

00101 01011000
10000 01000101
10010 01001001

IDUG[®] North America

01000101 01011000 01010000 01000101 01010010
01000100 01010101 01000111 00100001 00100000
01001110 01000011 01000101 00100000 01001001



Experience IDUG

Session: V07

Bind and Rebind Analysis

*Mike Bell
HLS Technologies*



May 13th, 2009 • 2:45PM to 3:45PM
DB2 for z/OS

What is a *Bad Access Path*?

- *Anything that generates a midnight call.*
- *Anything that generates a “come see me” note on your voice mail.*
- *The standard answer:*
 - *Run RUNSTATS*
 - *REBIND*
 - *Repeat until solved*

How do you know the Access Path is Bad?

- *Examine the data in PLAN_TABLE.*
- *Examine the data in DSN_STATEMNT_TABLE.*
- *EXPLAIN(YES)*
- *Check SMF records for CPU used per SQL statement.*
- *Wait for a phone call.*

Procedure to get your Packages Rebound on current DB2 Version

- *REBIND COLLID(collection.*.(*))*
- *This will attempt to rebound every version of every package.*
- *It will include in that list packages that are no longer valid if they exist in the collection.*

Aid and Abet

- *SELECT DISTINCT 'REBIND PACKAGE('*
CONCAT RTRIM(COLLID)
CONCAT ').(*)'*
FROM SYSIBM.SYSPACKAGE
;
REBIND PACKAGE(ADBL).(*)*

IS 'HOPING TO GOODNESS' A VALID MIGRATION STRATEGY?

- COMPARE PACKAGE TESTCOLLX.TEST01 IN P390H.PLAN_TABLE TO PREVIOUS
- 1 ACCESS PATH FOR COLLID - TESTCOLLX .TEST01 SQL ID- OLD PLAN_TABLE -
RUN DATE 2009/05/09
- PCK212I TIMESTAMP USED FOR PREVIOUS COMMAND = 2009050715502529
- VERSION = 2005-04-11-18.20.42.913225
- BIND_TIME = 2009-05-07-15.50.25.290000
- TIMESTAMP USED FOR CURRENT = 2009050909150119
- VERSION = 2005-04-11-18.20.42.913225
- BIND_TIME = 2009-05-09-09.15.01.190000
- PCK206I COMPARE COMPLETE FOR PROGRAM = TEST01 VERSION = 2005-04-11-18.20.42.913225
- BIND-TIME = 2009-05-09-09.15.01.190000
- PREVIOUS VERSION = 2005-04-11-18.20.42.913225
- PREVIOUS BIND-TIME = 2009-05-07-15.50.25.290000
- PCK203I STATEMENTS WITH SAME ACCESS PATH 1 STATEMENTS WITH DIFFERENT
ACCESS PATH 0 FOR PROGRAM TEST01
- PCK204I STATEMENTS WITH MATCHING EXPLAIN 1 STATEMENTS WITHOUT MATCHING
EXPLAIN 0 FOR PROGRAM TEST01

EXPLAIN DBRM

- *Read a DBRM.*
- *Identify SQL that can be processed by EXPLAIN.*
- *REPORT on the access path selected.*

TEST DBRM

- *Read a DBRM.*
- *Identify SQL that can be process by EXPLAIN.*
- *COMPARE the potential access path to an existing access path.*

COMPARE

- *COMPARE two sets of access path data from PLAN_TABLE and DSN_STATEMNT_TABLE.*
- *Identify differences.*

How old are you Packages?

- *SELECT BINDYEAR ,COUNT(*) AS COUNT
FROM TABLE (SELECT COLLID, NAME,
YEAR(BINDTIME) AS BINDYEAR FROM
SYSIBM.SYSPACKAGE T1)
AS SYSPACKAGE
GROUP BY BINDYEAR ;*

SPUFI Results

BINDYEAR COUNT

-----+-----+

<i>2004</i>	<i>1</i>
<i>2006</i>	<i>448</i>
<i>2007</i>	<i>243</i>
<i>2008</i>	<i>97</i>
<i>2009</i>	<i>367</i>

Old Bind's are now hazardous to your System

- *Version 8 started the requirement to REBIND.*
- *Required rebind for packages from version 2.3 or older.*
- *Uses an internal flag from the skeleton cursor table.*
- *Version 9 continues with rebind required for version 3.1 .*

What next?

- *Depends on how you want to structure the work.*
- *Start with the oldest packages.*
- *Start with packages that are missing data in `PLAN_TABLE` or `DSN_STATEMENT_TABLE`.*
- *Work forward.*

SPUFI Results

<i>COLLID</i>	<i>NAME</i>
-----+-----+	
<i>DSNTWR81</i>	<i>DSNTWR</i>
<i>DSNUT810</i>	<i>DSNUGSQL</i>
<i>DSNUTIL</i>	<i>DSNUGSQL</i>
<i>DSNJAR</i>	<i>DSNX9INJ</i>
<i>DSNJAR</i>	<i>DSNX9RPJ</i>

SPUFI Results

<i>COLLID</i>	<i>NAME</i>
-----+-----+-----+	
<i>DSNWZP</i>	<i>DSNWZP</i>
<i>ADBL</i>	<i>ADB27AC</i>
<i>ADBL</i>	<i>ADBASW</i>
<i>DSNACC</i>	<i>DSNACCDL</i>
<i>DSNACC</i>	<i>DSNACCDS</i>
<i>DSNACC</i>	<i>DSNACCJF</i>
<i>DSNACC</i>	<i>DSNACCUC</i>
<i>DSNACC</i>	<i>DSNACCDF</i>

Use Path Check to Identify Current Access Path

- *EXPLAIN DBRM dbrmname TO PACKAGE testcoll.* in creator.PLAN_TABLE.*
- *Read the DBRM.*
- *Take explainable SQL and run EXPLAIN.*
- *Save the results in the target name.*

Use SPUFI to generate Path Check Commands

- *SELECT DISTINCT 'EXPLAIN DBRM ' CONCAT
RTRIM(NAME) CONCAT ' TO PACKAGE ' CONCAT
RTRIM(COLLID) CONCAT '.* IN P390I.PLAN_TABLE '
FROM SYSIBM.SYSPACKAGE A
WHERE CREATOR = 'P390H' AND VALID = 'Y' AND
OPERATIVE = 'Y' AND NOT EXISTS (SELECT 1 FROM
P390H.PLAN_TABLE B WHERE A.COLLID = B.COLLID
AND A.NAME = B.PROGNAME AND A.VERSION =
B.VERSION);*

SYSPRINT

PCK031I EXECUTING LICENSED PATH CHECK V3R3

```

DEBUG LISTSQL
OPTIONS SQLERROR
CONNECT TO DB8G
SET CURRENT SQLID      = 'P390H'
EXPLAIN DBRM TDBRM2 TO PACKAGE TESTEXP.* IN P390I.PLAN_TABLE

```

```

ACCESS PATH FOR          COLLID - TESTEXP          .TDBRM2  SQL ID-          OLD
PLAN_TABLE -              RUN DATE 2009/05/04

```

```

IN  QRYNO M CREATOR  TNAME          TBNO AC MC  CREATOR  ACCESSNAME          IO
SORTUJOG LK PF FN QBNO PLNO MXSQ MJN PG JN OP
      294 0 P390H     SYSCOLUMNS          1 R      0              N
NNNNNNNN  IS S      1      1          0
PCK225I DSN_STATEMNT_TABLE ESTIMATED COST - EST SVC UNITS          209
CKP228I DSN_STATEMNT_TABLE ESTIMATED COST - EST CPU              108

```

SYSEXPLN

ACCESS PATH FOR COLLID - TESTEXP .TDBRM2 SQL ID- OLD PLAN_TABLE -
RUN DATE 2009/05/04

DBRM TDBRM2 IS VERSION A000A
DBRM TDBRM2 HAS A CONNECT TOKEN OF 0C1F0F0F00C14040
DBRM TDBRM2 WAS PRECOMPILED WITH LANGUAGE IBMCOB

STATEMENT 294 THIS IS THE NEW SQL STATEMENT.

STATEMENT# 294 DECLARE SELECT-1 CURSOR FOR SELECT NAME , TBNAME , COLNO , C
OLTYPE , LENGTH FROM SYSCOLUMNS WHERE TBNAME = : H

STEP 1 ACCESSES TABLE P390H .SYSCOLUMNS
USING A TABLESPACE SCAN.
THE ACCESS WILL USE SEQUENTIAL PREFETCH
THE TIMESTAMP FOR THIS EXPLAIN IS 2009050409431656
THE TABLE HAS -1 ROWS OF 969 BYTES.
DB2 ESTIMATES THIS QUERY WILL USE 108 MILLISECONDS OF CPU TIME
DB2 ESTIMATES THIS QUERY WILL USE 209 SERVICE UNITS
DB2 WAS FORCED TO USE ESTIMATES BECAUSE OF
TABLE CARDINALITY

SYSCHG



*PCK218I PROGRAM TEST01 HAS A NEW SQL
STATEMENT*

QUERYNO 333

COLLECTION TESTCOLLX

VERSION 2005-04-11-18.20.42.913225

Use Path Check to Identify Changes

- *TEST DBRM dbrmname AS PACKAGE testcoll.* in creator PLAN_TABLE.*
- *Read the DBRM.*
- *Take explainable SQL and run EXPLAIN.*
- *Compare the results with an existing access path.*

How shall I build the Commands?

- *SELECT DISTINCT 'TEST DBRM ' CONCAT
RTRIM(NAME) CONCAT ' AS PACKAGE '
CONCAT RTRIM(COLLID) CONCAT '.* IN
P390H.PLAN_TABLE ' FROM
SYSIBM.SYSPACKAGE A

WHERE CREATOR = 'P390H' AND VALID = 'Y'
AND OPERATIVE = 'Y' AND COLLID =
'TESTCOLLX'*

One of the Generated Commands

- *TEST DBRM TEST01 AS PACKAGE TESTCOLLX.*
IN P390H.PLAN_TABLE*

Commands

CONNECT TO DB8G

SET CATALOG QUALIFIER = 'SYSIBM'

SET CURRENT SQLID = 'PUBLIC'

TEST DBRM TEST01 AS PACKAGE TESTCOLLX.
IN P390H.PLAN_TABLE*

RESULTS (NOT GOOD)

```

IN  QRYNO M CREATOR  TNAME                TBNO AC MC  CREATOR  ACCESSNAME          IO
   SORTUJOG LK PF FN QBNO PLNO MXSQ MJN PG JN OP
*   327 0 P390H     SYSCOLUMNS                1 I    2 P390H    DSNDX01              N
   NNNNNNNN  IS L      1    1      0
PCK220I DATA CHANGE FOR COLUMN PREFETCH          WAS
PCK221I DATA CHANGE FOR COLUMN PREFETCH          IS NOW  L
PCK224I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - OLD EST SVC UNITS          1
CKP227I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - OLD EST CPU                1
PCK223I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST SVC UNITS          146
CKP226I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST CPU                76

PCK206I TEST COMPLETE FOR PROGRAM = TEST01  VERSION = 2005-04-11-
      18.34.53.754785
                                PREVIOUS VERSION = 2005-04-11-
      18.34.53.754785

PCK203I STATEMENTS WITH SAME ACCESS PATH          1  STATEMENTS WITH DIFFERENT
      ACCESS PATH          1  FOR PROGRAM TEST01

PCK204I STATEMENTS WITH MATCHING EXPLAIN          2  STATEMENTS WITHOUT MATCHING
      EXPLAIN          0  FOR PROGRAM TEST01
  
```

RESULTS (BETTER)

ACCESS PATH FOR COLLID - TESTCOLLX .TEST01 SQL ID- OLD PLAN_TABLE - PLAN_TABLE
 RUN DATE 2009/05/06

PCK206I TEST COMPLETE FOR PROGRAM = TEST01 VERSION = 2006-04-24-22.45.29.330423
 PREVIOUS VERSION = 2005-04-11-18.34.53.754785

PCK203I STATEMENTS WITH SAME ACCESS PATH FOR PROGRAM TEST01	2	STATEMENTS WITH DIFFERENT ACCESS PATH	0
PCK204I STATEMENTS WITH MATCHING EXPLAIN FOR PROGRAM TEST01	2	STATEMENTS WITHOUT MATCHING EXPLAIN	0

NEXT STEP

- *Process REBIND even though there are access path changes but depend on the estimated cost in DSN_STATEMNT_TABLE to select acceptable changes*
- *OPTIONS CPUPT nn*
- *Can specify nn = 0 or some other percentage*

TWO NEW DDNAMES



```
//STEP1      EXEC PGM=PATHCHK
//STEPLIB    DD      DSN=P390H.PDS.LOADX,DISP=SHR
//           DD      DSN=DSN810.SDSNLOAD,DISP=SHR
//SYSPRINT   DD      SYSOUT=*
//SYSEXPLN   DD      SYSOUT=*
//SYSCHG     DD      SYSOUT=*
//RBINDOUT   DD      SYSOUT=*
//PBINDOUT   DD      SYSOUT=*
//DBRMIN     DD      DSN=P390H.DB2V8.DBRMLIB2,DISP=SHR
//SYSIN      DD      *
```


SAMPLE OUTPUT

```

IN  QRYNO M CREATOR  TNAME          TBNO AC MC  CREATOR  ACCESSNAME          IO
   SORTUJOG LK PF FN  QBNO PLNO MXSQ MJN PG JN OP
*   327 0 PUBLIC    SYSCOLUMNS          1 I    2 PUBLIC    DSNDCX01            N
   NNNNNNNN IS      1    1      0
PCK220I DATA CHANGE FOR COLUMN MATCHCOLS          WAS          0
PCK220I DATA CHANGE FOR COLUMN PREFETCH          WAS          S
PCK221I DATA CHANGE FOR COLUMN MATCHCOLS          IS NOW          2
PCK221I DATA CHANGE FOR COLUMN PREFETCH          IS NOW
PCK224I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - OLD EST SVC UNITS          38
CKP227I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - OLD EST CPU          20
PCK223I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST SVC UNITS          1
CKP226I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST CPU          1

PCK206I TEST COMPLETE FOR PROGRAM = TEST01  VERSION = 2006-04-24-
      22.45.29.330423
                                PREVIOUS VERSION = 2007-11-09-
      13.47.05.581274

PCK203I STATEMENTS WITH SAME ACCESS PATH          1  STATEMENTS WITH DIFFERENT
      ACCESS PATH          1  FOR PROGRAM TEST01
PCK204I STATEMENTS WITH MATCHING EXPLAIN          2  STATEMENTS WITHOUT MATCHING
      EXPLAIN          0  FOR PROGRAM TEST01
CKP219I STATEMENTS OVER CPUPCT          0
  
```

THE NEW ACCESS PATH HAS LOWER PROCMS

- *RBINDOUT* contains

*REBIND PACKAGE (TESTCOLL8.TEST01.(2006-04-
24-22.45.29.330423))-
OWNER (PUBLIC)-
EXPLAIN(YES)*

WHY DID DB2 CHANGE THE MATCHCOLS?

- *The test case has a different DBRM with a changed SQL statement*
- *So REBIND would not implement that SQL change*

NEXT POSSIBLE SOLUTION OPTIONS CONVERTHINT

- *OPTIONS CONVERTHINT hintname*
- *If there is an access path change, then build a hint for the old access path*
- *HINT's are best used as a temporary solution to let you BIND or REBIND immediately*
- *Long term maintenance of HINT's is not simple*

COMMANDS

OPTIONS CATALOGSQL NOMATCHCREATOR

OPTIONS CONVERTHINT MIKEHINT

CONNECT TO DB8G

SET CURRENT SQLID = 'PUBLIC'

TEST DBRM TEST01 AS PACKAGE TESTCOLL8.
IN P390H.PLAN_TABLE*

RESULTS

```

IN  QRYNO M CREATOR  TNAME                TBNO AC MC CREATOR  ACCESSNAME          IO SORTUJOG LK PF FN QBNO PLNO MXSQ MJN PG JN OP
*    327 0 PUBLIC    SYSCOLUMNS          1 I  2 PUBLIC    DSNDX01             N NNNNNNNN IS          1  1      0
PCK220I DATA CHANGE FOR COLUMN MATCHCOLS          WAS                0
PCK220I DATA CHANGE FOR COLUMN PREFETCH          WAS S              0
PCK221I DATA CHANGE FOR COLUMN MATCHCOLS          IS NOW              2
PCK221I DATA CHANGE FOR COLUMN PREFETCH          IS NOW              2
PCK223I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST SVC UNITS          1
PCK226I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST CPU          1
PCK052I HINT CREATED FOR PACKAGE TESTCOLL8          PROGRAM= TEST01 MIKEHINT

IN  QRYNO M CREATOR  TNAME                TBNO AC MC CREATOR  ACCESSNAME          IO SORTUJOG LK PF FN QBNO PLNO MXSQ MJN PG JN OP
INS  334 0 PUBLIC    SYSTABLES            1 I  1 PUBLIC    DSNDX01             N NNNNNNNN IS L        1  1      0
PCK223I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST SVC UNITS          5
PCK226I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST CPU          3
  
```

- HINT created for query 327
- INS means a SQL statement was inserted
- Which means the test case is not correct

Results with CPUPCT



```
IN  QRYNO M CREATOR  TNAME                TBNO AC MC  CREATOR  ACCESSNAME          IO SORTUJOG LK PF FN QBNO PLNO MXSQ MJN PG JN OP
*    327 0 PUBLIC    SYSCOLUMNS          1 I  2 PUBLIC  DSNDX01             N NNNNNNNN IS          1  1      0
PCK220I DATA CHANGE FOR COLUMN MATCHCOLS          WAS          0
PCK220I DATA CHANGE FOR COLUMN PREFETCH          WAS  S
PCK221I DATA CHANGE FOR COLUMN MATCHCOLS          IS NOW          2
PCK221I DATA CHANGE FOR COLUMN PREFETCH          IS NOW
PCK223I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST SVC UNITS          1
PCK226I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST CPU          1

IN  QRYNO M CREATOR  TNAME                TBNO AC MC  CREATOR  ACCESSNAME          IO SORTUJOG LK PF FN QBNO PLNO MXSQ MJN PG JN OP
INS   334 0 PUBLIC    SYSTABLES           1 I  1 PUBLIC  DSNDTX01            N NNNNNNNN IS I,      1  1      0
PCK223I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST SVC UNITS          5
PCK226I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST CPU          3
```

- No HINT
- One SQL statement didn't exist in the old access path data
- INS means a SQL statement was inserted

REBIND IN PBINDOUT

*REBIND PACKAGE (TESTCOLL8.TEST01.(2005-04-11-18.34.53.754785))-
OWNER (PUBLIC)-
EXPLAIN(YES)*

RESULTS



```
IN  QRYNO M CREATOR  TNAME          TBNO AC MC  CREATOR  ACCESSNAME          IO SORTUJOG LK PF FN QBNO
   PLNO MXSQ MJN PG JN OP
*    327 0 P390H    SYSCOLUMNS      1 I    2 P390H    DSNDCX01          N NNNNNNNN  IS L    1
  1      0
PCK220I DATA CHANGE FOR COLUMN PREFETCH          WAS
PCK221I DATA CHANGE FOR COLUMN PREFETCH          IS NOW  L
PCK224I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - OLD EST SVC UNITS          1
PCK227I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - OLD EST CPU          1
PCK223I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST SVC UNITS          146
PCK226I DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST CPU          76
PCK052I HINT CREATED FOR PACKAGE  TESTCOLLX          PROGRAM=  TEST01  MIKEHINT
PCK244I REBIND WRITTEN FOR DBRM WITH NEW/CHANGED SQL
```

REBIND IN PBINDOUT



REBIND PACKAGE (TESTCOLLX.TEST01.(2005-04-11-18.34.53.754785))-

OWNER (P390H)-

EXPLAIN(YES) -

OPTHINT ('MIKEHINT')

REBIND is in PBINDOUT because it required a HINT to get access path

REBINDOUT

- *Basic is write REBIND only if all SQL access paths stay the same*
- *IF OPTIONS CPUPCT is specified, then all SQL access paths are the same or the PROCMS estimated cost is within the specified boundry*

PBINDOUT

- *If OPTIONS CONVERTHINT specified, then REBIND generated with OPTHINT*
- *If OPTIONS CPUPCT specified, any PACKAGES that had at least one SQL statement with estimated PROCMS cost increase greater than cpu percentage specified.*
- *Or both*

REVIEW

- Use basic *REBINDOUT* to get *REBIND* for packages with *NO* access path changes
- Use basic *REBINDOUT* with *OPTIONS CONVERTHINT* to get *REBIND* for packages with *NO* access path changes and *HINT*'s created for *SQL* with access path changes

ANOTHER OPTION

- *OPTIONS CPUPCT2 for when you only want to REBIND packages that will have access path changes*
- *Ignore packages that do not have access path changes*
- *If the estimated cost for every SQL statement is less than the percentage specified, put the REBIND in RBINDOUT*
- *If the estimated cost for one SQL statement is greater than the percentage specified, put the REBIND in PBINDOUT*

JCL IS THE SAME

```
//STEP1 EXEC PGM=PATHCHK
//STEPLIB DD DSN=P390H.PDS.LOADX,DISP=SHR
// DD DSN=DSN810.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSEXPLN DD SYSOUT=*
//SYSCHG DD SYSOUT=*
//RBINDOUT DD SYSOUT=*
//PBINDOUT DD SYSOUT=*
//DBRMIN DD DSN=P390H.DB2V8.DBRMLIB2,DISP=SHR
//SYSIN DD *
```


OUTPUT



TEST DBRM TEST04 AS PACKAGE TESTCOLL8.* IN P390H.PLAN_TABLE

ACCESS PATH FOR	COLLID - TESTCOLL8	.TEST04	SQL ID-	OLD PLAN_TABLE - PLAN_TABLE	RUN DATE 2009/05/08
IN QRYNO M CREATOR	TNAME	TBNO AC MC	CREATOR	ACCESSNAME	IO SORTUJOG LK PF FN QBNO PLNO MXSQ MJN PG JN OP
* 327 0 PUBLIC	SYS_COLUMNS	1 I 2	PUBLIC	DSNDCX01	N NNNNNNNN IS 1 1 0
* 327 1 PUBLIC	SYSTABLES	2 I 2	PUBLIC	DSNDTX01	N NNNNNNNN IS 1 2 0
PCK220I	DATA CHANGE FOR COLUMN INDEXONLY		WAS	Y	
PCK221I	DATA CHANGE FOR COLUMN INDEXONLY		IS NOW	N	
PCK224I	DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - OLD EST SVC UNITS				1
PCK227I	DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - OLD EST CPU				1
PCK223I	DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST SVC UNITS				2
PCK226I	DSN_STATEMNT_TABLE ESTIMATED COST CHANGE - NEW EST CPU				1

PCK206I TEST COMPLETE FOR PROGRAM = TEST04 VERSION = 2006-04-07-20.37.34.203896
PREVIOUS VERSION = 2007-11-09-13.47.43.720348

PBINDOUT

- *Has all the rest of the REBIND's*
- *PBINDOUT is NOT required for CPUPCT2*
- *This is a safe method to keep your access paths current without automatic rebind*
- *Estimated cost is still susceptible to old or invalid statistics*

A SHORT DISCUSSION OF TEST CASES

- *How does the presentation show consistent access path changes*
- *Different DBRM libraries with SQL changes to create access path differences*
- *NOT what you would normally expect for REBIND*

WHAT IF DBRM'S ARE MISSING

- *Usually safe assumption the SQL in SYSSTMT or SYSPACKSTMT is valid.*
- *PATHCHECK optional clause for EXPLAIN and TEST commands 'FROM CATALOG'*
- *There is also a product that identifies these problems in mass.*
- *3 way comparison with DBRM's, load modules and SYSPACKAGE*

ENOUGH ABOUT REBIND – WHAT ABOUT BIND’S

- *BIND is not usually optional*
- *The program is being installed because of a business request or legal requirement*
- *How can BIND be made safer?*

SAME PATHCHECK FEATURES AS BEFORE

- *EXPLAIN* to review the access path before the *BIND* is processed
- *TEST* command with *OPTIONS CONVERHINT* will create a *HINT* with the old access path even if *QUERYNO* has changed
- After the *BIND*, *COMPARE* to *PREVIOUS* will identify changes in access path even if *QUERYNO* is different

Our Hint Management Solutions

- Path Check– *Convert Hint will insure hints are not lost in the bind process*
- Express Hints – *An ISPF interface that finds and sets the correct values and then does a test to see if the hint will take*

Session: # V07

REBIND and BIND Analysis

Mike Bell

HLS TECHNOLOGIES

mbell@hlstechnologies.com

www.hlstechnologies.com