db=SAMPLE -t 2004-10-15-17.06.00
```

You can see how you can build up the options to produce powerful commands.

Another useful command will archive the db2diag.log file. Issue:

```
>db2diag -A

db2diag: Moving "C:\PROGRA~1\IBM\SQLLIB\DB2\db2diag.log"
         to "C:\PROGRA~1\IBM\SQLLIB\DB2\db2diag.log_2004-10-16-00.44.53"
```

You can see that the command has created a file called db2diag.log and appended to the filename the timestamp when the command was issued.

We now have two ways of creating a 'daily' db2diag.log file – use the first method if you want to preserve the original file; use the second method if you are not concerned about preserving the file. Of course, preserving the file makes searches easier, but you should have a db2diag.log archive strategy in place, otherwise the file will become too difficult to manage.

There is a lot that is new in managing and reporting from the db2diag.log. It should make tracking changes and using the file a lot more user friendly – give some of the commands a go.

C Leonard
Freelance Consultant (UK)

Net.Data table query builder

“Net.Data is a good choice for creating dynamic Web pages because using the macro language is simpler than writing your own Web server applications and because Net.Data lets you use languages that you already know, such as HTML, SQL, Perl, REXX, and JavaScript.” So says the *IBM Net.Data for OS/2, Windows NT, and Unix Administration and Programming Guide Version 7*, Chapter 1, *Introduction: Why Use Net.Data?* on Page 2.

Keeping in mind the text above, I have developed a Net.Data macro that enables the dynamic creation of queries simply by selecting the appropriate values for the table columns. Besides query creation and presentation of its result, it is also possible to present the contents of the columns that otherwise are not included in the query results (BLOB, CLOB, DB2IMAGE, DB2VIDEO, DB2AUDIO, or XML objects).

The macro functions in the following way.

First, you must choose one of the databases listed in the appropriate combo-box, which is populated with DB2 data source names from ODBC data sources on a database server. For the chosen database, enter the correct user-id and password.

Secondly, select the DB2 table. Choose values from the combo-box with distinct values from the appropriate column. Mark the check-box on the left of the desired column value and click on the *Filter* button to generate the filter for the query. Eventually, change the *Where* clause in the text area box and submit your query with the *Show Query Result* push-button.

Combo-boxes with grey backgrounds contain all BLOB, CLOB, DB2IMAGE, DB2VIDEO, DB2AUDIO, or XML objects. These objects are not included in query builder, but, when selected, they present the content. (Of course, for access to DB2IMAGE, DB2VIDEO and DB2AUDIO columns, AIV extenders must be started.)

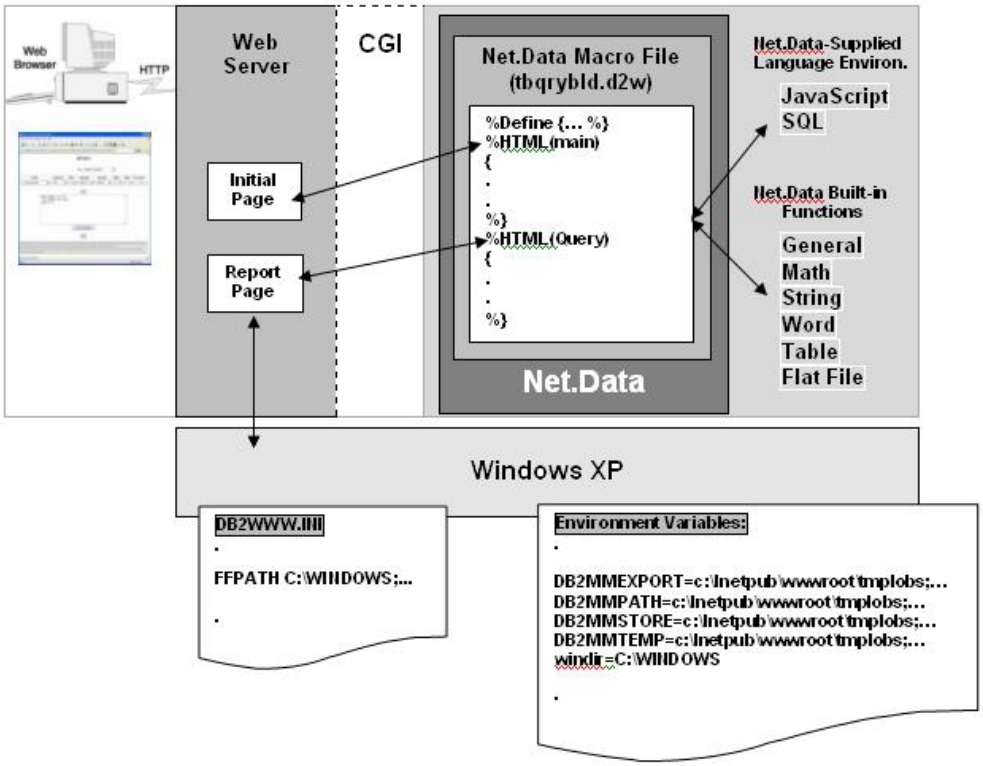

Figure 1: How to configure Net.Data

The configuration tasks include:

- Configuring Net.Data for CGI.
- Customizing the Net.Data initialization file (DB2WWW.INI):

```
FFI_PATH       C:\WINDOWS; C:\DB2WWW\Macro
```

- Customizing the Windows environment variables:

```
DB2MMEXPORT=c:\Inetpub\wwwroot\tmplobs;...
DB2MMPATH=c:\Inetpub\wwwroot\tmplobs;...
DB2MMSTORE=c:\Inetpub\wwwroot\tmplobs;...
DB2MMTEMP=c:\Inetpub\wwwroot\tmplobs;...
windir=C:\WINDOWS
```

  The default tmplobs directory for LOBs is under the directory specified by the HTML_PATH in the shipped Net.Data initialization file. It is accessible by any user ID. If the HTML_PATH value is changed, ensure that the user

ID under which the Web server is running has write access to the directory specified by HTML_PATH.

Figure 1 describes how to configure Net.Data and how to modify your configuration of the Web server for use with this Net.Data macro.

A macro needs to be invoked with the following URL from your browser: http://hostname/cgi-bin/db2www.exe/tbqrybld.d2w/main.

Note: all parameters in the text above should be tailored according to your environment.

TBQRYBLD.D2W

```
%{Macro: tbqrybld.d2w -- Net.Data TABLE QUERY BUILDER
    SHOWSQL="YES"
%}

%Define {
   WINDIR=%ENVVAR
   myFile = "$(WINDIR)\ODBC.INI"
   dbTable = %TABLE
   myTable = %TABLE
   valTable = %TABLE
   mySearch = "IBM DB2 ODBC DRIVER"
   DATABASE = ""
   LOGIN = ""
   PASSWORD = ""
   DTW_DEFAULT_REPORT = "NO"
   DTW_INI = %ENVVAR
   tb1 = ""
   tb2 = ""
   fnd = ""
   numRows = "Ø"
   selcol = ""
%}

%FUNCTION(DTW_SQL) selTables() {
   SELECT CONCAT(CONCAT(CREATOR, '.'), NAME)
   FROM SYSIBM.SYSTABLES
   ORDER BY 1
   FOR FETCH ONLY
   %MESSAGE {
        1ØØ   : "No rows were returned by the query" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
```

```
                ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                macro file. %} : exit
      -14Ø2   : {Error $(RETURN_CODE) occurred.  Make sure your
                DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
   %REPORT {
      %ROW {
         %IF ($(ROW_NUM) == "1")
           @DTW_ASSIGN(tb2, "$(V1)")
           @DTW_ASSIGN(fnd, "Ø")
         %ENDIF
         %IF ("$(V1)" == "$(tb1)")
           @DTW_ASSIGN(fnd, "1")
           <option selected value="$(V1)">$(V1)</option>
         %ELSE
           <option value="$(V1)">$(V1)</option>
         %ENDIF
      %}
      %IF ($(fnd) == "Ø")
        @DTW_ASSIGN(tb1, "$(tb2)")
      %ENDIF
   %}
%}

%FUNCTION(DTW_SQL) selColumns(IN tb, OUT T1) {
   SELECT *
   FROM SYSIBM.SYSCOLUMNS
   WHERE TBCREATOR = '@DTW_rSUBSTR("$(tb)", 1,
@DTW_rSUBTRACT(@DTW_rPOS(".", "$(tb)"), "1"))' AND
         TBNAME = '@DTW_rSUBSTR("$(tb)", @DTW_rADD(@DTW_rPOS(".",
"$(tb)"), "1"))'
   ORDER BY COLNO
   FOR FETCH ONLY
   %MESSAGE {
        1ØØ   : "No rows were returned by the query" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                 ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                 macro file. %} : exit
      -14Ø2   : {Error $(RETURN_CODE) occurred.  Make sure your
                DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
%}

%FUNCTION(DTW_SQL) selValues(IN tbname, colname, OUT T1) {
   SELECT DISTINCT "$(colname)"
   FROM $(tbname)
```

```
   FOR FETCH ONLY
   %MESSAGE {
        100   : "" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                 ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                 macro file. %} : exit
      -1402   : {Error $(RETURN_CODE) occurred.  Make sure your
                DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
%}

%FUNCTION(DTW_SQL) selImgValues(IN tbname, colname) {
   SELECT cast(mmdbsys.content($(colname), 'GIF') as blob(150000)),
          mmdbsys.format($(colname)),
          mmdbsys.comment($(colname))
   FROM $(tbname)
   FOR FETCH ONLY
   %MESSAGE {
        100   : "" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                 ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                 macro file. %} : exit
      -1402   : {Error $(RETURN_CODE) occurred.  Make sure your
                DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
   %REPORT {
        <option value=""></option>
      %ROW {
        <option value=$(V1)>$(V3)</option>
      %}
   %}
%}

%FUNCTION(DTW_SQL) selVidValues(IN tbname, colname) {
   SELECT mmdbsys.comment($(colname)),
          mmdbsys.content($(colname), mmdbsys.comment($(colname)), 1),
          mmdbsys.format($(colname))
   FROM $(tbname)
   FOR FETCH ONLY
   %MESSAGE {
        100   : "" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                 ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                 macro file. %} : exit
      -1402   : {Error $(RETURN_CODE) occurred.  Make sure your
                DB2 security service is installed and running. %} : exit
```

```
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
   %REPORT {
        <option value=""></option>
      %ROW {
        <option value=$(V1)>$(V1)</option>
      %}
   %}
%}

%FUNCTION(DTW_SQL) selAudValues(IN tbname, colname) {
   SELECT mmdbsys.comment($(colname)),
          mmdbsys.content($(colname), mmdbsys.comment($(colname)), 1),
          mmdbsys.format($(colname))
   FROM $(tbname)
   FOR FETCH ONLY
   %MESSAGE {
        100   : "" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                 ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                 macro file. %} : exit
      -1402   : {Error $(RETURN_CODE) occurred.  Make sure your
                DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
   %REPORT {
        <option value=""></option>
      %ROW {
        <option value=$(V1)>$(V1)</option>
      %}
   %}
%}

%FUNCTION(DTW_SQL) selClob(IN tbname, colname) {
   SELECT $(colname)
   FROM $(tbname)
   FOR FETCH ONLY
   %MESSAGE {
        100   : "" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                 ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                 macro file. %} : exit
      -1402   : {Error $(RETURN_CODE) occurred.  Make sure your
                DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
   %REPORT {
```

```
        <option value=""></option>
      %ROW {
        <option value=$(V1)>@DTW_rSUBSTR(V1, @DTW_rADD(@DTW_rLASTPOS("/
", V1), "1"))</option>
      %}
   %}
%}

%FUNCTION(DTW_SQL) selBlob(IN tbname, colname) {
   SELECT $(colname)
   FROM $(tbname)
   FOR FETCH ONLY
   %MESSAGE {
        100   : "" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                 ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                 macro file. %} : exit
      -1402   : {Error $(RETURN_CODE) occurred.  Make sure your
                DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
   %REPORT {
        <option value=""></option>
      %ROW {
        <option value=$(V1)>@DTW_rSUBSTR(V1, @DTW_rADD(@DTW_rLASTPOS("/
", V1), "1"))</option>
      %}
   %}
%}

%FUNCTION(DTW_SQL) selXmlClob(IN tbname, colname) {
   SELECT DB2XML.CLOB($(colname))
   FROM $(tbname)
   FOR FETCH ONLY
   %MESSAGE {
        100   : "" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                 ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                 macro file. %} : exit
      -1402   : {Error $(RETURN_CODE) occurred.  Make sure your
                DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
   %REPORT {
        <option value=""></option>
      %ROW {
        <option value=$(V1)>@DTW_rSUBSTR(V1, @DTW_rADD(@DTW_rLASTPOS("/
", V1), "1"))</option>
```

```
      %}
   %}
%}

%FUNCTION(DTW_SQL) selXml(IN tbname, colname) {
   SELECT DB2XML.Varchartovarchar($(colname))
   FROM $(tbname)
   FOR FETCH ONLY
   %MESSAGE {
        100   : "" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                 ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                 macro file. %} : exit
      -1402   : {Error $(RETURN_CODE) occurred.  Make sure your
                DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
   %REPORT {
      @DTW_ASSIGN(no, "0")
      <option value=""></option>
      %ROW {
        @DTW_ADD(no, "1", no)
        @selXml1(tbname, colname, ROW_NUM, no)
      %}
   %}
%}

%FUNCTION(DTW_SQL) selXml1(IN tbname, colname, rown, no) {
   SELECT DB2XML.Content($(colname), '$(DTW_INI)/tmplobs/xml$(no).xml')
   FROM $(tbname)
   FOR FETCH ONLY
   %MESSAGE {
        100   : "" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                 ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                 macro file. %} : exit
      -1402   : {Error $(RETURN_CODE) occurred.  Make sure your
                DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
   %REPORT {
      %ROW {
        %IF (ROW_NUM == rown)
          <option value=$(V1)>@DTW_rSUBSTR(V1,
@DTW_rADD(@DTW_rLASTPOS("/", V1), "1"))</option>
        %ENDIF
      %}
   %}
```

```
%}

%FUNCTION(DTW_SQL) selQuery1(IN tbname) {
   SELECT NAME, COLNO
   FROM SYSIBM.SYSCOLUMNS
   WHERE TBCREATOR = '@DTW_rSUBSTR("$(tbname)", 1,
@DTW_rSUBTRACT(@DTW_rPOS(".", "$(tbname)"), "1"))' AND
         TBNAME = '@DTW_rSUBSTR("$(tbname)", @DTW_rADD(@DTW_rPOS(".",
"$(tbname)"), "1"))' AND
         COLTYPE NOT IN ('BLOB', 'CLOB', 'ROWID', 'DISTINCT')
   ORDER BY COLNO
   FOR FETCH ONLY
   %MESSAGE {
        100   : "No rows were returned by the query" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                  ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                  macro file. %} : exit
      -1402   : {Error $(RETURN_CODE) occurred.  Make sure your
                 DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
   %REPORT {
      %ROW {
        %IF ($(ROW_NUM) == "1")
          @DTW_ASSIGN(selcol, """$(V1)""")
        %ELSE
          @DTW_ASSIGN(selcol, "$(selcol), ""$(V1)""")
        %ENDIF
      %}
   %}
%}

%FUNCTION(DTW_SQL) selQuery2(IN sellist, tbname, selection, OUT T1) {
   SELECT $(sellist)
   FROM $(tbname)
   $(selection)
   FOR FETCH ONLY
   %MESSAGE {
        100   : "No rows were returned by the query" : continue
       -567   : {Error $(RETURN_CODE) occurred - Invalid authorization
                  ID.  Try specifying LOGIN and PASSWORD in the Net.Data
                  macro file. %} : exit
      -1402   : {Error $(RETURN_CODE) occurred.  Make sure your
                 DB2 security service is installed and running. %} : exit
      +default: "Warning $(RETURN_CODE)" : continue
      -default: "Unexpected SQL Error $(RETURN_CODE)" : exit
   %}
%}
```

```
%HTML(main) {
<html>
  <head>
    <meta http-equiv="content-type"
                              content="text/html;charset=iso-8859-1" />
    <title>Net.Data TABLE QUERY BUILDER</title>
    <script language="JavaScript"> <!--
       function validateform(thisform) {
         var selInd = thisform.db.selectedIndex;
         if (selInd < Ø) {
           alert("*** Please select a database ***");
           thisform.db.focus();
           return (false);
         }
         if (thisform.usrid.value == "") {
           alert("Please enter a value for the \"UserId\" field.");
           thisform.usrid.focus();
           return (false);
         }
         if (thisform.pswd.value == "") {
           alert("Please enter a value for the \"Password\" field.");
           thisform.pswd.focus();
           return (false);
         }
         var text = thisform.pswd.value;
         var backwards = "";
         for (count = text.length; count >= Ø; count--)
           backwards += text.substring(count, count - 1);
         text = "";
         for (count = Ø; count < backwards.length; count++)
           text = text + backwards.charCodeAt(count).toString(16);
         thisform.pswd.value = text.toUpperCase();
         return (true);
       }
    // --> </script>
  </head>
  <body bgcolor="white">
    <form method="post" action="proctab" onsubmit="return
validateform(this)" name="main">
       <center><h2>Net.Data TABLE QUERY BUILDER</h2>
       <BR><BR>
       @DTWF_SEARCH(myFile, "ASCIITEXT", "", myTable, mySearch)
       @DTW_TB_DELETECOL(myTable, "1", "2")
       @DTW_TB_SETCOLS(dbTable, "1")
       @DTW_ASSIGN(i, "1")
       %WHILE (i <= @DTW_TB_rROWS(myTable)) {
         @DTW_TB_INSERTROW(dbTable, @DTW_rSUBTRACT(i, "1"), "1")
         @DTW_TB_SETV(dbTable, @DTW_rDELSTR(@DTW_TB_rGETV(myTable, i,
"1"), @DTW_rPOS("=", @DTW_TB_rGETV(myTable, i, "1"))), i, "1")
         @dtw_add(i, "1", i)
```

```
        %}
        @DTW_ASSIGN(i, "1")
        @DTW_ASSIGN(j, "1")
        @DTW_ASSIGN(tmp, "")
        %WHILE (i <= @DTW_rSUBTRACT(@DTW_TB_rROWS(dbTable), "1")) {
          @dtw_add(i, "1", j)
          %WHILE (j <= @DTW_TB_rROWS(dbTable)) {
   %IF (@DTW_TB_rGETV(dbTable, i, "1") > @DTW_TB_rGETV(dbTable, j, "1"))
               @DTW_ASSIGN(tmp, @DTW_TB_rGETV(dbTable, i, "1"))
             @DTW_TB_SETV(dbTable, @DTW_TB_rGETV(dbTable, j, "1"), i, "1")
               @DTW_TB_SETV(dbTable, tmp, j, "1")
             %ENDIF
             @dtw_add(j, "1", j)
          %}
          @dtw_add(i, "1", i)
        %}
        <table border="0" cellpadding="0" cellspacing="0" width="30%">
          <tr>
            <td width="50%">Database:</td>
             <td width="50%">@DTW_TB_SELECT(dbTable,"db","1","1","N")</td>
          </tr>
          <tr>
            <td width="50%">UserId:</td>
            <td width="50%"><input type="text" name="usrid" size="10"
value="db2admin" maxlength="30"></td>
          </tr>
          <tr>
            <td width="50%">Password:</td>
            <td width="50%"><input type="password" name="pswd" size="11"
value="" maxlength="30"></td>
          </tr>
          <tr>
            <td width="50%"> </td>
            <td width="50%"> </td>
          </tr>
          <tr>
            <td width="50%" align="center"><input type="submit"
value="Submit" name="B11"></td>
            <td width="50%" align="left"><input type="reset"
value="Reset" name="B21"></td>
          </tr>
        </table>
        %include "help1.hti"
        </center>
    </form>
  </body>
</html>
%}

%HTML(proctab) {
```

```
<html>
  <head>
    <meta http-equiv="content-type"
                              content="text/html;charset=iso-8859-1" />
    <title>$(db)</title>
    <script language="JavaScript"> <!--
       function setparm(thisform) {
         var selInd = thisform.tb.selectedIndex;
         if (selInd < Ø) {
           alert("*** Please select a table ***");
           thisform.tb.focus();
           return false;
         }
         var work = thisform.tb.options[selInd].value;
    location.href="proctab?tb1="+work+"&db=$(db)&usrid=$(usrid)&pswd=$(pswd)";
         return true;
       }
       function fillfilter(theForm) {
         var checkSelected = false;
         var i;
         var j = Ø;
         var op = " = ";
         var apost = "'";
         theForm.filter.value = "";
         for (i = Ø;  i < theForm.check.length;  i++) {
           if (theForm.check[i].checked) {
             var coln = theForm.hidname[i].value;
             if (theForm.hidtype[i].value == "CHAR" ||
                 theForm.hidtype[i].value == "VARCHAR" ||
                 theForm.hidtype[i].value == "DATE" ||
                 theForm.hidtype[i].value == "TIME" ||
                 theForm.hidtype[i].value == "TIMESTAMP") apost = "'";
             else apost = "";
             j++;
    if (j > 1) theForm.filter.value = theForm.filter.value + "\n  AND ";
             else theForm.filter.value = "WHERE ";
             if (theForm.colval[i].value == "") theForm.filter.value =
theForm.filter.value + coln + " IS NULL";
             else theForm.filter.value = theForm.filter.value + '"' +
coln + '"' + op + apost + theForm.colval[i].value + apost;
             checkSelected = true;
           }
         }
         if (!checkSelected) {
           alert("Please select one of the Check buttons.");
           return (false);
         }
         return (true);
       }
       function startQuery(theForm) {
```

```
    window.open("Query?tb1=$(tb1)&filter="+encodeStr(theForm.filter.value)+
"&db=$(db)&usrid=$(usrid)&pswd=$(pswd)", "_new",
"scrollbars=yes,status=yes,width=800,height=600")
         return (true);
       }
       function showDmb(colval, theIndex) {
         if (colval[theIndex].value.substring(0,1) == "/"  ||
             colval[theIndex].value.indexOf(":") > 0)
           window.open(colval[theIndex].value, "_new",
"scrollbars=yes,status=yes,width=800,height=600");
          else window.open("/tmplobs/"+colval[theIndex].value, "_new",
"scrollbars=yes,status=yes,width=800,height=600");
         return (true);
       }
       function showClob(colval, theIndex) {
         window.open(colval[theIndex].value, "_new",
"scrollbars=yes,status=yes,width=800,height=600");
         return (true);
       }
       function showBlob(colval, theIndex) {
         window.open(colval[theIndex].value, "_new",
"scrollbars=yes,status=yes,width=800,height=600");
         return (true);
       }
       function encodeStr(Str) {
       Str = escape(Str);
       Str = Str.replace(/\//g,"%2F");
       Str = Str.replace(/\?/g,"%3F");
       Str = Str.replace(/=/g,"%3D");
       Str = Str.replace(/&/g,"%26");
       Str = Str.replace(/@/g,"%40");
       return Str;
       }
    // --> </script>
  </head>
  <body bgcolor="white">
    <form method="post" action="main" name="proctab">
       <center><h2>$(db)</h2></center>
       @DTW_ASSIGN(DATABASE, "$(db)")
       @DTW_ASSIGN(LOGIN, "$(usrid)")
       @DTW_HEXTOCHAR("$(pswd)", pswd)
       @DTW_REVERSE("$(pswd)", pswd)
       @DTW_ASSIGN(PASSWORD, pswd)
       <BR>
       <table border="0" cellpadding="0" cellspacing="0" width="100%">
         <tr>
           <td width="20%" align="right">Table:  </td>
           <td width="80%">
              <select name="tb" size="1" onchange="setparm(this.form)">
                 @selTables()
```

```
            </select>
          </td>
        </tr>
        <tr>
          <td width="50%"> </td>
          <td width="50%"> </td>
        </tr>
      </table>
      @selColumns("$(tb1)", myTable)
      <CENTER>
      <table BORDER>
        <tr />
           @DTW_TB_ROWS(myTable, numRows)
           @DTW_ASSIGN(rowIndex, "1")
           %WHILE (rowIndex <= numRows) {
             <th>@DTW_TB_rGETV(myTable, rowIndex, "1")
                 <input type="hidden" name="hidname"
value="@DTW_TB_rGETV(myTable, rowIndex, "1")"></th>
                 <input type="hidden" name="hidtype"
value="@DTW_TB_rGETV(myTable, rowIndex, "5")"></th>
             @DTW_ADD(rowIndex, "1", rowIndex)
           %}
        <tr />
           @DTW_TB_ROWS(myTable, numRows)
           @DTW_ASSIGN(rowIndex, "1")
           %WHILE (rowIndex <= numRows) {
             <td>
               %IF ("BLOB" == @DTW_TB_rGETV(myTable, rowIndex, "5") ||
                    "CLOB" == @DTW_TB_rGETV(myTable, rowIndex, "5") ||
                    "ROWID" == @DTW_TB_rGETV(myTable, rowIndex, "5"))
                 <input type="checkbox" name="check" disabled
value="$(rowIndex)"> 
                 %IF ("BLOB" == @DTW_TB_rGETV(myTable, rowIndex, "5"))
                   <select name="colval" size="1" onchange="return
showBlob(colval, @DTW_rSUBTRACT(rowIndex, "1"))" style="background-
color: #C0C0C0">
           @selBlob("$(tb1)", "@DTW_TB_rGETV(myTable, rowIndex, "1")")
                   </select>
                %ELIF ("CLOB" == @DTW_TB_rGETV(myTable, rowIndex, "5"))
                   <select name="colval" size="1" onchange="return
showClob(colval, @DTW_rSUBTRACT(rowIndex, "1"))" style="background-
color: #C0C0C0">
            @selClob("$(tb1)", "@DTW_TB_rGETV(myTable, rowIndex, "1")")
                   </select>
                 %ELSE
                   <select name="colval" size="1" disabled>
                     <option value="$(rowIndex)"></option>
                   </select>
                 %ENDIF
               %ELSE
```

```
                %IF ("DISTINCT" == @DTW_TB_rGETV(myTable, rowIndex, "5"))
                     @DTW_ASSIGN(fnd, "Ø")
                     %IF (@DTW_TB_rCOLS(myTable) > 29)
               %IF ("DB2IMAGE" == @DTW_TB_rGETV(myTable, rowIndex, "17") ||
                    "DB2VIDEO" == @DTW_TB_rGETV(myTable, rowIndex, "17") ||
                    "DB2AUDIO" == @DTW_TB_rGETV(myTable, rowIndex, "17") ||
                     "XMLCLOB" == @DTW_TB_rGETV(myTable, rowIndex, "17") ||
                  "XMLVARCHAR" == @DTW_TB_rGETV(myTable, rowIndex, "17"))
                           @DTW_ASSIGN(fnd, "1")
               @DTW_ASSIGN(dmbtype, @DTW_TB_rGETV(myTable, rowIndex, "17"))
                           %ENDIF
                         %ELSE
               %IF ("DB2IMAGE" == @DTW_TB_rGETV(myTable, rowIndex, "28") ||
                    "DB2VIDEO" == @DTW_TB_rGETV(myTable, rowIndex, "28") ||
                    "DB2AUDIO" == @DTW_TB_rGETV(myTable, rowIndex, "28") ||
                     "XMLCLOB" == @DTW_TB_rGETV(myTable, rowIndex, "28") ||
                  "XMLVARCHAR" == @DTW_TB_rGETV(myTable, rowIndex, "28"))
                           @DTW_ASSIGN(fnd, "1")
               @DTW_ASSIGN(dmbtype, @DTW_TB_rGETV(myTable, rowIndex, "28"))
                           %ENDIF
                         %ENDIF
                         %IF ($(fnd) == "1")
                           <input type="checkbox" name="check" disabled
value="$(rowIndex)"> 
                           <select name="colval" size="1" onchange="return
showDmb(colval, @DTW_rSUBTRACT(rowIndex, "1"))" style="background-color:
#CØCØCØ">
                         %IF ($(dmbtype) == "DB2AUDIO")
        @selAudValues("$(tb1)", "@DTW_TB_rGETV(myTable, rowIndex, "1")")
                         %ELSE
                           %IF ($(dmbtype) == "DB2IMAGE")
        @selImgValues("$(tb1)", "@DTW_TB_rGETV(myTable, rowIndex, "1")")
                           %ELSE
                             %IF ($(dmbtype) == "DB2VIDEO")
        @selVidValues("$(tb1)", "@DTW_TB_rGETV(myTable, rowIndex, "1")")
                             %ELSE
                               %IF ($(dmbtype) == "XMLCLOB")
          @selXmlClob("$(tb1)", "@DTW_TB_rGETV(myTable, rowIndex, "1")")
                             %ELSE
             @selXml("$(tb1)", "@DTW_TB_rGETV(myTable, rowIndex, "1")")
                             %ENDIF
                           %ENDIF
                         %ENDIF
                         %ENDIF
                         </select>
                       %ELSE
 <input type="checkbox" name="check" disabled value="$(rowIndex)"> 
                         <select name="colval" size="1" disabled>
                           <option value="$(rowIndex)"></option>
                         </select>
```

```
                    %ENDIF
                  %ELSE
                    %IF (@DTW_TB_rCOLS(myTable) > 29)
                      @DTW_ASSIGN(pc, "9")
                    %ELSE
                      @DTW_ASSIGN(pc, "6")
                    %ENDIF
                    %IF (@DTW_TB_rGETV(myTable, rowIndex, pc) > "255")
                      @selValues("$(tb1)",
"SUBSTR(@DTW_TB_rGETV(myTable, rowIndex, "1"), 1, 255)", valTable)
                    %ELSE
                      @selValues("$(tb1)", "@DTW_TB_rGETV(myTable,
rowIndex, "1")", valTable)
                    %ENDIF
                    %IF (@DTW_TB_rROWS(valTable) > 0)
                      <input type="checkbox" name="check"
value="$(rowIndex)"> 
                  @DTW_TB_SELECT(valTable,"colval","1","1","N","","","1")
                    %ELSE
                      <input type="checkbox" name="check" disabled
value="$(rowIndex)"> 
                      <select name="colval" size="1" disabled>
                        <option value="$(rowIndex)"></option>
                      </select>
                    %ENDIF
                  %ENDIF
                %ENDIF
              </td>
              @DTW_ADD(rowIndex, "1", rowIndex)
            %}
        </table>
        </CENTER>
        <BR>
        <table border="0" width="100%">
          <tr>
            <td width="100%" align="center"><input type="button"
value="Filter" name="B13" onclick="return fillfilter(this.form)"></td>
          </tr>
          <tr>
            <td width="100%" align="center"><textarea rows="10"
name="filter" cols="80"></textarea></td>
          </tr>
          <tr>
            <td width="100%" align="center"><input type="button"
value="Show Query Result" name="B14" onclick="return
startQuery(this.form)"></td>
          </tr>
        </table>
        <BR>
        <table border="0" cellpadding="0" cellspacing="0" width="100%">
```

```
          <tr>
            <td width="100%" align="center"><input type="submit"
value="Back" name="B12"></td>
          </tr>
        </table>
        %include "help2.hti"
    </form>
  </body>
</html>
%}

%HTML(Query) {
<html>
  <head>
    <meta http-equiv="content-type"
                                content="text/html;charset=iso-8859-1" />
    <title>$(tb1)</title>
  </head>
  <body bgcolor="white">
    <form method="post" name="Query">
       <center><h2>$(tb1)</h2>
       %IF ("$(tb1)" != "")
         @DTW_ASSIGN(DATABASE, "$(db)")
         @DTW_ASSIGN(LOGIN, "$(usrid)")
         @DTW_HEXTOCHAR("$(pswd)", pswd)
         @DTW_REVERSE("$(pswd)", pswd)
         @DTW_ASSIGN(PASSWORD, pswd)
         @selQuery1("$(tb1)")
         @selQuery2("$(selcol)", "$(tb1)", "$(filter)", myTable)
         %IF (@DTW_TB_rROWS(myTable) > 0)
           @DTW_TB_TABLE(myTable, "BORDER")
         %ENDIF
       %ELSE
         "No data found !!!"
       %ENDIF
       </center>
    </form>
  </body>
</html>
%}
```

HELP1.HTI

```
   <br><br><br><br><br><br><br><br><br><br><br>
   <table border="1" cellpadding="0" cellspacing="0" style="border-
collapse: collapse; border-style: double; border-width: 3; background-
color: #e2e2e2" bordercolor="#808080" width="100%" id="AutoNumber1">
     <tr>
       <td>
```

```
   <p class="MsoNormal"><font size="2" color="#808080">Choose one of the
       databases listed in the appropriate combo-box, which is
       populated with DB2 data source names from ODBC data sources on
       the database server. For the chosen database enter the correct
       user id and password.</font></td>
     </tr>
   </table>
```

HELP2.HTI

```
   <br>
   <table border="1" cellpadding="0" cellspacing="0" style="border-
collapse: collapse; border-style: double; border-width: 3; background-
color: #e2e2e2" bordercolor="#808080" width="100%" id="AutoNumber1">
     <tr>
       <td>
        <p class="MsoNormal"><span lang="EN-US"><font size="2"
color="#808080">
        Select DB2 table, choose value from combo-box with distinct
values from appropriate column, mark check-box on the left from desired
column value and click on Filter push-button to generate filter for the
query. Eventually, change Where clause in the text area box and submit
query with Show Query Result push-button.<BR>
Combo-boxes with gray background, contain all BLOB, CLOB, DB2IMAGE,
DB2VIDEO, DB2AUDIO or XML objects. These objects are not included in
query builder, but when selected, they present the content.
</font></span></td>
     </tr>
   </table>
```

Nikola Lazovic
DB2 System Administrator
Postal Savings Bank (Serbia and Montenegro)

Offline catalog runstats for online REORG

In order to create a procedure for the management of DB2 tablespaces and indexes with the minimum impact on the production environment, I have created a set of REXX procedures that, after a runstats with UPDATE(NONE), builds a PO library with a member for every tablespace or index that needs to be reorganized. After the REORGs are executed, it

analyses the runstats output, and another process builds a set of runstats jobs for all the objects that have not been reorganized, but need to be. At the end of all the utilities, a rebind builder job completes the management process. This technique of runstats prevents any impact on the access paths in use by production jobs without updating the catalog until the end of REORG. In order to reduce the time taken for runstats execution the procedure gives you a way of spreading the object code between from one to nine jobs that can be submitted at the same time.

PARAMETER DESCRIPTION

The following parameters describe how you can customize the REXX EXEC batch procedure:

- SUBSYS – DB2 subsystem name.
- DATAB – database name.
- AUTOSUB – automatic job submit.
- DSTATS – statistic time filter.
- ACRPROG – acronym plans.
- PARALLEL – maximum number of parallel jobs (1–9).
- CRBIND – creator of rebind.

CHECKLIST FOR INSTALLATION

Follow these steps to install the components of the REXX procedure:

- Allocate a USER.LIBRARY.
- Copy all the REXX EXECs, macros, PROCs, parameters, and jobs into the USER.LIBRARY:

  1 REXX EXECs – $DB2PAR0, $DB2ALL0, $DB2ACC0, $DB2RUN1, and $DB2REO1.

  2 Macros – $MDB2019 and $MDB2007.

3 PROC – DB2REXX1, AXXXREOD, AXXXREOT, AXXXREDP, and AXXXREIX.

4 PARMs – AXXXREOD, AXXXREOT, AXXXREDP, and AXXXREIX.

- Customize $DB2PAR0 REXX.

The test environment is DB2 V5 in a z/OS environment.

$DB2RUN1 REXX EXEC

```
/* REXX */
   /*-          P A R A M E T E R                -*/
   /*-    - Subsys   : DB2 subsystem name        -*/
   /*-    - Datab    : Database name             -*/
   /*-    - Autosub  : Submit job                -*/
   /*-    - DStats   : Stats Time filter         -*/
   /*-    - AcrProg  : Project alias             -*/
   /*-    - Parallel : Max parallelism number    -*/
   /*-    - CRbind   : Rebind creator            -*/
 arg parameter ; parm     = translate(parameter,' ',',')
 nparm    = words(parm)  ; subsys   = word(parm,1) ;
 datab    = word(parm,2)
 tsname   = word(parm,3) ; autosub  = word(parm,4) ;
 DStats   = word(parm,5)
 AcrProg  = word(parm,6) ; Parallel = word(parm,7) ;
 Crbind   = word(parm,8)
   /*--  test input parameter                --*/
 if nparm < 8 then do ; say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Incomplete parameter string !!!!! ' Parameter
    say '>>>>>>>>' ; say '' ; say '' ; exit ; end
 if tsname = '*' then tsname = ''
 if autosub = '*' then autosub = NO
 if autosub ¬= NO & autosub ¬= SI then do ; say '' ;  say '' ;
  say '>>>>>>>>'
    say '>>>>>>>>  Wrong Autosub 'autosub' variable !!!!!'
    say '>>>>>>>>  Specify : */NO/YES
    say '>>>>>>>>' ; say '' ; say '' ; exit ; end
 if Parallel < 2 | Parallel > 9 then do ; say '' ; say '' ;
 say '>>>>>>>>'
    say '>>>>>>>>  Wrong Parallel variable !!!!!'
    say '>>>>>>>>  Specify a number between 2 to 9   '
    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)
 $unitta = word(result,10);$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,20);
$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     =    ;  £numjob = 0 ; £a      = 0     ; £b      = 0 ;
  £c      = 1
 £g      = 1  ;  Stpnum  = 0 ; comdb1  = datab ; AcrPro =
substr(Acrprog,2,3)
   /*--  Runstats file allocationstats        --*/
 outdsrun= $hiwork'.'subsys'.'datab'.JOBRUNS'
 prmalloc = subsys' 'outdsrun' 0 15,15 f,b 80 27920 firun si'
 call $db2all0 prmalloc ; if word(result,1) = 99 then  exit
   /*--  Runstats sysprint allocation file    --*/
 outdssrun= $hiwork'.'subsys'.'datab'.SYSPRUNS'
 prmalloc = subsys' 'outdssrun' 0 450,150 f,b 124 0 fisrun si'
 call $db2all0 prmalloc ; if word(result,1) = 99 then exit
 outdssru1= $hiwork'.'subsys'.'datab'.SYSPRUNS.FULLP'
 prmalloc = subsys' 'outdssru1' 200 450,150 v,b,a 124 0 fisrun1 si'
 call $db2all0 prmalloc ; if word(result,1) = 99 then exit
   /*--  Max parallelism > 1                  --*/
 if Parallel > 1 then do ; "free fi(fisrun)"
    DO £a = 1 to Parallel ; outdssru2=
$hiwork'.'subsys'.'datab'.SYSPRUNS.$'||£a
       prmalloc = subsys' 'outdssru2' 200 450,150 v,b,a 124 0 fisrun si'
       call $db2all0 prmalloc ; "free fi(fisrun)"
       if word(result,1) = 99 then exit ; end ; end
   /*--  File bridge name                     --*/
 outdstsin = $hiwork'.'subsys'.'datab'.$DB2RUN0.SYSTSIN'
 outdsprt  = $hiwork'.'subsys'.'datab'.$DB2RUN0.SYSPRINT'
 outdsrec  = $hiwork'.'subsys'.'datab'.$DB2RUN0.SYSREC00'
   /*--  SYSIN file allocation                --*/
```

```
 dsn = sysdsn('''''outdsin''''')
 if dsn ¬= OK then do ; say '' ; outdsin=
$hiwork'.'subsys'.'datab'.$DB2RUN0.SYSIN'
    prmalloc = subsys' 'outdsin' 0 5,1 f,b 80 2792 sysin si'
    call $db2all0 prmalloc ; if word(result,1) = 99 then exit ; end
  else do "alloc da('"outdsin"') f("sysin") shr reuse" ; end
 sk.1='SELECT A.DBNAME,A.NAME,A.STATSTIME,B.CARD,B.CARDF           '
 sk.2='       FROM SYSIBM.SYSTABLESPACE A ,                        '
 sk.3='            SYSIBM.SYSTABLES     B                         '
 sk.4='       WHERE A.DBNAME = '''datab'''                         '
 sk.5='         AND A.DBNAME   = B.DBNAME                          '
 sk.6='         AND A.CREATOR  = B.CREATOR                         '
 sk.7='         AND A.NAME     = B.TSNAME                          '
 sk.8='         AND B.TYPE = ''T''                                 '
 if DStats = '*' then
 sk.9='                                                           '
 else
 sk.9='         AND A.STATSTIME < '''DStats'''                     '
 sk.10='         AND A.NAME   LIKE '''tsname'%'' ORDER BY 5 desc ; '
 sk.0=10
 jobw = sysin ; call WriteRec ; call $db2acc0
subsys','datab||'.$DB2RUN0'
 if word(result,1) = 99 then do ; call Free ;  exit ;  end
 RCdb2  = word(result,1) ; okunlo = word(result,2) ;
NRrec  = word(result,3)
   /*--  DB2 access OK start elaboration      --*/
 if RCdb2 = 00 then do
   /*--  Read record                          --*/
    "alloc da('"outdsrec"') f("sysrec00") shr reuse" ;
xx=outtrap(trpread01.)
       "execio * diskr sysrec00 (stem sysrec00. finis" ; xx=outtrap(off)
    if rc > 0 then do ; do £a = 1 to trpread01.0 ; say trpread01.£a ;
end
       say '' ; say '' ;  say '>>>>>>>>'
       say '>>>>>>>>  Read file error "'outdsrec'"    '
       say '>>>>>>>>  RC='rc'. Verify.'
       say '>>>>>>>>' ; say '' ; say '' ; exit ;  end
    DO £b = 1 to Parallel
       jobw = firun ; "alloc da('"outdsrun"') f("jobw") mod reuse" ;
call hdrjob
       DO £i = £c to sysrec00.0 ; $DBname  =
strip(substr(sysrec00.£i,1,8))
          $TSname  = strip(substr(sysrec00.£i,9,8))
          $TimeRun = strip(substr(sysrec00.£i,17,16))
          $Card    = c2x(substr(sysrec00.£i,43,4))
   /*--  if Card set to ffffffff              --*/
          if $Card  ¬= 'FFFFFFFF' then
             $Card = x2d(d2x(c2d(substr(sysrec00.£i,43,4))))
          else do
   /*--  Cardf set to -1                      --*/
```

```
            $SCardf = c2x(substr(sysrec00.£i,47,1))
            if $SCardf = 'C1' then do ;$Cardf = '-1' ; $Card = $Cardf ;
end
   /*--  Cardf set                              --*/
            else do
               LCardf0 = substr($SCardf,2,1) ;  LCardf1 = LCardf0  % 2
               LCardf2 = LCardf0 // 2
               if LCardf2 > 0 then  LCardf1 = LCardf1 + LCardf2
               $WCardf = c2x(substr(sysrec00.£i,48,LCardf1))
               $Cardf  = x2d(substr($WCardf,1,LCardf0))
               $Card = $Cardf ; end  ; end
   /*--  Write jcl jcl                          --*/
          Stpnum = Stpnum + 1 ; jobw = firun
         "alloc da('"outdsrun"') f("jobw") mod reuse" ; call Stpjob
sk.£g=' RUNSTATS TABLESPACE ('$DBname'.'$TSname') INDEX(ALL)
SHRLEVEL(CHANGE)'
          £g = £g + 1 ; if DStats = '*' then
sk.£g='           REPORT(YES) UPDATE(NONE)
'
          else
sk.£g='           REPORT(YES) UPDATE(ALL)
'
          £g = £g + 1 ;  £i = £i + Parallel - 1
          end    /***** DO £i = 1 to sysrec00.0 *****/
sk.£g='//* ------------------------------------------------ *      '
       £g = £g + 1
sk.£g='//LAB1     IF (RC LT 5) THEN                                 '
       £g = £g + 1
sk.£g='//STEP999   EXEC PGM=IEBCOPY                                 '
       £g = £g + 1
sk.£g='//SYSPRINT DD  SYSOUT=*                                      '
       £g = £g + 1
sk.£g='//INP      DD
DISP=SHR,DSN='$hiwork'.'subsys'.'datab'.SYSPRUNS.$'||£c
       £g = £g + 1
sk.£g='//OUT      DD  DISP=SHR,DSN='outdssru1
       £g = £g + 1
sk.£g='//SYSIN    DD  *                                             '
       £g = £g + 1
sk.£g='   COPY INDD=((INP,R)),OUTDD=OUT                             '
       £g = £g + 1
sk.£g='//LAB1END  ENDIF                                             '
       £g = £g + 1
sk.£g='//* ------------------------------------------------ *      '
       £g = £g + 1 ; jobw = firun ; "alloc da('"outdsrun"') f("jobw")
mod reuse"
       sk.0=£g ; Call WriteRec ; £c = £c + 1 ; £g = 1
       end   /***** DO £b = 1 to Parallel   *****/
sk.1='//J'AcrPro'ESPK JOB ('$accn'),''Runstats
database'',CLASS='$class', '
```

```
sk.2='//      MSGCLASS='$msgcla',USER='$user',REGION='$region',        '
sk.3='//      MSGLEVEL=('$msglvl'),NOTIFY='$notif',TYPRUN=HOLD         '
sk.4='/*JOBPARM BYTES=999999,LINES=9999                                '
sk.5='//* ------------------------------------------------ *        '
sk.6='//* ----  Split Partitioned DS /Build Online Reorg  -- *        '
sk.7='//* ------------------------------------------------ *        '
sk.8='//DB2PROC  JCLLIB ORDER=('$proclib')                             '
sk.9='//JOBLIB   DD DISP=SHR,DSN='$dsnload
sk.10='//* ------------------------------------------------ *        '
sk.11='//STEP00   EXEC   DB2REXX1                                      '
sk.12='//REXX00.SYSTSIN  DD  *                                         '
sk.13='  ISPSTART CMD($DB2REO1
'subsys','datab','AcrProg','Parallel','Crbind')'
sk.14='//* ------------------------------------------------ *       '
       sk.0=14 ; jobw = firun ;
       "alloc da('"outdsrun"') f("jobw") mod reuse" ;  Call WriteRec
 say '' ; say '>>>>>>>>  +-------------------------------+'
 say '>>>>>>>>    TOT    job Runstat    :' right(£numjob,6,' ')
 say '>>>>>>>>    Tot    tablespace     :' right(sysrec00.0,6,' ')
 say '>>>>>>>>  +-------------------------------+' ;  say ''
   /*--  Submit Runstats jobs                 ----*/
    if autosub =  si then do
       xx=outtrap(trp08.); address tso "submit '"outdsrun"'" ;
xx=outtrap(off)
       if rc > 0 then do ; do a = 1 to trp08.0 ; say trp08.a ; end ;
exit ;  end
       say '' ;   say time()
       say time() '---->> Submitted '£numjob' job of Runstats ....'
       say time() ;  say '' ; say ''
       xx=outtrap(trpdummy.); address tso "delete '"outdsrun"'"
       xx=outtrap(off) ; end ; end
 else do
    say '' ; say '>>>>>>>>' ; say '>>>>>>>>'
    say '>>>>>>>>    Unpredictable error end elaboration'
    say '>>>>>>>>' ; say '>>>>>>>>' ; say '' ; end ; call Free ;  exit
   /*--  Routine write headr job              ----*/
 Hdrjob:
    £numjob = £numjob + 1 ;  dbcen = center(Datab,10) ; Stpnum = 0
sk.1='//J'AcrPro'ERU'£c' JOB ('$accn'),''Runstats
database'',CLASS='$class', '
sk.2='//      MSGCLASS='$msgcla',USER='$user',REGION='$region',        '
sk.3='//      MSGLEVEL=('$msglvl'),NOTIFY='$notif
sk.4='/*JOBPARM BYTES=999999,LINES=9999                                '
sk.5='//* ------------------------------------------------ *        '
sk.6='//* ------     R u n s t a t s  'dbcen'       ------ *          '
sk.7='//* ------------------------------------------------ *        '
sk.8='//DB2PROC  JCLLIB ORDER=('$proclib')                             '
sk.9='//JOBLIB   DD DISP=SHR,DSN='$dsnload
    sk.0=9 ; Call WriteRec ; return
   /*-- Routine write step                  ----*/
```

```
 Stpjob:
    comds    = $hiwork'.'subsys'.'datab'.SYSPRUNS.$'||£c'('$Tsname')'
    $Cardful = right($Card,11,' ') ; Stpnum   = right(Stpnum,3,'Ø')
sk.£g='//* ------ Card:'$Cardful' al '$TimeRun' ------ *       '
    £g = £g + 1
sk.£g='//STEP'Stpnum' EXEC
PGM=DSNUTILB,PARM='''subsys',RU'time('s')$Tsname'''
    £g = £g + 1
sk.£g='//SYSPRINT DD DISP=SHR,DSN='comds
    £g = £g + 1
sk.£g='//SYSIN    DD *                                                '
    £g = £g + 1 ; return
   /*--  Routine write output record         ----*/
 WriteRec :
    "EXECIO * DISKW "jobw" (STEM sk. FINIS"
 ClearRec :
    DO f = 1 to sk.Ø ; sk.f = blk ; end ; return
   /*--  Free work datasets                  ----*/
 Free    :
    xx=outtrap(trpdummy.)
       "free fi(firun)" ; "free fi(sysrecØØ)" ;
 address tso "delete '"outdsprt"'"
       "free fi(sysprint)" ; address tso "delete '"outdsrec"'"
       address tso "delete '"outdstsin"'" ; "free fi(sysin)"
       address tso "delete '"outdsin"'"
    xx=outtrap(off) ; return
```

$DB2REO1 REXX EXEC

```
/* REXX */
   /*---------------------------------------------*/
   /*-  - input    : Output Runstats Dataset     -*/
   /*-          P A R A M E T E R                -*/
   /*-  - subsys   : DB2 Subsystem name          -*/
   /*-  - datab    : Database name               -*/
   /*-  - AcrProg  : Alias project              -*/
   /*-  - Parallel : Parallelism number          -*/
   /*-  - CRbind   : Rebind Creator             -*/
   /*---------------------------------------------*/
arg Parameter ; parm = translate(Parameter,' ',',')
nparm    = words(parm)  ; subsys   = word(parm,1) ;
 datab    = word(parm,2)
AcrProg  = word(parm,3) ; Parallel = word(parm,4) ;
 CRbind   = word(parm,5)
   /*--  test input parameter                --*/
 if nparm < 5 then do ; say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Incomplete parameter string !!!!! ' Parameter
    say '>>>>>>>>' ; say '' ; say '' ; exit ;  end
   /*--  Parameters assignment               --*/
```

```
 call @db2par0 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)
 $unitta = word(result,10);$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,20);
$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    =     ; £a     = 0  ; £c     = 0  ;  £d     = 0  ; £b     = 0
 £e     = 0   ; £g     = 0  ; £h     = 0  ;  £hPunt = 0  ; £hctr  = 0
 £i     = 0   ; £l     = 0  ; £m     = 0  ;  £n     = 0  ; £o     = 1
 £p     = 1   ; £z     = 0  ; £y     = 0  ;  £f613  = 0  ; £f617  = 0
 £f618  = 0   ; ctreo  = 0  ; ctreom = 0  ;  noix   = 'N' ; veremp =
 Sireo  = 'N' ;  AcrPro = substr(Acrprog,2,3)
 $Tempo = space(translate(time(),'',':'),0)
 $Data  = space(translate(date(u),'','/'),0) ; Partloop = 0
   /*--  Partitioned --> Sequential .FULLP    --*/
 inpds00 = $hiwork'.'subsys'.'datab'.SYSPRUNS.FULLP' ;  Mstart = ' '
 "ispexec lminit dataid("inpd1")  dataset('"inpds00"') enq(shr)"
 "ispexec lmopen dataid("inpd1")  option(input)"
 "ispexec lmmlist dataid("inpd1") option(save) member("Mstart")
   stats(yes) group("subsys") "
   /*--  Read Lmlist output .FULLP           --*/
 inpds01 = $hiwork'.'subsys'.MEMBERS'
 xx=outtrap(trplmml.) ;
 "alloc da('"inpds01"') f(outlmml) shr reuse" ;xx=outtrap(off)
 if rc > 0 then do ; do £l = 1 to trplmml.0 ; say trplmml.£l ; end
    say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Allocation for "'inpds01'"'
    say '>>>>>>>>  Ko - RC='rc'. Verify      '
    say '>>>>>>>>' ; say '' ; say '' ; exit ; end
 xx=outtrap(trplmml.)
```

```
    "execio * diskr outlmml (stem outlmml. finis" ; "free fi(outlmml)"
 xx=outtrap(off)
 if rc > Ø then do ; do £l = 1 to trplmml.Ø ; say trplmml.£l ;  end
    say '' ;  say '' ;  say '>>>>>>>>'
    say '>>>>>>>>  Read file error "'inpdsØ1'"'
    say '>>>>>>>>  RC='rc'. Verify            '
    say '>>>>>>>>' ;  say '' ;  say '' ;  exit ;  end
 if outlmml.Ø = Ø then do ; say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  The input file "'inpdsØ1'"'
    say '>>>>>>>>  is empty !!!!!!  Verify   '
    say '>>>>>>>>' ; say '' ;  say '' ;  exit ;  end
 jobw = fisyspru ; inpdsØ2 = $hiwork'.'subsys'.'datab'.SYSPRUNS'
 "alloc da('"inpdsØ2"') f("jobw") mod reuse"
 Do £m = 1 to outlmml.Ø
    inpdsØ3 =
$hiwork'.'subsys'.'datab'.SYSPRUNS.FULLP('word(outlmml.£m,1)')'
    "alloc da('"inpdsØ3"') f(runpart) shr reuse"
    „execio * diskr runpart  (stem runpart. finis"
    do £n = 1 to runpart.Ø ; sk.£p = runpart.£n
       £p = £p + 1 ; end ; £o = £o + 1
   /*--  Write and Clean buffer SK.           --*/
    if £o > 1ØØ then do ; sk.Ø=£p ; jobw = fisyspru ; Nrec = '*' ; call
WriteRec
       £o = 1 ; £p = 1 ; end ; end ; sk.Ø=£p
 jobw = fisyspru ; Nrec = '*' ;  call WriteRec
   /*--  File bridge name                     --*/
 outdstsin = $hiwork'.'subsys'.'datab'.$DB2REOØ.SYSTSIN'
 outdsprt  = $hiwork'.'subsys'.'datab'.$DB2REOØ.SYSPRINT'
 outdsrec  = $hiwork'.'subsys'.'datab'.$DB2REOØ.SYSRECØØ'
   /*--  SYSIN file allocation                --*/
 dsn = sysdsn('''outdsin''')
 if dsn ¬= OK then do ; say '' ; outdsin=
$hiwork'.'subsys'.'datab'.$DB2REOØ.SYSIN'
    prmalloc = subsys' 'outdsin' Ø 5,1 f,b 8Ø 2792Ø sysin si'
    call $db2allØ prmalloc ; if word(result,1) = 99 then exit ; end
  else do "alloc da('"outdsin"') f("sysin") shr reuse" ; end
sk.1=' SELECT CREATOR,NAME,CLUSTERING,CLUSTERED,DBNAME,INDEXSPACE '
sk.2='        FROM SYSIBM.SYSINDEXES                              '
sk.3='        WHERE DBNAME =  '''datab'''                         '
sk.4='          and CLUSTERING = '''Y''';                         '
 sk.Ø=4 ; jobw = sysin ; Nrec = '*' ; call WriteRec
 call $db2accØ subsys','datab||'.$DB2REOØ'
 if word(result,1) = 99 then do ; call Free ; exit ; end
 RCdb2  = word(result,1) ; okunlo = word(result,2) ;
NRrec  = word(result,3)
   /*--  DB2 access OK start elaboration      --*/
 if RCdb2 = ØØ then do ; dsn = sysdsn('''outdsrec''')
    if dsn = OK then do ; xx=OUTTRAP(trpØ6.)
          "ispexec edit dataset('"outdsrec"') macro($mdb2Ø19)"
;xx=OUTTRAP(OFF) ; end
```

```
    else do; say '' ;  say '' ; say '>>>>>>>>'
       say '>>>>>>>>  Allocation "'outdsrec'" failed !!!!!'
       say '>>>>>>>>          Verify                      '
       say '>>>>>>>>' ; say '' ; say '' ; exit ;  end
   /*--  Read output DB2 record               --*/
    "alloc da('"outdsrec"') f("sysrec00") shr reuse" ;
xx=outtrap(trpread01.)
       "execio * diskr sysrec00 (stem sysrec00. finis" ; xx=outtrap(off)
    if rc > 0 then do ; do £a = 1 to trpread01.0 ; say trpread01.£a ;
end
       say '' ; say '' ; say '>>>>>>>>'
       say '>>>>>>>>  Read file error "'outdsrec'"    '
       say '>>>>>>>>  RC='rc'. Verify                 '
       say '>>>>>>>>' ; say '' ; say '' ; exit ; end ;  end
 else do ; say '' ; say '>>>>>>>>' ; say '>>>>>>>>'
    say '>>>>>>>>    Uupredictable DB2 Error. End elaboration'
    say '>>>>>>>>' ; say '>>>>>>>>' ; say '' ; end ; call Free
   /*--  Macro for Runstats output            --*/
 dsn = sysdsn('''inpds02''')
 if dsn = OK then do ; xx=OUTTRAP(trp06.)
       "ispexec edit dataset('"inpds02"') macro($mdb2007)"
;xx=OUTTRAP(OFF) ; end
 else do ; say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Allocation "'inpds02'" failed '
    say '>>>>>>>>           Verify              '
    say '>>>>>>>>' ; say '' ; say '' ; exit ; end
   /*--  Read output Runstats                 --*/
 xx=outtrap(trp0.)
    "alloc da('"inpds02"') f(outruns) shr reuse"; xx=outtrap(off)
 if rc > 0 then do ; do a = 1 to trp0.0 ; say trp0.a ; end
    say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  Allocation error for "'inpds02'"'
    say '>>>>>>>>  RC='rc'. Verify                 '
    say '>>>>>>>>' ; say '' ; say '' ; exit ; end
 xx=outtrap(trp1.);"execio * diskr outruns (stem outruns. finis" ; "free
fi(outruns)"
 xx=outtrap(off)
 if rc > 0 then do ; do a = 1 to trp1.0 ; say trp1.a ; end
    say '' ;  say '' ;  say '>>>>>>>>'
    say '>>>>>>>>  Read file error for "'inpds02'"'
    say '>>>>>>>>  RC='rc'. Verify                 '
    say '>>>>>>>>' ; say '' ;  say '' ; exit ;  end
 if outruns.0 = 0 then do ;  say '' ; say '' ; say '>>>>>>>>'
    say '>>>>>>>>  The input file "'inpds02'"'
    say '>>>>>>>>  is empty !!!!!!   verify  '
    say '>>>>>>>>' ; say '' ; say '' ; exit ; end
   /*--  Reorg job file allocation            --*/
 outreopc = $hiwork'.'subsys'.'datab'.$DB2REO0.OPCINP'
 prmalloc = subsys' 'outreopc' 200 150,75 f,b 80 27920 fireoop si'
 call $db2all0 prmalloc ; if word(result,1) = 99 then exit
```

```
   /*--  Report Reorg file allocation         --*/
 outreoin=  $hiwork'.'subsys'.'datab'.$DB2REØ0.REORINP'
 prmalloc = subsys' 'outreoin' Ø 15,15 f,b 133 133Ø fireoin si'
 call $db2allØ prmalloc ; if word(result,1) = 99 then exit
 data = substr(date(s),7,2)'/'substr(date(s),5,2)'/'substr(date(s),1,4)
          say '' ;  say '' ; say 'Elaborazione del: 'data
          say '             ore: 'time()  ; say ''
           say '+-------------------------+----------------------------
--------+----------+------------+'
          say '|    T A B L E S P A C E    |            I N D E X
 | R E O R G | PARTITION  |'
          say '|                           |
 |   TS   IX |  C A R D   |'
         sk.1='Elaborazione del: 'data
         sk.2='             ore: 'time()
          sk.3='+-------------------------+----------------------------
--------+----------+------------+'
         sk.4='|    T A B L E S P A C E    |            I N D E X
 | R E O R G | PARTITION  |'
         sk.5='|                           |
 |   TS   IX |  C A R D   |'
          sk.6='+-------------------------+----------------------------
--------+----------+------------+'
 jobw = fireoin ; "alloc da('"outreoin"') f("jobw") mod reuse"
 sk.Ø = 6 ; Nrec = '*' ;  call WriteRec
 DO £d = 1 to outruns.Ø ; call ClRecOut ; MsgSel = word(outruns.£d,1)
    if MsgSel = 'DSNUØ5ØI' then do
           say '+-------------------------+----------------------------
--------+------+--+------------+'
       £d = £d + 1 ; do until MsgSel = 'DSNUØ5ØI'
          MsgSel = word(outruns.£d,1)
   /*--  Msg di fine Utility                 --*/
          if MsgSel = 'DSNUØ1ØI' then leave
   /*--  Msg Tablespace without index        --*/
          if MsgSel = 'DSNU718I' then do ; noix = 'Y' ; end
   /*--  Msg Tablespace COPY pending         --*/
          if MsgSel = 'DSNU38ØI' then do
     say '' ; say 'Tablespace in Copy Pending state. Remove the problem'
             say outruns.£d ; £d = £d + 1 ; iterate ; end
   /*--  Msg Tablespace                      --*/
          if MsgSel = 'DSNU613I' then do ;  £f613 = £f613 + 1
             wrecØØ = word(outruns.£d,11) ; wrecØ1 = word(outruns.£d,1Ø)
             wrecØ2 = word(outruns.£d,9)  ; DTwrecØØ = datatype(wrecØØ)
             if DTwrecØØ ¬= NUM then do
                say 'Wrong type of Partitionn !!!!. End elaboration'
                say outruns.£d ; exit ; end
          if noix = 'Y' then do ; £f618 = £f618 + 1 ; a1a.£f613 = wrecØ2
                a1b.£f613 = substr(wrecØ1,1,4)
             a1c.£f613 = wrecØØ ; a2a.£f613 = 'Tablespace without Index'
                a7.£f613  = '    ' ; noix = 'N' ; iterate ; end
```

```
call Msg_DSNU613I ; a1a.£f613 = wrec02 ; a1b.£f613 = substr(wrec01,1,4)
     a1c.£f613 = wrec00 ; a5.£f613  = reo613 ; sireo613.£f613  = reo613
           end                /* End if MsgSel = 'DSNU613I'       */
  /*--  Msg Table                                --*/
         if MsgSel = 'DSNU614I' then do
      wrec00 = word(outruns.£d,9) ; Dsnu614_TbName = word(outruns.£d,9)
           call Msg_DSNU614I ; Dsnu614_Card = npCard ;  end
  /*--  Msg index                                --*/
         if MsgSel = 'DSNU618I' then do
            £f618 = £f618 + 1 ; wrec00 = word(outruns.£d,9) ;
            wrec01 = word(outruns.£d,10) ; wrec02 = word(outruns.£d,11)
call Msg_DSNU618I ; a2a.£f618 = wrec00 ; a2b.£f618 = substr(wrec01,1,4)
          a2c.£f618 = wrec02 ; a3.£f618 = reo6181 ; a4.£f618 = reo6182
            a7.£f618 = npCard ; Dsnu618_Card.£f618 = npCard
            sireo6181.£f618 = reo6181 ; sireo6182.£f618 = reo6182 ; end
  /*--  Msg index                                --*/
         if MsgSel = 'DSNU617I' then do
            £y = £d -1 ; veremp = substr(outruns.£y,25,22)
            £f617 = £f617 + 1 ; wrec00 = word(outruns.£d,9)
            if veremp = '- INDEXSPACE IS EMPTY' then a6.£f617 = 'N'
            else do ; call Msg_DSNU617I ; a6.£f617 = reo617
               sireo617.£f617 = reo617 ; end ; end ; £d = £d + 1
        end                  /* End do until MsgSel = 'DSNU050I' */
```

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

Massimo Balzano
DB2 Systems Programmer (Italy)

DB2 utility generation process – part 2

This month we conclude the code for a utility generation process.

```
QUEUE "//*"
"EXECIO 1 DISKW RUNSFILE"
QUEUE  "//*************************************************************"
"EXECIO 1 DISKW RUNSFILE"
QUEUE "//*      RUNSTATS TABLE "||STRIP(TBC,T,' ')||"."||TBN
"EXECIO 1 DISKW RUNSFILE"
QUEUE  "//*************************************************************"
"EXECIO 1 DISKW RUNSFILE"
QUEUE "//*"
"EXECIO 1 DISKW RUNSFILE"
```

```
QUEUE "//"||TSTEPNUM||" EXEC PGM=DSNUTILB,REGION=4Ø96K,PARM='<DSN>'"
"EXECIO 1 DISKW RUNSFILE"
QUEUE "//STEPLIB   DD DISP=SHR,DSN=DSN71Ø.SDSNLOAD"
"EXECIO 1 DISKW RUNSFILE"
QUEUE "//          DD DISP=SHR,DSN=DSN71Ø.SDSNEXIT"
"EXECIO 1 DISKW RUNSFILE"
QUEUE "//SYSPRINT  DD SYSOUT=*"
"EXECIO 1 DISKW RUNSFILE"
QUEUE "//UTPRINT   DD SYSOUT=*"
"EXECIO 1 DISKW RUNSFILE"
QUEUE "//SYSUDUMP  DD SYSOUT=*"
"EXECIO 1 DISKW RUNSFILE"
QUEUE "//SYSIN     DD *"
"EXECIO 1 DISKW RUNSFILE"
QUEUE "  RUNSTATS TABLESPACE "||STRIP(DBN,T,' ')||"."||TSN
"EXECIO 1 DISKW RUNSFILE"
QUEUE "           TABLE (ALL) INDEX (ALL) UPDATE (ALL) "
"EXECIO 1 DISKW RUNSFILE"
QUEUE "/*"
"EXECIO 1 DISKW RUNSFILE"
RETURN
RCVT_FILE:
IF SUBSTR(TSTEPNUM,6,2) = "ØØ"
THEN
  DO
    QUEUE JC1
    QUEUE JC2
    QUEUE JC3
    "EXECIO 3 DISKW RCVTFILE"
  END
QUEUE "//*"
"EXECIO 1 DISKW RCVTFILE"
QUEUE  "//****************************************************************"
"EXECIO 1 DISKW RCVTFILE"
QUEUE "//*     RECOVER TABLESPACE "||STRIP(DBN,T,' ')||"."||TSN
"EXECIO 1 DISKW RCVTFILE"
QUEUE  "//****************************************************************"
"EXECIO 1 DISKW RCVTFILE"
QUEUE "//*"
"EXECIO 1 DISKW RCVTFILE"
QUEUE "//"||TSTEPNUM||" EXEC PGM=DSNUTILB,REGION=4Ø96K,PARM='<DSN>'"
"EXECIO 1 DISKW RCVTFILE"
QUEUE "//STEPLIB   DD DISP=SHR,DSN=DSN71Ø.SDSNLOAD"
"EXECIO 1 DISKW RCVTFILE"
QUEUE "//          DD DISP=SHR,DSN=DSN71Ø.SDSNEXIT"
"EXECIO 1 DISKW RCVTFILE"
QUEUE "//SYSPRINT  DD SYSOUT=*"
"EXECIO 1 DISKW RCVTFILE"
QUEUE "//SYSIN     DD *"
"EXECIO 1 DISKW RCVTFILE"
```

```
QUEUE "  RECOVER TABLESPACE "||STRIP(DBN,T,' ')||"."||TSN
"EXECIO 1 DISKW RCVTFILE"
QUEUE "/*"
"EXECIO 1 DISKW RCVTFILE"
RETURN
RCVI_FILE:
IF SUBSTR(ISTEPNUM,6,2) = "ØØ"
THEN
  DO
    QUEUE JC1
    QUEUE JC2
    QUEUE JC3
    "EXECIO 3 DISKW RCVIFILE"
  END
QUEUE "//*"
"EXECIO 1 DISKW RCVIFILE"
QUEUE  "//*************************************************************"
"EXECIO 1 DISKW RCVIFILE"
QUEUE "//*     RECOVER INDEX "||STRIP(TBC,T,' ')||"."||IXN
"EXECIO 1 DISKW RCVIFILE"
QUEUE  "//*************************************************************"
"EXECIO 1 DISKW RCVIFILE"
QUEUE "//*"
"EXECIO 1 DISKW RCVIFILE"
QUEUE "//"||ISTEPNUM||" EXEC PGM=DSNUTILB,REGION=4Ø96K,PARM='<DSN>'"
"EXECIO 1 DISKW RCVIFILE"
QUEUE "//STEPLIB   DD DISP=SHR,DSN=DSN71Ø.SDSNLOAD"
"EXECIO 1 DISKW RCVIFILE"
QUEUE "//          DD DISP=SHR,DSN=DSN71Ø.SDSNEXIT"
"EXECIO 1 DISKW RCVIFILE"
QUEUE "//SYSPRINT  DD SYSOUT=*"
"EXECIO 1 DISKW RCVIFILE"
QUEUE "//SYSIN     DD *"
"EXECIO 1 DISKW RCVIFILE"
QUEUE "  RECOVER INDEX "||STRIP(TBC,T,' ')||"."||IXN
"EXECIO 1 DISKW RCVIFILE"
QUEUE "/*"
"EXECIO 1 DISKW RCVIFILE"
RETURN
REOG_FILE:
IF SUBSTR(TSTEPNUM,6,2) = "ØØ"
THEN
  DO
    QUEUE JC1
    QUEUE JC2
    QUEUE JC3
    "EXECIO 3 DISKW REOGFILE"
  END
QUEUE "//*"
"EXECIO 1 DISKW REOGFILE"
```

```
QUEUE  "//*************************************************************"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//*   REORGANIZE TABLESPACE "||STRIP(DBN,T,' ')||"."||TSN
"EXECIO 1 DISKW REOGFILE"
QUEUE  "//*************************************************************"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//*"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//"||TSTEPNUM||" EXEC PGM=DSNUTILB,REGION=4Ø96K,PARM='<DSN>'"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//STEPLIB   DD DISP=SHR,DSN=DSN71Ø.SDSNLOAD"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//          DD DISP=SHR,DSN=DSN71Ø.SDSNEXIT"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//SYSPRINT  DD SYSOUT=*"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//SYSDISC   DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".SYSDISC"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//SYSUT1    DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".SYSUT1"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//SORTOUT   DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".SORTOUT"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//SYSPUNCH  DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".SYSPUNCH"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//DATAWKØ1  DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".DATAWKØ1"
```

```
"EXECIO 1 DISKW REOGFILE"
QUEUE "//DATAWKØ2  DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".DATAWKØ2"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//DATAWKØ3  DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".DATAWKØ3"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//DATAWKØ4  DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".DATAWKØ4"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//SORTWKØ1  DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".SORTWKØ1"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//SORTWKØ2  DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".SORTWKØ2"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//SORTWKØ3  DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".SORTWKØ3"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//SORTWKØ4  DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".SORTWKØ4"
"EXECIO 1 DISKW REOGFILE"
```

```
QUEUE "//COPY00001   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPY00001,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//COPY00002   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPY00002,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//COPY00003   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPY00003,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//COPY00004   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPY00004,"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW REOGFILE"
QUEUE "//SYSIN     DD *"
"EXECIO 1 DISKW REOGFILE"
QUEUE "  REORG TABLESPACE "||STRIP(DBN,T,' ')||"."||TSN
"EXECIO 1 DISKW REOGFILE"
QUEUE "         COPYDDN (COPY00001,COPY00002)"
"EXECIO 1 DISKW REOGFILE"
QUEUE "         RECOVERYDDN (COPY00003,COPY00004)"
"EXECIO 1 DISKW REOGFILE"
QUEUE "/*"
"EXECIO 1 DISKW REOGFILE"
RETURN
RBLD_FILE:
IF SUBSTR(ISTEPNUM,6,2) = "00"
THEN
  DO
    QUEUE JC1
    QUEUE JC2
    QUEUE JC3
    "EXECIO 3 DISKW RBLDFILE"
  END
QUEUE "//*"
"EXECIO 1 DISKW RBLDFILE"
```

```
QUEUE  "//*************************************************************"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//*     REBUILD INDEX "||STRIP(TBC,T,' ')||"."||IXN
"EXECIO 1 DISKW RBLDFILE"
QUEUE  "//*************************************************************"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//*"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//"||TSTEPNUM||" EXEC PGM=DSNUTILB,REGION=4Ø96K,PARM='<DSN>'"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//STEPLIB   DD DISP=SHR,DSN=DSN71Ø.SDSNLOAD"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//          DD DISP=SHR,DSN=DSN71Ø.SDSNEXIT"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//SYSREC    DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(IXN,T,' ')||".SYSREC"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//SYSUT1    DD DISP=(MOD,DELETE,CATLG),UNIT=SYSDA,"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//             VOL=SER=TSOPK1,SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//             DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(IXN,T,' ')||".SYSUT1"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//SYSPRINT  DD SYSOUT=*"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "//SYSIN     DD *"
"EXECIO 1 DISKW RBLDFILE"
QUEUE "  REBUILD INDEX "||STRIP(TBC,T,' ')||"."||IXN
"EXECIO 1 DISKW RBLDFILE"
QUEUE "/*"
"EXECIO 1 DISKW RBLDFILE"
RETURN
TCPY_FILE:
IF SUBSTR(TSTEPNUM,6,2) = "ØØ"
THEN
  DO
    QUEUE JC1
    QUEUE JC2
    QUEUE JC3
    "EXECIO 3 DISKW TCPYFILE"
  END
QUEUE "//*"
"EXECIO 1 DISKW TCPYFILE"
QUEUE  "//*************************************************************"
"EXECIO 1 DISKW TCPYFILE"
```

```
QUEUE "//*   IMAGE COPY TABLESPACE "||STRIP(DBN,T,' ')||"."||TSN
"EXECIO 1 DISKW TCPYFILE"
QUEUE  "//*************************************************************"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//*"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//"||TSTEPNUM||" EXEC PGM=DSNUTILB,REGION=4Ø96K,PARM='<DSN>'"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//STEPLIB   DD DISP=SHR,DSN=DSN71Ø.SDSNLOAD"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//          DD DISP=SHR,DSN=DSN71Ø.SDSNEXIT"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//SYSPRINT  DD SYSOUT=*"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//DSSPRINT  DD SYSOUT=*"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//COPYØØ1   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPYØØ1,"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//COPYØØ2   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPYØØ2,"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//COPYØØ3   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPYØØ3,"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//COPYØØ4   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPYØØ4,"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "//SYSIN     DD *"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "  COPY TABLESPACE "||STRIP(DBN,T,' ')||"."||TSN
"EXECIO 1 DISKW TCPYFILE"
QUEUE "       COPYDDN (COPYØØ1,COPYØØ2)"
"EXECIO 1 DISKW TCPYFILE"
```

```
QUEUE "       RECOVERYDDN (COPY00003,COPY00004)"
"EXECIO 1 DISKW TCPYFILE"
QUEUE "/*"
"EXECIO 1 DISKW TCPYFILE"
RETURN
ICPY_FILE:
IF SUBSTR(ISTEPNUM,6,2) = "00"
THEN
  DO
    QUEUE JC1
    QUEUE JC2
    QUEUE JC3
    "EXECIO 3 DISKW ICPYFILE"
  END
QUEUE "//*"
"EXECIO 1 DISKW ICPYFILE"
QUEUE  "//*****************************************************************"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//*   IMAGE COPY INDEX "||STRIP(TBC,T,' ')||"."||IXN
"EXECIO 1 DISKW ICPYFILE"
QUEUE  "//*****************************************************************"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//*"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//"||ISTEPNUM||" EXEC PGM=DSNUTILB,REGION=4096K,PARM='<DSN>'"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//STEPLIB   DD DISP=SHR,DSN=DSN710.SDSNLOAD"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//          DD DISP=SHR,DSN=DSN710.SDSNEXIT"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//SYSPRINT  DD SYSOUT=*"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//DSSPRINT  DD SYSOUT=*"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//COPY001   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPY001,"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//COPY002   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPY002,"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//COPY003   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPY003,"
```

```
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//COPY00Ø4   DD DSN=<UID>."||STRIP(DBN,T,' ')||"."||,
STRIP(TSN,T,' ')||".COPY00Ø4,"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//             DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(5,5),RLSE),"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//             UNIT=SYSDA,VOL=SER=TSOPK1"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "//SYSIN     DD *"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "   COPY INDEX "||STRIP(TBC,T,' ')||"."||IXN
"EXECIO 1 DISKW ICPYFILE"
QUEUE "         COPYDDN (COPY00Ø1,COPY00Ø2)"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "         RECOVERYDDN (COPY00Ø3,COPY00Ø4)"
"EXECIO 1 DISKW ICPYFILE"
QUEUE "/*"
"EXECIO 1 DISKW ICPYFILE"
RETURN
END_PROCESS:
QUEUE "/*"
"EXECIO 1 DISKW DELTFILE"
"EXECIO Ø DISKR UTILFILE (FINIS"
SAY 'NUMBER OF TABLES  INCLUDED IN JOBS' TSTEPN
SAY 'NUMBER OF INDEXES INCLUDED IN JOBS' ISTEPN
EXIT Ø
```

UTILJCL

```
//<UID>R JOB    (ACCT#),'* CREATE UTILS *',
//   USER=<UID>,MSGCLASS=X,CLASS=A,NOTIFY=UID,REGION=ØM,TIME=144Ø
//*ROUTE PRINT U61
//*
//*****************************************************************
//*
//* CHANGE <UID> to valid userid to run the JCL
//* CHANGE <HLQ> to valid high level qualifier for datasets
//*
//*****************************************************************
//*
//*****************************************************************
//*  DELETE OUTPUT DATASETS CREATED IN THE NEXT STEP
//*****************************************************************
//*
//STEP00Ø1 EXEC PGM=IDCAMS
```

```
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
  DELETE '<HLQ>.<DBNAME>.UTILFILE'
/*
//*
//*****************************************************************
//*  REFORMAT THE SPUFI FILE FOR INPUT INTO THE NEXT STEP
//*****************************************************************
//*
//STEP002 EXEC PGM=IKJEFT01,REGION=4096K,PARM='%REFUTILO'
//SYSEXEC   DD DISP=SHR,DSN=<UID>.UTIL.EXEC
//SYSTSPRT  DD SYSOUT=*,DCB=BLKSIZE=121
//SYSTSIN   DD DUMMY
//UTILKEYS  DD DSN=<HLQ>.<DBNAME>.UTILKEYS.OUTLIST,DISP=SHR
//UTILFILE  DD DSN=<HLQ>.<DBNAME>.UTILFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(TRK,(1,1)),
//             DCB=(LRECL=187,BLKSIZE=9350,RECFM=FB)
//*
//*****************************************************************
//*  DELETE OUTPUT DATASETS CREATED IN THE NEXT STEP
//*****************************************************************
//*
//STEP003 EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
  DELETE '<HLQ>.<DBNAME>.UNLDFILE'
  DELETE '<HLQ>.<DBNAME>.LOADFILE'
  DELETE '<HLQ>.<DBNAME>.DELTFILE'
  DELETE '<HLQ>.<DBNAME>.REPRFILE'
  DELETE '<HLQ>.<DBNAME>.RUNSFILE'
  DELETE '<HLQ>.<DBNAME>.ICPYFILE'
  DELETE '<HLQ>.<DBNAME>.TCPYFILE'
  DELETE '<HLQ>.<DBNAME>.RCVTFILE'
  DELETE '<HLQ>.<DBNAME>.RCVIFILE'
  DELETE '<HLQ>.<DBNAME>.RBLDFILE'
  DELETE '<HLQ>.<DBNAME>.REOGFILE'
/*
//*
//*****************************************************************
//*  CREATE ALL UTILITIES FROM THE REFORMATTED INPUT DATASET
//*****************************************************************
//*
//STEP004 EXEC PGM=IKJEFT01,REGION=2048K,PARM='%UTIL'
//SYSEXEC   DD DISP=SHR,DSN=<UID>.UTIL.EXEC
//SYSTSPRT  DD SYSOUT=*,DCB=BLKSIZE=121
//SYSTSIN   DD DUMMY
//UNLDFILE  DD DSN=<HLQ>.<DBNAME>.UNLDFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
```

```
//              DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB)
//LOADFILE  DD DSN=<HLQ>.<DBNAME>.LOADFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB)
//RUNSFILE  DD DSN=<HLQ>.<DBNAME>.RUNSFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB)
//REPRFILE  DD DSN=<HLQ>.<DBNAME>.REPRFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB)
//DELTFILE  DD DSN=<HLQ>.<DBNAME>.DELTFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB)
//ICPYFILE  DD DSN=<HLQ>.<DBNAME>.ICPYFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB)
//TCPYFILE  DD DSN=<HLQ>.<DBNAME>.TCPYFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB)
//RCVTFILE  DD DSN=<HLQ>.<DBNAME>.RCVTFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB)
//RCVIFILE  DD DSN=<HLQ>.<DBNAME>.RCVIFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB)
//RBLDFILE  DD DSN=<HLQ>.<DBNAME>.RBLDFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB)
//REOGFILE  DD DSN=<HLQ>.<DBNAME>.REOGFILE,
//             DISP=(MOD,CATLG,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=32000,RECFM=FB)
//UTILFILE  DD DISP=SHR,DSN=<HLQ>.<DBNAME>.UTILFILE
```

Tim Foster
DBA
Timian Systems (USA)

NEON Systems and Defywire have announced an alliance to co-market and co-sell their Shadow technology with the Defywire Mobility Suite. The joint solution will allow customers to extend mainframe data and applications to a variety of hand-held mobile devices.

NEON's Shadow product offers the ability to create mainframe-based Web services and capture real-time mainframe events. Shadow z/Services enables J2EE or .NET developers to use Web services to service-enable mainframe CICS, IMS, and Advantage CA-IDMS applications. Shadow z/Events allows developers to utilize real-time mainframe-based business events, occurring within DB2, Adabas, IMS/DB, VSAM and CICS/TS environments, with distributed applications such as Defywire Mobility Suite.

Defywire's wireless middleware connects mobile workers with front and back office systems in real-time. With the Defywire Mobility Suite, application developers can build one design and then deliver it to mobile phones, PDAs, tablets, and laptops.

For further information contact:
NEON, 14100 Southwest Freeway, Suite 500, Sugar Land, TX 77478, USA.
Tel: (281) 491 4200.
URL: www.neonsys.com/newsroom/press_releases/2004/041207.asp.

* * *

Cognos has enhanced its support of DB2 and DB2 Information Integrator in its latest release of Cognos ReportNet for customers who have to access a diverse set of data sources including mainframe, database, and content systems.

DB2 Information Integrator extends the reach of ReportNet to a wide variety of information sources, including traditional mainframe non-DB2 data sources such as VSAM and IMS, and content repositories such as DB2 Content Manager, FileNet, and EMC/Documentum.

For further information contact:
Cognos, 3755 Riverside Drive, PO Box 9707, Station T, Ottawa, ON, Canada, K1G 4K9.
Tel: (613) 738 1440.
URL: www.cognos.com/news/releases/2004/1208.html.

* * *

SteelEye Technology has announced Version 4.3 of LifeKeeper for Windows and Version 4.0 of LifeKeeper Data Replication for Windows.

LifeKeeper provides data replication, high-availability clustering, and disaster recovery for enterprise applications in Windows 2000/2003 environments. It now provide increased protection for enterprise applications and data through enhanced system-level resource monitoring and recovery, support for iSCSI storage devices, support for data replication to multiple sites across wide-area networks for disaster recovery, and integration of the product administration functions with a central application and data availability management console.

LifeKeeper for Windows provides application protection for DB2, Exchange, SQL Server, Oracle, IIS, as well as file shares, IP addresses, and LAN Manager.

For further information contact:
SteelEye Technology, 2660 Marine Way, Suite 200, Mountain View, CA 94043, USA.
Tel: (650) 318 0108.
URL: www.steeleye.com/news/press_releases/120804.html.