

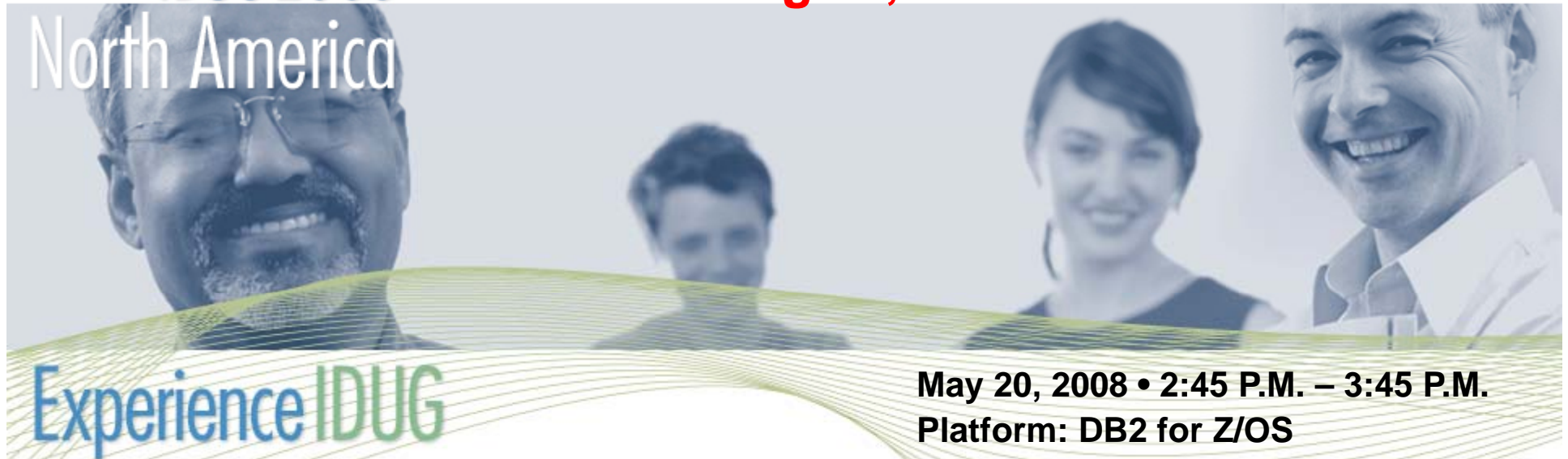


Using SPUFI to Explore DB2 Hints

Mike Bell
HLS Technologies, Inc.

IDUG 2008

North America



Experience IDUG

May 20, 2008 • 2:45 P.M. – 3:45 P.M.
Platform: DB2 for Z/OS

Session V04_15897



Using SPUFI to Explore DB2 Hints

Mike Bell

HLS Technologies, Inc.

mbell@hlstechnologies.com

Overview

The theory of hints

- The pragmatics of hints
- Requirements for using SPUFI
- SPUFI examples
- Exposure in using hints
- Our Hint Management Solutions

The Theory of Hints

- Sometimes the optimizer needs help in finding the correct index
- Sometimes the optimizer needs help in picking the correct access path
- Changing the statistics are not always an option
- The statistics changes needed for a correctly changed access path are not always known

Catch 22 in the Theory

- We are told to use query numbers to avoid the hint number changing underneath us
- The reason we have to revert to a hint is because we can not modify the originating source even to insert query numbers
- We will deal with the SQL sequence number in the examples and show a method for their use in production later

The Pragmatics of Hints

- Hints are a method of suggesting an alternative to the optimizer
- The suggestion has to be valid
- Not all access path options are accepted. e.g., MAXCOLS
- You must use a V6 or higher PLAN_TABLE
- Only one hint name is allowed in a bind
- You must bind for the hint to take effect

Requirements for using SPUFI

- The DSNZPARM for hints needs to be turned on
- The PLAN_TABLE must be at V6 or higher
- You will need to add an index to the PLAN_TABLE
- Be careful in V8 CM not to use the V8 PLAN_TABLE as you will be unable to add indexes as needed

Assumptions for our SPUFI Examples

- You are on a test or sandbox system where you may try things before working in a more production-like environment
- You either have hints turned on in your DSNZPARM or
- You have downloaded file [Hints Utility](#) from the HLS website to turn hints on and off

Unintentional Exposures in using Hints

- Most generated rebinds are unaware of the presence of hints and so rebinds drop the hint
- A bind of program changes will drop a hint unless the hint is specified in the Bind control statements
- Usually the person doing the rebinds is unaware of a hint being present
- Often the programmer making changes is unaware of a hint being present

Setting up Hints using Spufi

- SPUFI to do an INSERT
- Building an INSERT using SPUFI
- Testing your hint using SPUFI
- SPUFI to INSERT (SELECT FROM)
- What values do you have to include
- What values are ignored
- Some samples of error messages

Method 1

```
INSERT INTO P390H.PLAN_TABLE
VALUES (327          , 1          , '          '
' TEST01 ' , 1          , 0          , '          '
' PUBLIC '
' SYSCOLUMNS ' , 1          , ' I ' , 2          ,
' PUBLIC '
' DSNDX01          , ' N' , ' N' , ' N' , ' N' , ' N' , ' N' , ' N' , ' N'
' N'
' IS' , ' 2007121211104887' , ' ' , ' L' , ' ' , 0 ,
' 2007-11-017. 30. 05. 785661'
' TESTCOLLX
' SELECT' , ' 2007-12-12-11. 10. 48. 870000' ,
' PRODHI NT' , ' ' )
```

Results

```
DSNT408I  SQLCODE = -104, ERROR:  I LLEGAL SYMBOL ", ".  SOME SYMBOLS THAT
          MIGHT BE  LEGAL ARE: ( + - ? : CASE CAST NULL USER <FLOAT>
          CURRENT DEFAULT  <DECIMAL>
DSNT418I  SQLSTATE   = 42601 SQLSTATE RETURN CODE
DSNT415I  SQLERRP    = DSNHPARS SQL PROCEDURE DETECTING ERROR
DSNT416I  SQLERRD    = 0 0 0 -1 489 0 SQL DIAGNOSTIC INFORMATION
DSNT416I  SQLERRD    = X'00000000' X'00000000' X'00000000' X
          'FFFFFFF' X'000001E9' X'00000000' SQL DIAGNOSTIC INFORMATION
```

- Have to specify NULL not just multiple commas for null values

Method 2

```
INSERT INTO P390H.PLAN_TABLE
VALUES(327          , 1          , '          ' ,
' TEST01 ' , 1          , 0          ,
' PUBLIC ' ,
' SYSCOLUMNS ' , 1          , ' I ' , 2          ,
' PUBLIC ' ,
' DSNDCX01 ' , ' N ' , ' N ' , ' N ' , ' N ' , ' N ' , ' N '
, ' N ' , ' N ' , ' N ' , ' S ' , ' 2007121211104887 ' ,
' L ' , '          ' , 0          , ' 2007-11-05-17.30.05.785661 ' ,
' TESTCOLLX ' , NULL , NULL , NULL , NULL , NULL , NULL ,
NULL ,
' SELECT ' , ' 2007-12-12-11.10.48.870000 ' ,
' PRODHI NT ' , ' ' )
```

Next Result

```
DSNT408I  SQLCODE = -408, ERROR:  THE VALUE IS NOT COMPATIBLE WITH THE  
          DATA TYPE  OF ITS TARGET  
DSNT418I  SQLSTATE   = 42821 SQLSTATE RETURN CODE  
DSNT415I  SQLERRP    = DSNXOCAS SQL PROCEDURE DETECTING ERROR  
DSNT416I  SQLERRD    = -100  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION  
DSNT416I  SQLERRD    = X' FFFFFFF9C'  X' 00000000'  X' 00000000'  
          X' FFFFFFFF'  X' 00000000'  X' 00000000' SQL DIAGNOSTIC INFORMATION
```

- How many null parameters do I need to make it line up?

Aha!

```
DSNE615I NUMBER OF ROWS AFFECTED IS 1  
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
```

- And this was just to insert a row into PLAN_TABLE

I was Cheating

```
SELECT 'INSERT INTO P390H.PLAN_TABLE
VALUES(' CONCAT CHAR(QUERYNO
CONCAT CHAR(QBLOCKNO
CONCAT APPLNAME
CONCAT PROGNAME
CONCAT CHAR(PLANNO
CONCAT CHAR(METHOD
CONCAT CREATOR
CONCAT TNAME
```

- And continued with Columns and Concat

Select from PLAN_TABLE

```
SELECT FROM P390H.PLAN_TABLE
WHERE APPLNAME = ''
AND COLLID    = 'TESTCOLLX'
AND PROGRAM  = 'TEST01'
AND QUERYNO  = 327
```

Some Comments on Building the Insert with SPUFI

- The default value for a text literal is VARCHAR
- Means the maximum length is 255

```
DSNE612I DATA FOR COLUMN HEADER COLUMN NUMBER 1 WAS TRUNCATED
```

- Even with V7 the generated I Nsert statement was over 512 bytes
- Broke it into 3 SELECT statements
- And put it together with edit

Other Alternatives

```
CAST(' INSERT INTO P390H.PLAN_TABLE VALUES(' AS VARCHAR (1024))
```

- Just for the first item in the SELECT
- Makes the whole string LONG VARCHAR
- Then you get to set the limit for SPUFI

```
11 MAX CHAR FIELD .. ==> 255 (Maximum width for character fields)
```

- Or use DSNTI AUL

Testing the Hint

```
SET CURRENT OPTIMIZATION HINT = 'PRODHINT';  
EXPLAIN PLAN SET QUERYNO = 327  
FOR SELECT NAME, TBNAME, COLNO, COLTYPE, LENGTH FROM  
PUBLIC.SYSCOLUMNS WHERE TBNAME = ?  
;
```

OOPS!

```
DSNT404I  SQLCODE = 20007, WARNING:  USE OF OPTIMIZATION HINTS IS
DISALLOWED BY A DB2 SUBSYSTEM PARAMETER. THE SPECIAL
REGISTER 'OPTIMIZATION HINT' IS SET TO THE DEFAULT VALUE OF
BLANKS.
```

```
DSNT418I  SQLSTATE      = 01602 SQLSTATE RETURN CODE
```

```
DSNT415I  SQLERRP      = DSNXRS00 SQL PROCEDURE DETECTING ERROR
```

```
DSNT416I  SQLERRD      = 0  0  0  -1  0  0 SQL DIAGNOSTIC  INFORMATION
```

```
DSNT416I  SQLERRD      = X' 00000000'  X' 00000000'  X' 00000000'
X' FFFFFFFF'
X' 00000000'  X' 00000000'  SQL DIAGNOSTIC INFORMATION
```

Fix it

```
//HINTSON EXEC PGM=HINTON, PARM=DB7G  
//STEPLIB DD DISP=SHR, DSN=DSN710. SDSNEXT
```

- Available at www.hlstechnologies.com
- Or you can assemble DSNZPARM and restart DB2

Hints on

```
IEF236I  ALLOC.  FOR P390HT3 HINTSON
IEF237I  0A25  ALLOCATED TO STEPLIB
IEF142I  P390HT3 HINTSON - STEP WAS EXECUTED - COND CODE 0000
IEF285I   DSN710.SDSNEXT           KEPT
IEF285I   VOL SER NOS= Z7DB71.
```

- Not much of a report but we do receive a Condition code 0000

Retry

```
SET CURRENT OPTIMIZATION HINT = 'PRODHINT' ;  
-----+-----+-----+-----+-----+-----+-----+  
--  
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0  
-----+-----+-----+-----+-----+-----+-----+  
EXPLAIN PLAN SET QUERYNO = 327 FOR  
    SELECT NAME, TBNAME, COLNO, COLTYPE, LENGTH  
    FROM PUBLIC.SYSCOLUMNS  
    WHERE TBNAME = ?  
    ;  
-----+-----+-----+-----+-----+-----+-----+  
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
```

Why SQLCODE = 0 ?

```
Collid =DSNESP  
Proname =DSNEM68  
Queryno = what ever you test with (327)
```


Change the Insert Again

```
INSERT INTO P390H.PLAN_TABLE
VALUES(327          , 1          , '          '
' DSNESM68' , 1          , 0          , '          '
' PUBLIC'
' SYSCOLUMNS' , 1          , 'I' , 2          ,
' PUBLIC'
' DSNDCX01' , 'N' , 'N' , 'N' , 'N' , 'N' , 'N' , 'N' , 'N' , 'N' ,
' IS' , '2007121211104887' , 'L' , '          ' , 0
' DSNESPCS' , '          ' , '          ' , NULL , NULL ,
NULL , NULL , NULL , NULL , NULL , NULL
'          ' , '          ' , '          ' , 'SELECT' , '2007-12-12-11.10.48.870000' ,
' PRODHI NT' , '          ' )
```

Finally!

```
DSNT404I  SQLCODE = 394, WARNING:  USER SPECIFIED OPTIMIZATION HINTS
        USED DURING ACCESS PATH SELECTION
DSNT418I  SQLSTATE      = 01629 SQLSTATE RETURN CODE
DSNT415I  SQLERRP      = DSNXOPTH SQL PROCEDURE DETECTING ERROR
DSNT416I  SQLERRD      = 20  0  0  1141936894  0  0 SQL DIAGNOSTIC
        INFORMATION
DSNT416I  SQLERRD      = X' 00000014'  X' 00000000'  X' 00000000'
        X' 441092FE' X' 00000000'  X' 00000000' SQL DIAGNOSTIC INFORMATION
```

DB2 Validates that the Hint is Intended for this Program

- PROGNAME, APPLNAME, COLLID, VERSION, QUERYNO, QBLOCKNO, OPTHINT
- If you intend to do many hints, it would make a good index

Check the Hint

COLLID	PROGNAME	QUERYNO	OPHI NT	HI NT_USED
DSNESPCS	DSNESM68	327		PRODHI NT
DSNESPCS	DSNESM68	327	PRODHI NT	

Changed the Correlation Name to Table1

```
EXPLAIN ALL FOR
  SELECT NAME, TBNAME, COLNO, COLTYPE, LENGTH
  FROM PUBLIC.SYSCOLUMNS TABLE1
  WHERE TBNAME = ? QUERYNO 327
  ;
-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNT404I  SQLCODE = 394, WARNING:  USER SPECIFIED OPTIMIZATION HINTS USED DURING
ACCESS PATH SELECTION
```

- Still used the hint

Changed the Where Clause

```
EXPLAIN ALL FOR
  SELECT NAME, TBNAME, COLNO, COLTYPE, LENGTH
  FROM PUBLIC.SYSCOLUMNS TABLE1
  WHERE NAME = ?
  QUERYNO 327
  ;
-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNT404I  SQLCODE = 394, WARNING:  USER SPECIFIED OPTIMIZATION HINTS USED
DURING ACCESS PATH SELECTION
```

- It used the hint even though the index for the hint does Not contain that column

What Happened?

MATCHCOLS

+-----+-----+-----
0 PUBLIC DSNDCX01

- Instead of the hint

2 PUBLIC DSNDCX01

- You can use the hint to force DB2 to use a specific index but you can Not force match columns from what the optimizer knows

Another SQL to Build a Hint

```
INSERT INTO P390H.PLAN_TABLE  
  ("QUERYNO"),  
  QBLOCKNO,  
  APPLNAME,  
  PROGNAME,  
  PLANNO,  
  METHOD,  
  CREATOR,
```

- Leaving out the columns for parallel processing because they can be null

The SELECT Portion

```
SELECT  
  "QUERYNO",  
  QBLOCKNO,  
  APPLNAME
```

-etc leaving out the columns for parallel processing because they can be null

```
FROM PUBLIC.PLAN_TABLE  
  WHERE APPLNAME = ''  
  AND COLLID = 'TESTCOLLX'  
  AND PROGRAM = 'TEST01'  
  AND QUERYNO = 327  
  AND BIND_TIME = '2007-07-19-12.29.49.402261' ;
```

And it Works

```
DSNE615I NUMBER OF ROWS AFFECTED IS 1  
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
```

- Unless your hint is disposable, you may want a special `PLAN_TABLE` that doesn't get purged to keep them in

Let's Test a Bind

```
DSN SYSTEM(DB7G)
  BIND PACKAGE(TESTCOLL) MEMBER(TEST01) -
  OWNER (P390H) QUALIFIER (PUBLIC) OPTHI NT(' PRODHI NT' ) -
  ACTION(REPLACE) VALIDATE(BIND) ISOLATION(CS) EXPLAIN (YES)
  END
```

- Normal bind package with OPTHI NT specified

The Output

```
DSNT232I  -DB7G SUCCESSFUL BIND FOR  
          PACKAGE = DALLAS7.TESTCOLL.TEST01. (2007-11-05-17.30.05.785661)
```

- Not what we wanted to see
- Did not give +394 or +395 so it didn't find the hint
- More likely the hint specifies a different version identifier

So let's Fix it

```
UPDATE P390H.PLAN_TABLE
  SET VERSION = '2007-11-05-17.30.05.785661' ,
      COLLID = 'TESTCOLL'
  WHERE APPLNAME = ''
      AND COLLID = 'TESTCOLLX'
      AND BIND_TIME = '2007-07-19-12.29.49.402261'
  ;
```

```
-----+-----+-----+-----+-----+-----
DSNE615I NUMBER OF ROWS AFFECTED IS 1
```

- I had the COLLID wrong in the bind and the version

It Found the Hint

```
DSNX105I  -DB7G BIND SQL WARNING  
USING P390H AUTHORITY  
PLAN=(NOT APPLICABLE)  
DBRM=TEST01  
STATEMENT=327  
SQLCODE=395  
SQLSTATE=01628  
TOKENS=32
```

- But it didn't like it

Error Codes are listed under SQLCODE +395

- 2 TABNO is invalid.
- 3 TNAME is not specified.
- 4 TNAME is ambiguous.
- 5 TABNO doesn't agree with TNAME.
- 6 QBLOCKNO doesn't agree with TNAME.
- 7 PAGE_RANGE is invalid.
- 8 PREFETCH is invalid.
- 9 METHOD is invalid.
- 10 SORTN_JOIN is invalid.

Error Codes are listed under SQLCODE +395 continued

- 11 SORTC_JOIN is invalid.
- 12 ACESSTYPE is invalid.
- 13 ACCESSCREATOR or ACCESSNAME is invalid.
- 14 TYPE 1 index can't be used with isolation level 'UR'.
- 15 Specified index can't be used as requested.
- 16 Multi-index access can't be done.
- 17 Invalid ACESSTYPE combination.
- 18 METHOD specified for first table accessed.
- 19 Nested-loop join can't be done as requested.
- 20 Merge join can't be done as requested.

Error Codes are listed under SQLCODE +395 continued

```
21 Hybrid join can't be done as requested.  
22 PARALLELISM_MODE requested can't be done.  
23 PARALLELISM_MODE is invalid.  
24 ACCESS_DEGREE is invalid.  
25 JOIN_DEGREE is invalid.  
26 A table is missing.  
27 A table is redundant.  
28 PRIMARY_ACESSTYPE is invalid.  
29 ACCESS_PGROUP_ID is not specified.  
30 JOIN_PGROUP_ID is not specified.
```

Error Codes are listed under SQLCODE +395 continued

31 PARALLELISM_MODE is not specified.
32 CREATOR or TNAME is invalid.
33 Join sequence is incorrect.
34 Full outer join requires merge join method.
35 WHEN_OPTIMIZE is invalid or inconsistent.
99 Unexpected Error.

Our Error was 32

```
CREATOR or TNAME is invalid.
```

- They lied – it was the correlation name

```
327          PUBLIC  SYSCOLUMNS          1
327  PRODHINT  PUBLIC  SYSCOLUMNS          1  -----
```

So Fix it

```
UPDATE P390H.PLAN_TABLE
  SET CORRELATION_NAME = ' '
  WHERE APPLNAME = ' '
  AND COLLID = 'TESTCOLL'
  AND BIND_TIME = '2007-07-19-12.29.49.402261'
  ;
-----+-----+-----+-----+-----+
DSNE615I NUMBER OF ROWS AFFECTED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL,
SQLCODE IS 0
```


Finally!

```
DSNX105I  -DB7G BIND SQL WARNING  
USING P390H AUTHORITY PLAN=(NOT APPLICABLE)  
        DBRM=TEST01  
        STATEMENT=327  
        SQLCODE=394  
        SQLSTATE=01629  
        TOKENS=
```

Now Check it

```
QUERYNO  OPTHI NT  HI NT_USED
        327                PRODHI NT
        327  PRODHI NT
```

- Still have to check the actual access path

Let's review Matching

- PROGRAMNAME, APPLNAME, COLLID, VERSION, QUERYNO, QBLOCKNO, OPTHINT
- Version(AUTO) means you have to update the hint for each compile
- QUERYNO means you have to change the QUERYNO if a programmer adds an IF statement and moves the query
- QBLOCKNO means a V7 hint will NOT work when V8 does a transformation and collapses the access path
- OPTHINT only has one value for the BIND so if you have multiple hints for the same program, they have to have the same OPTHINT value

More Matching

- Table creator, name, tabno, and correlation_name have to match
- If you specify an index, the ACCESSCREATOR and ACCESSNAME have to match
- You can only have one hint in the plan_table for a given program and queryno
- If you want to change it, delete the old hint and create a new one

This is just to Make an Old Access Path into a Hint

- Did Not change any of the old values for tables or indexes
- Did Not change JOIN sequence or method

Some Columns that the Optimizer Ignores if they are in a Hint

- MATCHCOLS – you can force the optimizer to use an index but you can NOT force it to use the data in the index
- All the sort columns SORTN_UNIQ, SORTN_JOIN, SORTN_ORDERBY, SORTN_GROUPBY, SORTC_UNIQ, SORTC_JOIN, SORTC_ORDERBY, SORTC_GROUPBY
- The optimizer decides on sorts based on the SQL

Now some easier Methods

Express Hints for DB2

COMMAND ==>

Select the entity to be processed:

```
Subsystem ID . . . . . DB7G
  PLAN_TABLE owner . . . . . p390h
DBRM/package name . . . . . test01
```

Enter "/" to Select option Process packages only

Press ENTER to proceed.

So I might be a little sloppy with Test Programs

Express Hints - Item Selection

COMMAND ==>

Target DBRM : TEST01

Choose an item to process:

Type Plan or Collection

```
-----  
.   PKG   HEXCOLL  
.   PKG   TESTCOLL  
.   PKG   TESTCOLLNEW  
.   PKG   TESTCOLLX  
.   PKG   TESTCOLLXX1  
.   PKG   TESTCOLL7  
.   PKG   TESTFREE  
.   PKG   TESTFREE2  
.   PKG   TESTFREE3  
.   PKG   TESTFREE4
```


Three Old Versions

```
Express Hints - Version List
```

```
COMMAND ===>
```

```
Package (DBRM) . . . : TEST01  
Collection . . . . : TESTCOLLX
```

Choose the version to process:

```
----- Version -----  
. 2005-04-11-18.20.42.913225  
. 2005-04-11-18.34.53.754785  
. 2006-05-23-19.00.27.449848
```

Look! Queryno 327 Again

```
COMMAND ===>                               Scrol I ===>

Package . . . . . : TEST01
Col l e c t i o n . . . . . : TESTCOLL
Versi on . . . . . : 2005-04-11-18. 20. 42. 913225
```

Choose a statement to process:

```
Queryno  Statement
-----
.      154  DECLARE SYSIBM . SYSCOLUMNS TABLE ( NAME VARCHAR ( 18 ) NOT
.      225  DECLARE SYSIBM . SYSTABLES TABLE ( NAME VARCHAR ( 18 ) NOT N
.      327  DECLARE SELECT-1 CURSOR FOR SELECT NAME , TBNAME , COLNO , C
.      333  DECLARE SELECT-2 CURSOR FOR SELECT NAME , CREATOR , TYPE , D
.      444  OPEN SELECT-1
.      450  FETCH SELECT-1 INTO : H , : H , : H , : H , : H
.      462  CLOSE SELECT-1
.      466  OPEN SELECT-2
.      472  FETCH SELECT-2 INTO : H , : H , : H , : H , : H
.      484  CLOSE SELECT-2
```

Start with Option 1

```
COMMAND ==>
```

```
DBRM . . . . . : TEST01  
Collection . . . . . : TESTCOLLX  
Version . . . . . : 2005-04-11-18.20.42.913225  
Queryno . . . . . : 327
```

Select one of the options below:

- 1 MODIFY Change access path
- 2 NAME Assign hint name
- 3 MAKE Create hint

Only Accesses a Single Table

```
COMMAND ==>                               Scrol I ==>

DBRM . . . . . : TEST01
Col l e c t i o n . . . . . : TESTCOLL
V e r s i o n . . . . . : 2007-11-05-17.30.05.785661
Q u e r y n o . . . . . : 327
```

Choose a query element to edit, or specify a different sequence

```
Sequence  Table Accessed
-----  -
          1      P390H. SYSCOLUMNS
```

So let's turn off Prefetch

```
COMMAND ===>
```

```
Package . . : TEST01  
Query No. . : 327
```

Prefetch mode (PREFETCH)

1. S Sequential prefetch
2. L List prefetch
3. D Dynamic (Version 8 only)
4. Blank (no prefetch)

Type of Access (ACCESSTYPE)

1. I By the index named below
2. I1 1-fetch index scan
3. N Index scan with 'IN' predicate
4. R Table space scan
5. M Multiple index scan
6. Blank Not applicable

Back out to the Menu

Express Hints - Access Path Control

COMMAND ===>

```
DBRM . . . . . : TEST01
Collection . . . . : TESTCOLLX
Version . . . . . : 2005-04-11-18. 20. 42. 913225
Queryno . . . . . : 327
```

Select one of the options below:

1. MODIFY Change access path
2. NAME Assign hint name
3. MAKE Create hint

Now Specify the Hint Name (OPTHINT)

Express Hints - Hint Name

COMMAND ==>

Current hint name:

(none)

New hint name:

Prodhint

If you use the same Hint Name (OPTHINT Value)

Confirm Hint

```
COMMAND ===>
```

```
++ WARNING ++
```

The hint name you have specified is already used by at least one PLAN_TABLE entry. If you use this hint name, the hint may be shared by multiple queries with unintended results.

Back to the Menu to Create the Hint

Express Hints - Access Path Control

COMMAND ==>

```
DBRM . . . . . : TEST01
Collection . . . . . : TESTCOLLX
Version . . . . . : 2005-04-11-18. 20. 42. 913225
Queryno . . . . . : 327
```

Select one of the options below:

1. MODIFY Change access path
2. NAME Assign hint name
3. MAKE Create hint

What Happened?

Confirm Hint

```
COMMAND ===>
```

This Hint has been validated and accepted by DB2. If you wish, the hint can be saved in the plan table so that it may be used to bind with in the future.

Press ENTER to save the hint in the plan table.

Press END to return to the previous panel without saving the hint.

So now we have two Hints for the same Collid, Progname, and Version

```
DELETE
  FROM P390H.PLAN_TABLE
  WHERE APPLNAME = ''
  AND PROGNAME = 'TEST01'
  AND COLLID = 'TESTCOLL'
  AND TIMESTAMP = '2007071912294940'
  ;
```

Forgot to Change the Qualifier

```
DSNX105I  -DB7G BIND SQL WARNING  
          USING P390H AUTHORITY  PLAN=(NOT APPLICABLE)  
          DBRM=TEST01  
          STATEMENT=327  
          SQLCODE=395  
          SQLSTATE=01628  
          TOKENS=32
```

Another Successful Hint

```
DSNX105I  -DB7G BIND SQL WARNING  
USING P390H AUTHORITY  
PLAN=(NOT APPLICABLE)  
DBRM=TEST01  
STATEMENT=327  
SQLCODE=394  
SQLSTATE=01629  
TOKENS=
```

Hint was Used

```
QUERYNO  OPTHI NT  HI NT_USED
-----+-----+-----
327          PRODHI NT
327  PRODHI NT
And
PREFETCH
--+-----
```

- Hard to show prefetch value = spaces but?

HLS Product PATHCHK

```
//STEP EXEC PGM=PATHCHK
//SYSIN DD *
  OPTIONS CONVERTHINT PRODHINT
  CONNECT TO DB7G
  SET CURRENT SQLID = 'P390H'
  TEST DBRM TEST01 AS PACKAGE
  TESTCOLL.* IN P390H.PLAN_TABLE
```

Batch Program

```
IN  QRYNO M CREATOR  TNAME          TBNO AC MC  CREATOR ACCESSNAME
*   327  0  P390H   SYSCOLUMNS          1  R  0
PCK220I DATA CHANGE FOR COLUMN PREFETCH          WAS
PCK221I DATA CHANGE FOR COLUMN PREFETCH          IS NOW  S
PCK224I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - OLD EST SVC UNITS          209
PCK223I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST SVC UNITS          209
```


Another Lazy way to Get a Hint

```
DSNX105I  -DB7G BIND SQL WARNING  
          USING P390H AUTHORITY  PLAN=(NOT APPLICABLE)  
          DBRM=TEST01  
          STATEMENT=327  
          SQLCODE=394  
          SQLSTATE=01629  
          TOKENS=
```

Path Check

- Will preserve a hint across SQL changes
- Can even create the hint if queryno changes
- Corrects for the new VERSION(AUTO)
- Make the access path the same as the old access path
- Don't think about it

Now Recompile to get new Version

```
14.29.35 JOB03691 -JOBNAME STEPNAME PROCSTEP RC
14.29.35 JOB03691 -P390HC01 PRECOMP 00
14.29.36 JOB03691 -P390HC01 COB2COMP 00
14.29.37 JOB03691 -P390HC01 LKED2 00
14.29.37 JOB03691 -P390HC01 DB2BI ND 00
```

Building a Presentation at the Last Minute

```
DSNX105I  -DB7G BIND SQL WARNING
          USING P390H AUTHORITY PLAN=(NOT APPLICABLE)
          DBRM=TEST01
          STATEMENT=327
          SQLCODE=394
          SQLSTATE=01629
          TOKENS=
DSNT254I  -DB7G DSNTBCM2 BIND OPTIONS FOR
          PACKAGE = DALLAS7.TESTCOLL.TEST01. (2008-02-28-20.29.35.368621)
```

Notice we Aren't Using Queryno =327

```
I N QRYNO M CREATOR TNAME TBNO AC MC CREATOR ACCESSNAME
* 333 O P390H SYSCOLUMNS 1 R O
PCK220I DATA CHANGE FOR COLUMN PREFETCH WAS
PCK221I DATA CHANGE FOR COLUMN PREFETCH IS NOW S
PCK224I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - OLD EST SVC UNITS 209
PCK223I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST SVC UNITS 209
```

Created the Hint Row with the New Queryno

```
QUERYNO  OPTHI NT  HI NT_USED
-----+-----+-----
      333                PRODHI NT
      333  PRODHI NT
```

And No Hints for Parallel Access

- I am only trying to cover basic DEGREE=1 access paths
- Didn't even cover multiple step access paths. The first method has to be 0 then 1-4
- See the Presentation Titled [Implementing DB2 for z/OS Optimization Hints](#) by Patrick Bossman

Our Hint Management Solutions

- Path Check – Convert Hint will insure hints are not lost in the bind process
- Hint / Rebind – Will insure that hints are not lost in the rebind process, which is the most common source of a loss
- Express Hints – An ISPF interface that finds and sets the correct values and then does a test to see if the hint will take

You are Now a HINT Expert!

Please Come visit us at our Booth to receive your
Expert Pin!

