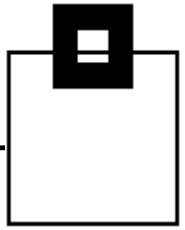
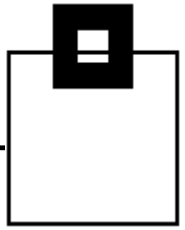


Access Path 2GO

Using RUNSTATS Rescue

Roy Boxwell

August 1st 2016

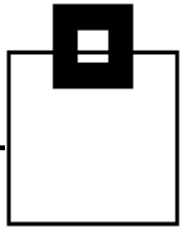


Access Path DB2GO

Using RUNSTATS Rescue

Roy Boxwell

August 1st 2016



Access Path DB2GO

Using RUNSTATS Rescue

Roy Boxwell

August 1st 2016

Agenda

- **RUNSTATS Rescue** – An overview
- First things first – Prerequisites
- How can you rescue dynamic SQL?
- How can you rescue static SQL?
  - With **RUNSTATS Rescue** for static SQL
  - With IBM's Plan Management
- Q&A

RUNSTATS Rescue – An Overview

- What is it?

  RUNSTATS Rescue is a Pocket Tool (A stand alone tool for “pocket money” or “pin money” prices). The idea is to enable a DBA to „rescue“ or „reset“ an access path back to a known good access path as quickly and simply as possible.

- Why do you need it?

  The PLAN MANAGEMENT function of DB2 REBIND is great for static SQL but fails with dynamic SQL of course.

- When do you need it?

  When an SQL access path goes belly-up obviously!

First things first - prerequisites

The first screen in RUNSTATS Rescue shows you the five possible options:

First selecting “Setup RUNSTATS Rescue” is a good idea!

First things first - prerequisites

These are the settings that can, and should, be changed. By default GDG usage is set to N, but it is highly recommended to use GDGs.

```
ImpactExpert for DB2 z/OS ----- RUNSTATS Rescue Settings ---- Setting 1 from 3
Command ===> _____________________________________________ Scroll ===> CSR
                                                            DB2: QB1A
Primary cmd: END, CAN(cel), F(ilter), T(ext on/off), L(ocate) setting
Line    cmd: S(elect), R(eset to DEFAULT)

Profile: BOXWELL      Creator . .: BOXWELL
                      Description: Default profile for IQA
    Category
      Setting                          Value     Valid Input
    ------------------------------------------------------------------------
    BIX RUNSTATS Rescue
 _ S  USE GDG FILES                    Y         Y/N

 _ S  GDG NAME                         BOXWEL..  CHAR(35)

 _ S  VSAM PREFIX FOR RUNSTATS RESCUE  BOXWEL..  CHAR(33)
    ------------------------------------------------------------------------
```

There is a sample JCL member about how to do this...

First things first - prerequisites

The next option builds the “Extract statistics” job that should be scheduled to run whenever you wish, but normally *after* the scheduled DB2 RUNSTATS jobs.

That’s it! Now just wait until some problems happen…

How can you rescue dynamic SQL?

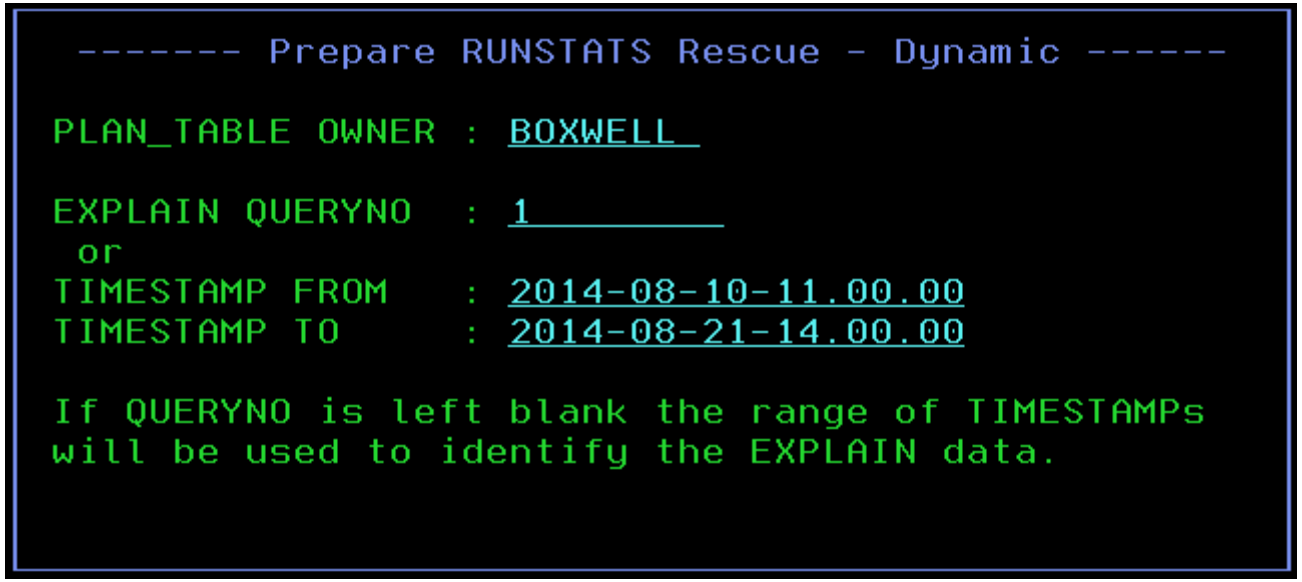
- The “problem” occurs…telephones start ringing and the feeling of “warm boss’s breath” down the back of the neck begins…all in all – Not good!
- First you find the “bad guy” using any method you have! Naturally it would be great if you were using our **SQL WorkloadExpert** product…
- Once found you simply EXPLAIN the “bad guy” and remember two things. First, the PLAN_TABLE owner just used for the EXPLAIN and second, the QUERYNO you also just used on the EXPLAIN. That is it!

How can you rescue dynamic SQL?

With these two things in mind you then select the third option on the pop-up panel:

How can you rescue dynamic SQL?

Which then leads to this pop-up:



Where you can enter the data from the EXPLAIN or even a range of times to select the data to rescue.

How can you rescue dynamic SQL?

Which then leads to the “catalog browser” panel:

```
ImpactExpert for DB2 z/OS ----- Tables of Explained SQL ------- Table 1 from 2
Command ===> _____________________________________________ Scroll ===> CSR
MODE:                                                          DB2: QB1A
Primary cmd: END, CAN(cel), Z(oom), L(ocate) creator
Line    cmd: C(olumns), D(atabase), I(ndexes), L(CoLdist), P(artitions),
             T(ablespace), Z(oom)

    Creator  +   Name              +   Database  Tablespace  Statstime          +
    ---------   ------------------   --------  ----------  -------------------
_   SYSIBM       SYSTABLES            DSNDB06   SYSTSTAB    2016-06-30-11.32.28
_   SYSIBM       SYSVIEWDEP           DSNDB06   SYSTSVWD    2016-06-16-12.14.06
    ---------   ------------------   --------  ----------  -------------------
```

This shows you all of the Database Tablespaces that will be rescued. Once finished looking around hit PF3.

How can you rescue dynamic SQL?

Which then leads to the “Last Time OK” pop-up:

First we will change the time and see what happens!

How can you rescue dynamic SQL?

Time now 09:40 and hit enter

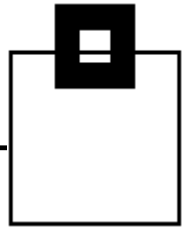
How can you rescue dynamic SQL?

The top half of the screen shows the general timestamps involved:

```
ImpactExpert for DB2 z/OS -------------------------- LINE 000000
Command ===> ____________________________________________________ Sc
Press END to continue

-------------------------------------------------------------------
Timestamp of GDG generation : 2016-08-01-09.37.53.360000
Dataset   of GDG generation : BOXWELL.RUNSTAT.RESCUE.G0004V00
Specified search timestamp  : 2016-08-01-09.40.00.000000
Determined minimum statstime: 2016-06-16-12.14.06.047427
Determined maximum statstime: 2016-06-30-11.32.28.228328
Determined maximum create TS: 2010-08-09-16.27.29.571338
-------------------------------------------------------------------
```

How can you rescue dynamic SQL?



The bottom half of the screen shows the accessed objects detailed timestamps involved:

```
Queryno :          1                EXPLAIN_TIME : 2016-08-01-09.12.20.310000

Tablespace DSNDB06.SYSTSTAB           Statstime : 2016-06-30-11.32.28.228328
Table SYSIBM.SYSTABLES                Statstime : 2016-06-30-11.32.28.228328
                                      Created   : 1985-04-01-00.00.00.000000
 - Index SYSIBM.DSNDTX01              Statstime : 2016-06-30-11.32.28.228328
    Indexspace: DSNDB06.DSNDTX01      Created   : 0001-01-01-00.00.00.000000

Tablespace DSNDB06.SYSTSVWD           Statstime : 2016-06-16-12.14.06.047427
Table SYSIBM.SYSVIEWDEP               Statstime : 2016-06-16-12.14.06.047427
                                      Created   : 1985-04-01-00.00.00.000000
 - Index SYSIBM.DSNGGX06              Statstime : 2016-06-16-12.14.06.047427
    Indexspace: DSNDB06.DSNGGX06      Created   : 2010-08-09-16.27.29.571338
```

How can you rescue dynamic SQL?

Now all of these timestamps are *older* than the “last good timestamp” so RUNSTATS Rescue actually tells you:

```
All creation timestamps and statstimes are older than the specified time
when everything was ok. It is possible that instead of rescuing older
statistics the creation of new statistics could be the solution for the
problems.
```

How can you rescue dynamic SQL?

Pressing PF3 leads to the final pop-up:

Where the automatically selected GDG data is there or you can *still* select the data manually.

How can you rescue dynamic SQL?

Pressing ENTER then generates a JCL job that creates the Rescue file to be used by the actual “rescue” job.

At the very end of the job are the objects to be rescued:

```
000126 //PDB2TSIN DD *
000127 DSNDB06.SYSTSTAB
000128 DSNDB06.SYSTSVWD
000129 //
```

How can you rescue dynamic SQL?

Once the job has finished then select the last option on the RUNSTATS Rescue to actually rescue the RUNSTATS:

How can you rescue dynamic SQL?

At the end of that job there are the “special” RUNSTATS to remove any referring statements from the Dynamic Statement Cache:

```
000074 //RUNSTATS EXEC PGM=DSNUTILB,REGION=32M,
000075 // PARM='QB1A,RSCURUNS'
000076 //STEPLIB  DD DISP=SHR,DSN=DSNB10.SDSNEXIT.QB1A
000077 //         DD DISP=SHR,DSN=DSNB10.SDSNLOAD
000078 //SYSPRINT DD SYSOUT=*
000079 //SYSIN    DD *
000080 RUNSTATS TABLESPACE DSNDB06.SYSTSTAB
000081 UPDATE NONE REPORT NO
000082
000083 RUNSTATS TABLESPACE DSNDB06.SYSTSVWD
000084 UPDATE NONE REPORT NO
000085
000086 //
```

And you are done!

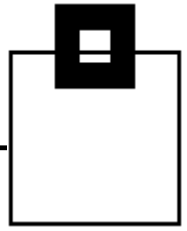
How can you rescue dynamic SQL?

Now, backing up a bit, if we go back to the “Last time OK” pop-up panel and do *not* change the time then RUNSTATS rescue does everything autonomically:

```
ImpactExpert for DB2 z/OS ------------------------- LINE 000000
Command ===> ____________________________________________________ Sc
Press END to continue

-------------------------------------------------------------------
Timestamp of GDG generation : 2016-08-01-09.41.03.300000
Dataset   of GDG generation : BOXWELL.RUNSTAT.RESCUE.G0005V00
Determined search timestamp : 2016-08-01-10.02.47.264523
Determined minimum statstime: 2016-08-01-10.02.47.264523
Determined maximum statstime: 2016-08-01-10.02.48.607188
Determined maximum create TS: 2010-08-09-16.27.29.571338
-------------------------------------------------------------------
```

How can you rescue dynamic SQL?



We also see the detailed timestamps in the review:

```
Queryno :          1                EXPLAIN_TIME : 2016-08-01-10.01.45.750000

Tablespace DSNDB06.SYSTSTAB           Statstime : 2016-08-01-10.02.47.264523
Table SYSIBM.SYSTABLES                Statstime : 2016-08-01-10.02.47.264523
                                      Created   : 1985-04-01-00.00.00.000000
 - Index SYSIBM.DSNDTX01              Statstime : 2016-08-01-10.02.47.264523
    Indexspace: DSNDB06.DSNDTX01      Created   : 0001-01-01-00.00.00.000000

Tablespace DSNDB06.SYSTSVWD           Statstime : 2016-08-01-10.02.48.607188
Table SYSIBM.SYSVIEWDEP               Statstime : 2016-08-01-10.02.48.607188
                                      Created   : 1985-04-01-00.00.00.000000
 - Index SYSIBM.DSNGGX06              Statstime : 2016-08-01-10.02.48.607188
    Indexspace: DSNDB06.DSNGGX06      Created   : 2010-08-09-16.27.29.571338
```

Hitting ENTER then puts us back at the GDG Selection Pop-up panel, ready to submit the job.

How can you rescue dynamic SQL?

In both methods you can always override the selection and then you are presented with a list of all datasets that you can use as input for the RUNSTATS Rescue job:

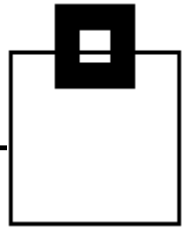
```
ImpactExpert for DB2 z/OS --- GDG Generation Overview ---- Generation 1 from 5
Command ===> _____________________________________________ Scroll ===> CSR
                                                          DB2: QB1A

Primary cmd: END, CAN(CEL), L(ocate) submit time
Line    cmd: S(elect)

 Selected GDG generation: -
                          _
 Search Timestamp       : 2016-06-16-12.14.06.047427

    Job Submit Time       GDG dataset
    ------------------    ----------------------------------------
_   2016-08-01-09.41.03   BOXWELL.RUNSTAT.RESCUE.G0005V00
_   2016-08-01-09.37.53   BOXWELL.RUNSTAT.RESCUE.G0004V00
_   2016-08-01-09.12.36   BOXWELL.RUNSTAT.RESCUE.G0003V00
_   2016-08-01-09.08.05   BOXWELL.RUNSTAT.RESCUE.G0002V00
_   2016-08-01-08.54.14   BOXWELL.RUNSTAT.RESCUE.G0001V00
    ------------------    ----------------------------------------
```

How can you rescue static SQL?



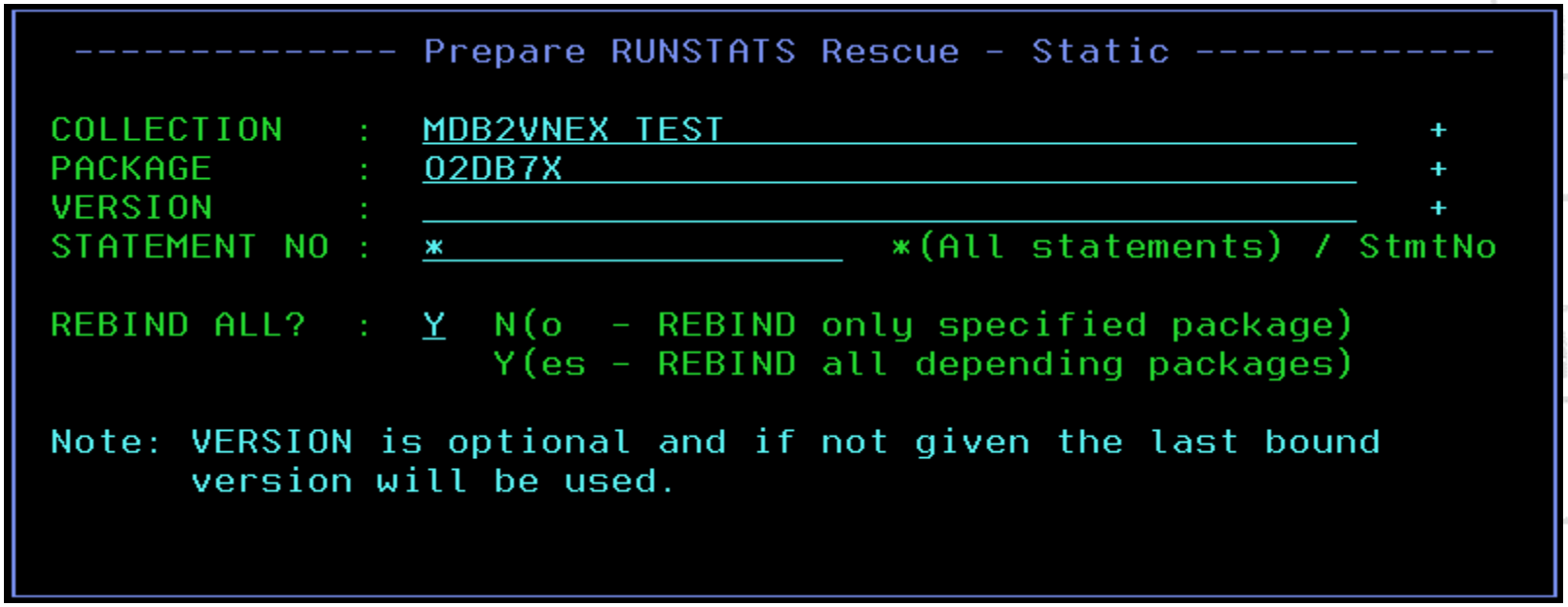
- There are two ways of rescuing static SQL:
  - First up is PLAN STABILITY (actually called PLAN MANAGEMENT, of course) which enables you to SWITCH from the CURRENT package to the PREVIOUS or even to the ORIGINAL, if you run with PLANMGMT set to EXTENDED – which, by the way, is the default in DB2 10.
  - Now this works fine – unless you happen to have inadvertently invalidated your copies. This happens very easily indeed e.g. when you DROP and recreate a VIEW or INDEX or do a column ALTER. Then the PREVIOUS or ORIGINAL are dead and cannot help you. That is when **RUNSTATS Rescue** helps you once again!

How can you rescue static SQL?

Select the fourth option on the pop-up panel:

How can you rescue static SQL?

Which then leads to the „Prepare RUNSTATS Rescue – Static“ pop-up:



Everything that then follows is the same as for Dynamic with one small difference…

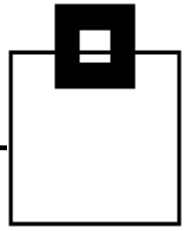
How can you rescue static SQL?

To „rescue“ Static a REBIND must be done of course:

```
000074 //REBINDPK EXEC PGM=IKJEFT01,DYNAMNBR=20
000075 //STEPLIB  DD DISP=SHR,DSN=DSNB10.SDSNEXIT.QB1A
000076 //         DD DISP=SHR,DSN=DSNB10.SDSNLOAD
000077 //SYSTSPRT DD SYSOUT=*
000078 //SYSPRINT DD SYSOUT=*
000079 //SYSTSIN  DD *
000080   DSN SYSTEM(QB1A)
000081   REBIND PACKAGE +
000082   (ANK_PCKT0310.+
000083    STATSHCB.+
000084   (2015-09-09-11.37.19.353367)) +
000085   EXPLAIN(YES) FLAG(W)
000086   REBIND PACKAGE +
000087   (BERN0510.+
000088    DSMBMAIN.+
000089   ()) +
000090   EXPLAIN(YES) FLAG(W)
```

Care must be taken with the choice „REBIND All?“

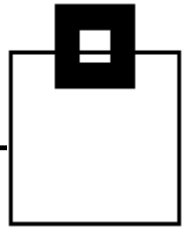
How can you rescue static SQL?



- Now going back a step…

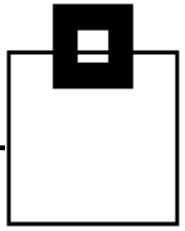
- IBM introduced PLAN MANAGEMENT (What we all really call Package Stability) back in DB2 V8 and the only thing that has changed since those Halcyon days is:
  - The SYSPACKCOPY now exists in DB2 10 NFM so once you are in NFM the SYSPACKAGE data is automatically copied into this table. It contains a maximum of two rows per package. PREVIOUS (COPYID = 1) and, if applicable, ORIGINAL (COPYID = 2). Using this table you can actually clean up your production PLAN_TABLES quite easily – But remember it is only used once a package has been rebound in DB2 10 NFM!

How can you rescue static SQL?



- PLAN MANAGEMENT uses the new REBIND control word SWITCH where you can only use PREVIOUS or ORIGINAL.
- Issuing the REBIND PACKAGE(COLL.ROYBOY.(V1)) SWITCH(PREVIOUS) will attempt to swap the previous package with the current package and, hopefully, all is well.
- Issuing the REBIND PACKAGE(COLL.ROYBOY.(V1)) SWITCH(ORIGINAL) will delete the previous package, move the current package to be the previous package and then clone the original package to be the current package and, hopefully, all is well.

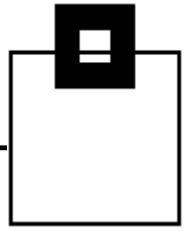
How can you rescue static SQL?



- All that is left is the PLANMGMTSCOPE control word for the FREE package command.
  - Default is ALL but you can also use INACTIVE. The difference is simply that ALL frees *all* of the packages and INACTIVE frees *only* the PREVIOUS and ORIGINAL.

- There is one restriction with PLAN MANAGEMENT in that a REBIND will *fail* if any of the following parameters are not 100% the same:
  - OWNER, QUALIFIER, [DBPROTOCOL,] ENABLE, DISABLE, PATH, PATHDEFAULT, IMMEDWRITE, and the three new xxxTIMESENSITIVE parms in DB2 11.

How can you rescue static SQL?



- Now one other REBIND option of interest was introduced in DB2 10 called APRETAINDUP which is YES or NO and defaults to YES.

- It controls whether or not wholly duplicate access path packages should be copied to SYSPACKCOPY or not.

  You should seriously consider changing this from YES to NO!

Questions???

Many thanks for your attention and now….