

Access Path Stability on Db2 for z/OS

David Simpson

dsimpson@themisinc.com

 Follow [@ThemisTraining](https://twitter.com/ThemisTraining)

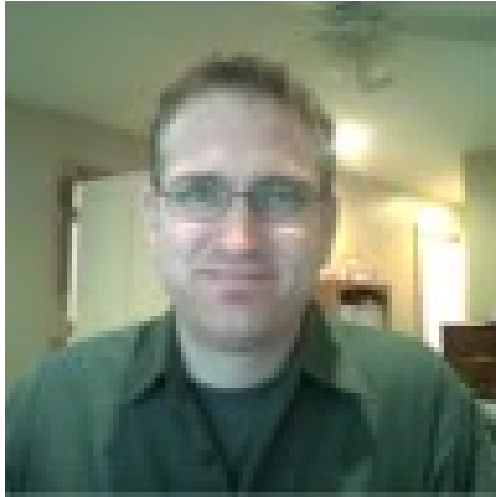


Themis Education

- Most complete DB2 Curriculum in the industry
- Offerings include a complete mainframe curriculum in addition to Oracle, Java, .NET, Linux, UNIX, etc.
- Training Venues:
 - At your facility for groups
 - Public enrollment at various locations in the USA
 - Distance learning public enrollment with live instructors.
- Webinars:
 - Visit www.themisinc.com/webinars for upcoming schedule and replays of past webinars and to download today's slides.



David Simpson



David Simpson is currently the Vice President of Themis Inc. He teaches courses on SQL, Application Programming, Database Administration as well as optimization, performance and tuning. He also installs and maintains the database systems used for training at Themis and works with our network of instructors to deliver high quality training solutions to our customers worldwide.

Since 1993 David has worked as a developer and DBA in support of very large transactional and business intelligence systems. David is a certified DB2 DBA on both z/OS and LUW. David was voted Best User Speaker and Best Overall Speaker at IDUG North America 2006. He was also voted Best User Speaker at IDUG Europe 2006 and is a member of the IDUG Speakers Hall of Fame. David is also an IBM Champion for Information Management.

dsimpson@themisinc.com

www.themisinc.com

@ThemisDave

@ThemisTraining

Agenda

- Review of the history
- Plan Management
- Access Path Compare
- Access Path Reuse
- Bind / Explain Output
- Best Practices

Stability History



Access Path Stability

- DB2 9
 - DSNZPARM PLANMGMT
 - REBIND PACKAGE
 - PLANMGMT (BASIC | EXTENDED | OFF)
 - SYSPACKAGE holds current
 - REBIND PACKAGE
 - SWITCH(PREVIOUS) or SWITCH(ORIGINAL)



Access Path Stability

- DB2 10
 - SYSPACKCOPY catalog table
 - COPYID 1 = PREVIOUS
 - COPYID 2 = ORIGINAL
 - REBIND APRETAINDUP
 - Native SQL stored procedure packages
 - APCOMPARE added
 - APREUSE added
 - EXPLAIN PACKAGE statement added
 - BIND / REBIND with EXPLAIN(ONLY) added

BIND vs REBIND

Chapter 16. BIND PACKAGE (DSN)

The DSN subcommand BIND PACKAGE builds an application package. DB2 records the description of the package in the catalog tables and saves the prepared package in the directory.

Chapter 56. REBIND PACKAGE (DSN)

The DSN subcommand REBIND PACKAGE rebinds an application package when you make changes that affect the package, but have not changed the SQL statements in the program.

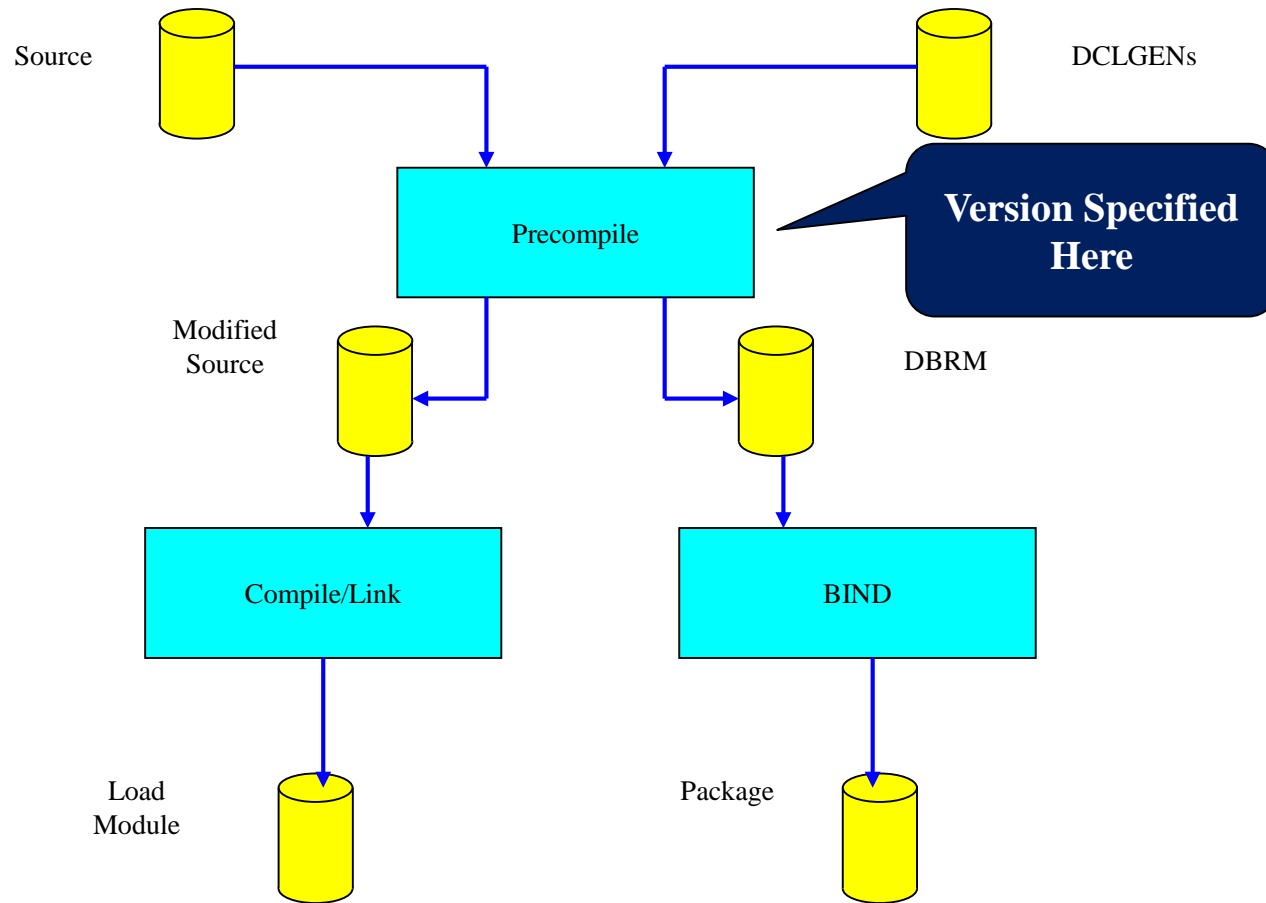
For example, you can use REBIND PACKAGE when you change the authorizations, create a new index for the package, or use RUNSTATS. When the REBIND PACKAGE(*) command is issued, trigger packages will not be affected.

REBIND PACKAGE is generally faster and more economical than BIND PACKAGE. You should use BIND PACKAGE with the ACTION(REPLACE) option under the following conditions:

- When you change the SQL statements
- When you recompile the program
- When you have previously run BIND PACKAGE with the SQLERROR(CONTINUE) option

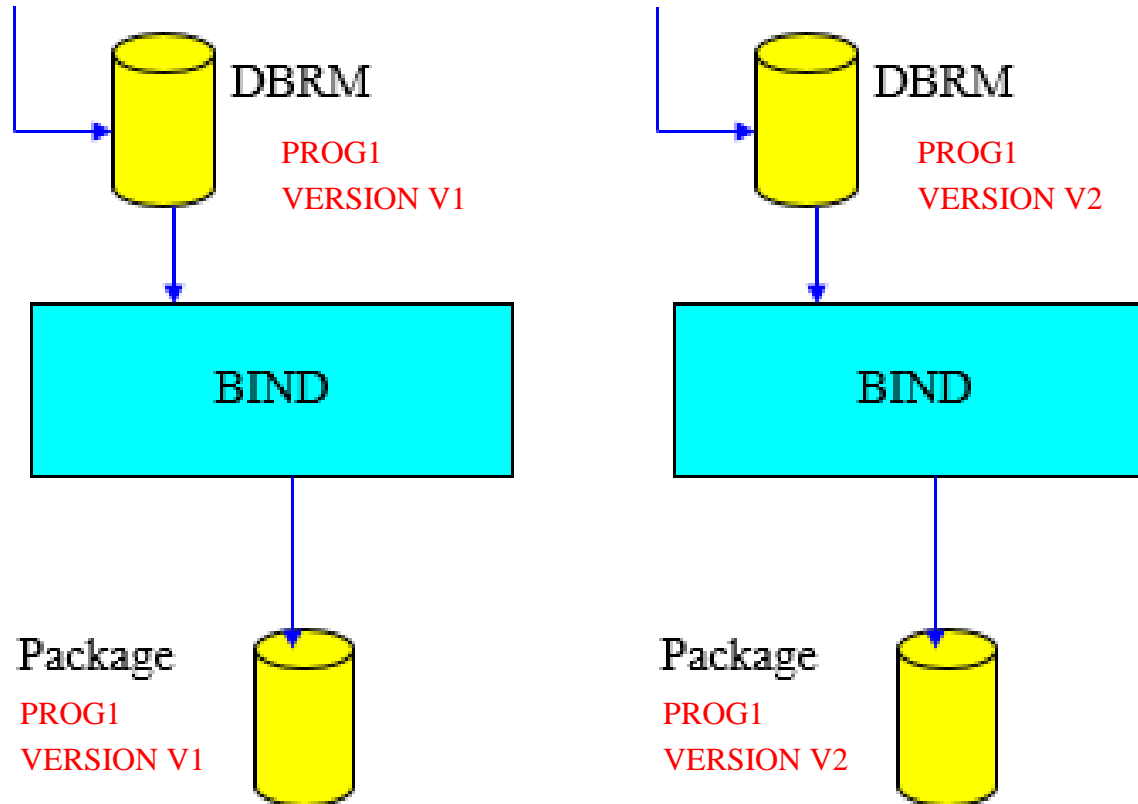
**From the
DB2 11
Command
Reference**

Program Preparation





Package Versioning



Plan Management



PLANMGMT

REBIND PACKAGE (*loc.collection.package.(version)*)
PLANMGMT(OFF)

Replace contents of package
with new control structures

Access Paths may change!



PLANMGMT

REBIND PACKAGE (*loc.collection.package.(version)*)
PLANMGMT(**BASIC**)

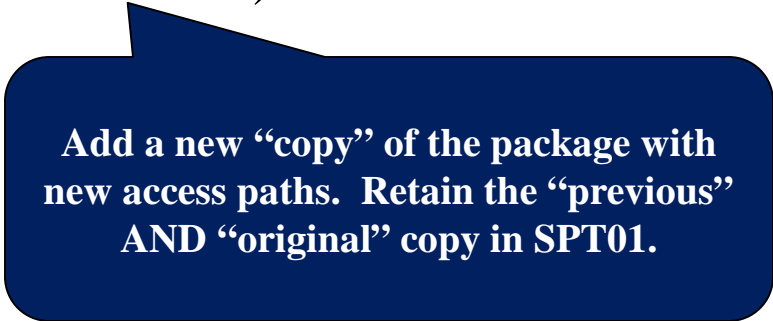
Add a new “copy” of the package with new access paths. Retain the “previous” copy in SPT01.

PLAMGMT

REBIND PACKAGE (*loc.collection.package.(version)*)
PLANMGMT(**EXTENDED**)



Default beginning
in DB2 10



Add a new “copy” of the package with
new access paths. Retain the “previous”
AND “original” copy in SPT01.

Previous: most recent copy.

Original: the oldest copy... usually the one from “bind”

*If there is no “original” when this is first done, the old
“current” will be copied to both “previous” and “original”.*

Catalog Support

- SYSIBM.SYSPACKCOPY contains rows for the “previous” and “original” copies
- A copy of SYSPACKAGE for package copies not currently in use.
- Timestamps of when they were bound also available

COPYID	INTEGER NOT NULL	The version of the copy of the package that this row explains: 1 The previous copy of the package 2 The original copy of the package
--------	---------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------



Switching between copies

REBIND PACKAGE (*loc.collection.package.(version)*)
SWITCH(**PREVIOUS**)

“Previous” becomes
current and current
becomes “previous”.

REBIND PACKAGE (*loc.collection.package.(version)*)
SWITCH(**ORIGINAL**)

“Original” becomes current
and “current” becomes
“previous”. “Original”
remains unchanged.

Considerations

- When a package goes invalid... Think about which copies of the package may be affected.
 - If a table is dropped, all copies of the dependent packages will be marked invalid.
 - If an index is dropped, only copies that included that index in an access path will be affected.
- Switching to ORIGINAL eliminates the “previous” copy.
- Please distinguish between package versions and copies. Each version may have the three copies.



FREE with PLANMGMTSCOPE

FREE PACKAGE (*loc.collid.name.(version)*)
PLANMGMTSCOPE(**ALL**)

Frees all copies of the
package (Default).

FREE PACKAGE (*loc.collid.name.(version)*)
PLANMGMTSCOPE(**INACTIVE**)

Frees only the previous
and original copies.



Duplicate Access Paths

REBIND PACKAGE (*loc.collection.package.(version)*)
PLANMGMT (EXTENDED) **APRETAINDUP(NO)**

Discard “previous” copy if
access paths did not
change.



How does it work?


- An copy of the PLAN_TABLE entries for the package is stored internally (not readable by humans) at bind time in SPT01 beginning in DB2 9.
- This allows the BIND / REBIND process to compare access paths across versions and copies.
- So many possibilities...



EXPLAIN PACKAGE

```
EXPLAIN PACKAGE  
  COLLECTION 'collection'  
  PACKAGE 'package'  
  VERSION 'version'  
  COPY 'CURRENT';
```

Externalizes the explain data
(PLAN_TABLE ONLY) for the package
into the owner's PLAN_TABLE.



Access Paths as
they exist in
the current package



EXPLAIN(ONLY)

BIND (or REBIND)
PACKAGE...

...

EXPLAIN(ONLY)

**Explain the statements
against the current
environment without
producing a package**

Compare Access Paths

Access Path Compare

BIND (or REBIND) PACKAGE...

...

APCOMPARE(NONE | WARN | ERROR)

Warning issued
if path changes

Bind fails if
any paths
change

For BIND, the comparison will be with the version being bound (if it exists) or with the most recent version available.

APCOMPARE (ERROR)

```
READY
  DSN SYSTEM(DB1C)
DSN
  REBIND PACKAGE (THEMISCL.LOTSASQ1.(2016-04-20-00.46.36.129541))
    APCOMPARE (ERROR)
DSNT285I  -BC DSNTBBP2 REBIND FOR PACKAGE = DB1C.THEMISCL.LOTSASQ1,
          USE OF APCOMPARE RESULTS IN:
          2 STATEMENTS WHERE COMPARISON IS SUCCESSFUL
          2 STATEMENTS WHERE COMPARISON IS NOT SUCCESSFUL
          0 STATEMENTS WHERE COMPARISON COULD NOT BE PERFORMED.
DSNT233I  -BC UNSUCCESSFUL REBIND FOR
          PACKAGE = DB1C.THEMISCL.LOTSASQ1.(2016-04-20-00.46.36.129541)
DSN
END
```

Evaluating the Failures

- REBIND the package with EXPLAIN(ONLY)
 - Generates the PLAN_TABLE entries for the “new” access paths
- Run the EXPLAIN PACKAGE statement
 - Generates the PLAN_TABLE entries for the “old” access paths
- Compare and evaluate

APCOMPARE(WARN)

```
READY
  DSN SYSTEM(DB1C)
DSN
  REBIND PACKAGE(THEMISCL.LOTSASQ1.(2016-04-20-00.46.36.129541))
    APCOMPARE(WARN)
DSNT285I  -BC DSNTBBP2 REBIND FOR PACKAGE = DB1C.THEMISCL.LOTSASQ1,
          USE OF APCOMPARE RESULTS IN:
          2 STATEMENTS WHERE COMPARISON IS SUCCESSFUL
          2 STATEMENTS WHERE COMPARISON IS NOT SUCCESSFUL
          0 STATEMENTS WHERE COMPARISON COULD NOT BE PERFORMED.
DSNT254I  -BC DSNTBRB2 REBIND OPTIONS FOR
          PACKAGE = DB1C.THEMISCL.LOTSASQ1.(2016-04-20-00.46.36.129541)
...
DSNT232I  -BC SUCCESSFUL REBIND FOR
          PACKAGE = DB1C.THEMISCL.LOTSASQ1.(2016-04-20-00.46.36.129541)
```

Access Path Reuse



Access Path Reuse

BIND (or REBIND) PACKAGE...

...

APREUSE(NONE | ERROR | **WARN**)

New in DB2 11
Warn if reuse
not possible

Force reuse of previous
access paths. Bind error
occurs if this is not possible

For BIND, the comparison will be with the version being bound (if it exists) or with the most recent version available.

Access Path Reuse

- For BIND, the comparison will be with the version being bound (if it exists)
- or with the most recent version available
 - DSNT292I message is issued
 - Because the package versions differ, it is possible that not all statements have a match.
 - APREUSE only applies to statements that are identical between the two versions. Statement numbers need not be the same.

Access Path Reuse

DSN

```
REBIND PACKAGE(THEMISCL.LOTSASQ1.(2016-04-20-00.45.48.382660))
```

APREUSE (ERROR)

```
DSNT286I -BC DSNTBBP2 REBIND FOR PACKAGE = DB1C.THEMISCL.LOTSASQ1,
```

```
USE OF APREUSE RESULTS IN:
```

```
4 STATEMENTS WHERE APREUSE IS SUCCESSFUL
```

```
0 STATEMENTS WHERE APREUSE IS EITHER NOT SUCCESSFUL
```

```
OR PARTIALLY SUCCESSFUL
```

```
0 STATEMENTS WHERE APREUSE COULD NOT BE PERFORMED
```

```
0 STATEMENTS WHERE APREUSE WAS SUPPRESSED BY OTHER HINTS.
```

```
DSNT254I -BC DSNTBRB2 REBIND OPTIONS FOR
```

```
PACKAGE = DB1C.THEMISCL.LOTSASQ1.(2016-04-20-00.45.48.382660)
```

Access Path Reuse

- “Force” is such a harsh word...
- APREUSE will pass previous plan table into the bind as an *optimization hint*
- Sometimes this doesn't work...
 - Indexes no longer available
 - Query re-write different across different DB2 versions

DB2 12 Enhancements

- FREE only “original” or “previous” copies of a package or only invalid copies
- SWITCH option will no longer allow you to switch to an invalid copy
- APREUSESOURCE option will allow the REUSE hint to be directed to the “previous” or “original”
- DSN_STATEMENT_TABLE columns will now tell you if APREUSE was effective and which package was used

Best Practices

Support for DB2 Version Migration

- Many will simply REBIND everything using APREUSE(ERROR)
- A more thorough approach...
 - REBIND everything using APCOMPARE(ERROR)
 - REBIND packages failing above using EXPLAIN(ONLY)
 - Analyze differences. For DB2 11 you will frequently see more matching columns.
- Possible role for APREUSE(WARN)

Regular Change Control

- PLANMGMT(EXTENDED) everywhere with possible use of APRETAINDUP(NO).
- APCOMPARE(WARN) *or maybe (ERROR)* in production binds.

David Simpson

Themis Training

dsimpson@themisinc.com