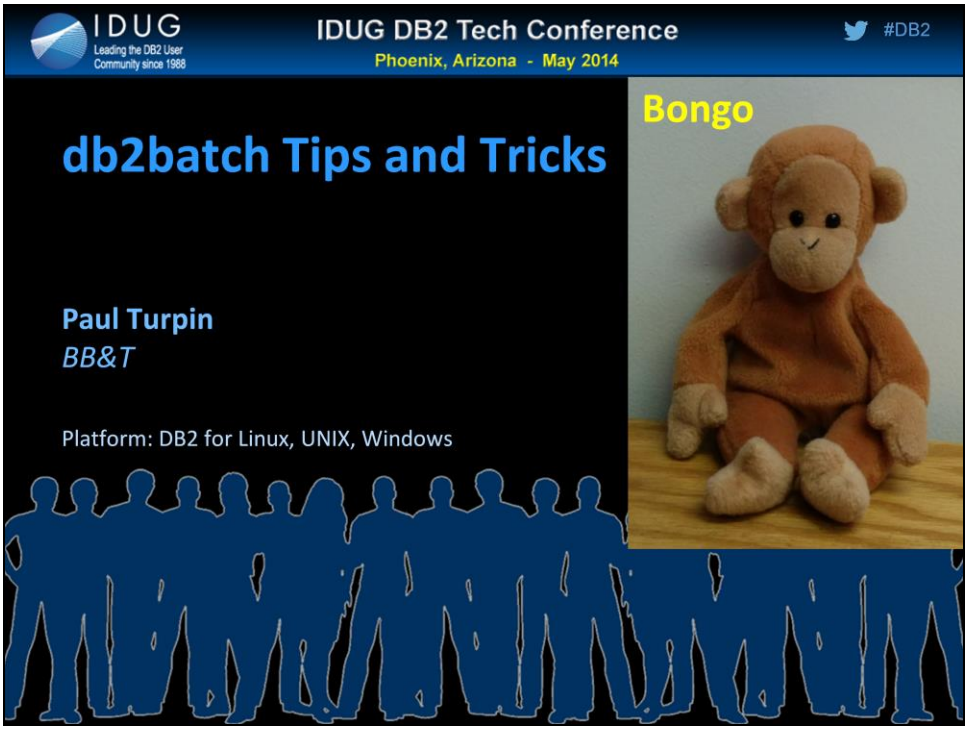


db2batch is a very powerful tool, but not always easy to use. It is my favorite performance tool and I want to share my enthusiasm about it. db2batch captures actual timings and other metrics as the query you are testing is running. Some of those metrics are SQL execution timings, plus a number of metrics including CPU time, sort information (number of sorts, sort time, sort overflows), rows read, rows written, rows updated (and inserted and deleted), rows fetched and bufferpool usage details. In addition an explain can be captured at the time of execution. db2batch can work with SQL with literals or parameter markers.

This presentation will teach you how to use the numerous db2batch parameters. It will also show the difference in using db2batch with SQL with literals and SQL with parameter markers.

Sharing Is Good

- Feel free to **tweet** or **blog** anything you find interesting in this session
- **#DB2** is a good hashtag



What We Will Cover

- Important db2batch options
- How to do a snapshot in db2batch
- How to use with literal SQL queries
- How to use with parameter marker queries
- How to do an explain in db2batch
- How to interpret db2batch output

Benchmarking tool

- Can capture many performance metrics including:
  - SQL execution timings
  - CPU time
  - Sort information (number of sorts, sort time, sort overflows)
  - Rows Read
  - Rows Written
  - Rows Updated (and inserted and deleted)
  - Rows Fetched
  - Bufferpool usage details (physical and logical) for Data, Index, temporary

Documentation

10.1

http://www-01.ibm.com/support/knowledgecenter/#!/SSEPGG_10.1.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0002043.html?cp=SSEPGG_10.1.0%2F3-6-2-6-14

9.7

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.cmd.doc/doc/r0002043.html

Example command

```
db2batch -d TEST -f coraudit1.sql -r coraudit1.sql.out
-o f 100 p 4 -msw table on -mss dbase_tables
```

Important options

-d Database

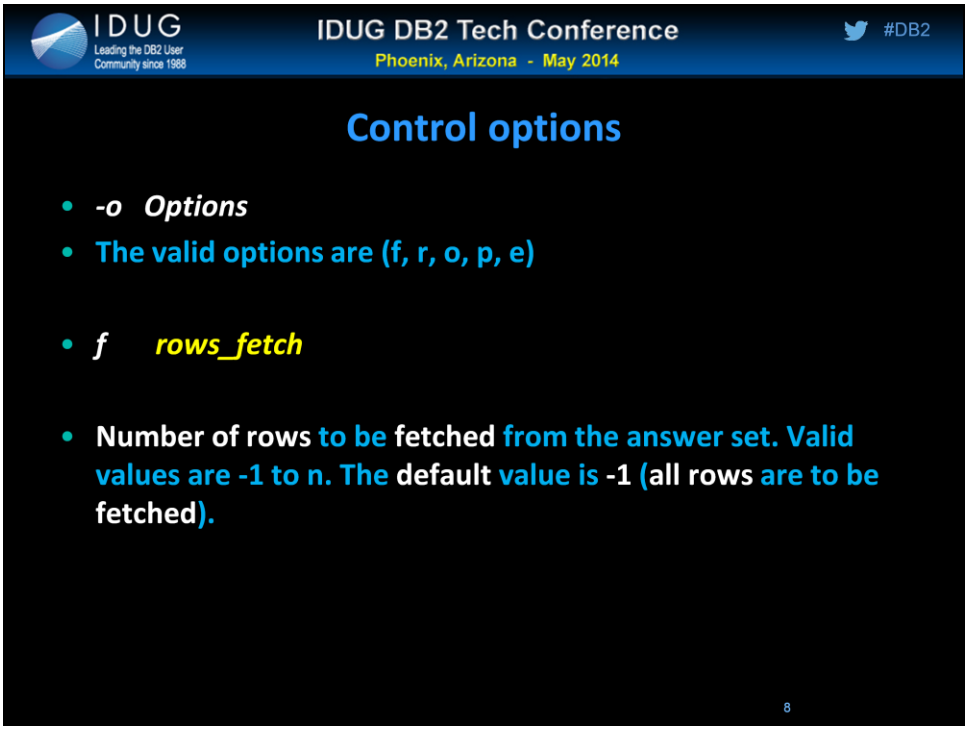
-f SQL file

-r Output file

- Can use **-r /dev/null**, so results data is not written to output file, but sent to the screen

-z Output file

- Another way to specify output file



Optimization class 9

This class directs the optimizer to use all available optimization techniques. These include:

All available statistics

All query rewrite rules

All possibilities for join enumeration, including Cartesian products and unlimited composite inners

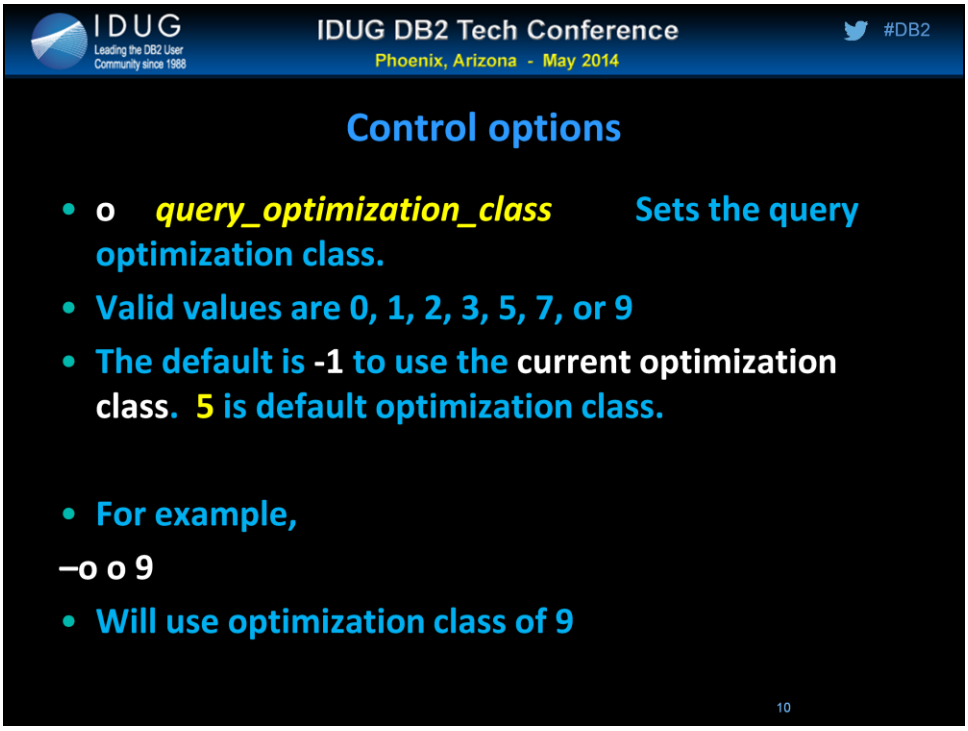
All access methods

Control options

- *r rows_out*

- Number of fetched rows to be sent to output. Valid values are -1 to n. The default value is -1 (all fetched rows are to be sent to output).

- For example,
- -o r 100
- Specifies fetch all rows and return 100 rows (f is not specified so default of all rows are fetched)



Optimization class 9

This class directs the optimizer to use all available optimization techniques. These include:

All available statistics

All query rewrite rules

All possibilities for join enumeration, including Cartesian products and unlimited composite inners

All access methods

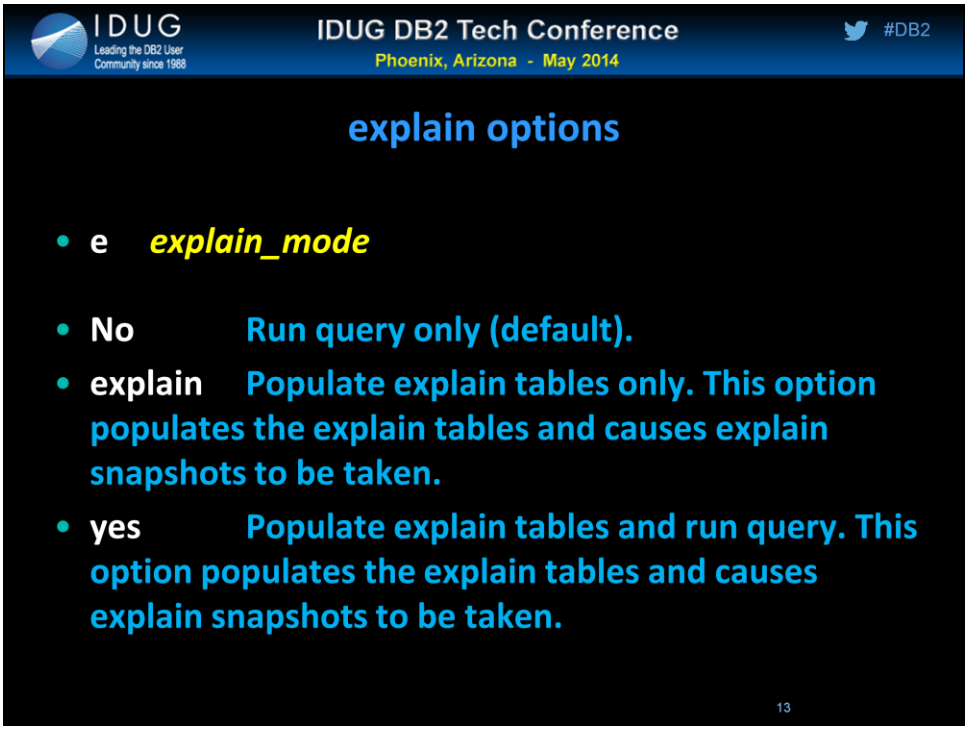
http://www-01.ibm.com/support/knowledgecenter/#!/SSEPGG_10.1.0/com.ibm.db2.luw.admin.perf.doc/doc/r0005278.html

perf_detail options

- *p perf_detail*
- Specifies the level of performance information to be returned. Valid values are:
- 0 Do not return any timing information or monitoring snapshots.
- 1 Return elapsed time only.
- 2 Return elapsed time and a snapshot for the application.
- 3 Return elapsed time, and a snapshot for the database manager, the database, and the application.

perf_detail options

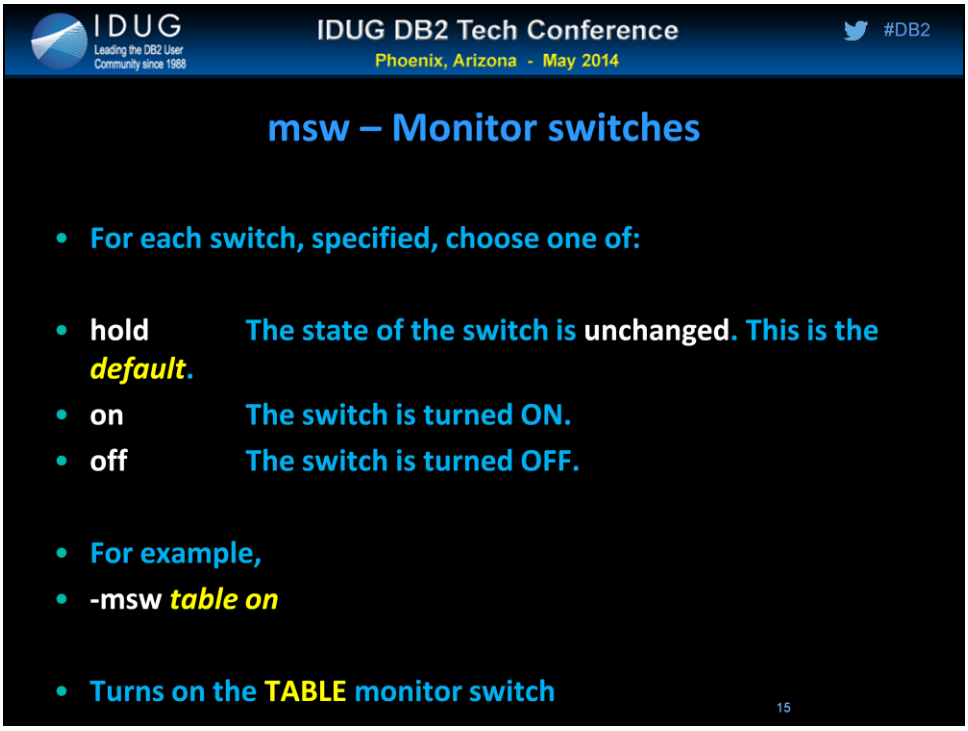
- 4 Return a snapshot for the database manager, the database, the application, and the statement (the latter is returned only if autocommit is OFF, and single statements, not blocks of statements, are being processed).
- 5 Return a snapshot for the database manager, the database, the application, and the statement (the latter is returned only if autocommit is OFF, and single statements, not blocks of statements, are being processed). Also return a snapshot for the buffer pools, table spaces and FCM (an FCM snapshot is only available in a multi-database-partition environment).
- Default value is 1.



http://www-01.ibm.com/support/knowledgecenter/#!/SSEPGG_10.1.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0002043.html?cp=SSEPGG_10.1.0%2F3-6-2-6-14

msw – Monitor switches

- -msw *switch*
- Sets the state of each specified monitor switch. You can specify any of the following: uow, statement, table, bufferpool, lock, sort, and timestamp.
- The special switch all sets all of the above switches.
- -msw *all on*
- would turn on all monitor switches



Works with uow, statement, table, bufferpool, lock, sort, and timestamp monitor switches

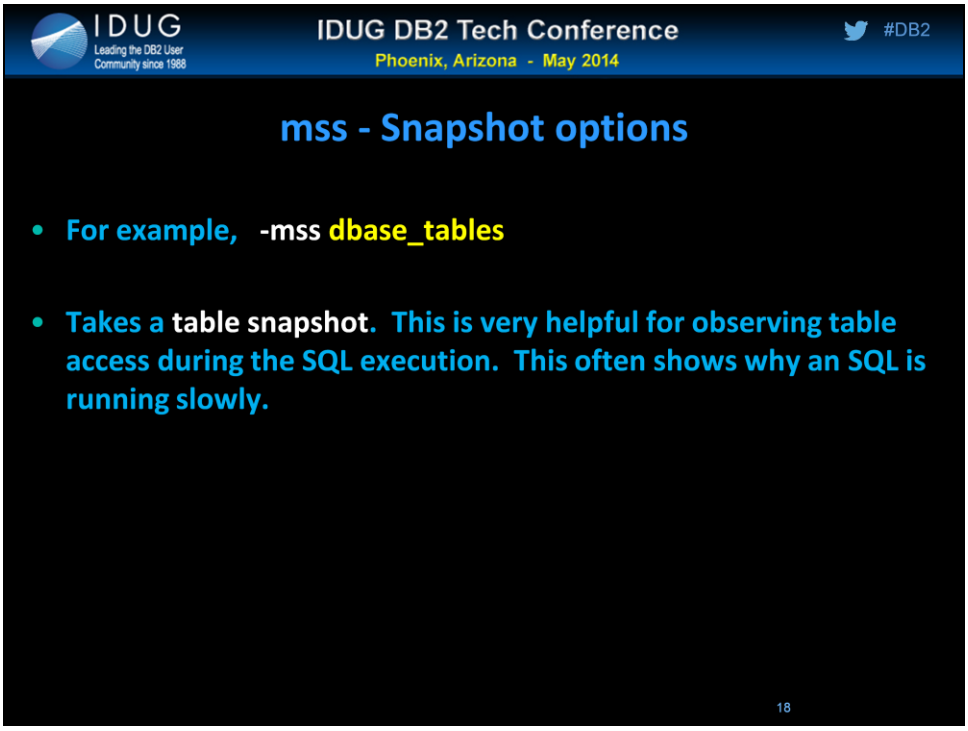
mss - Snapshot options

- **-mss *snapshot***
- **Specifies the monitoring snapshots that should be taken after each statement or block is executed**
- **More than one snapshot can be taken at a time, with the information from all snapshots combined into one large table before printing.**

mss - Snapshot options

- Possible snapshots: applinfo_all, dbase_applinfo, dcs_applinfo_all, db2, dbase, dbase_all, dcs_dbase, dcs_dbase_all, dbase_remote, dbase_remote_all, agent_id, dbase_appls, appl_all, dcs_appl_all, dcs_appl_handle, dcs_dbase_appls, dbase_appls_remote, appl_remote_all, dbase_tables, appl_locks_agent_id, dbase_locks, dbase_tablespaces, bufferpools_all, dbase_bufferpools, dynamic_sql

- The special snapshot ***all*** takes all of the above snapshots. By default, no monitoring snapshots are taken.



If you see a nested loop join in an explain this can help give details about how many loops are actually executed.

Isolation level

- -iso Isolation level
- Default level used by db2batch is RR
- Can use:
- RR – repeatable read
- RS – read stability
- CS – cursor stability
- UR – uncommitted read

19

Additional Options

- -i Specifies to measure elapsed time intervals. Valid values:
- short Measure the elapsed time to run each statement. This is the *default*.
- long Measure the elapsed time to run each statement including overhead between statements.
- complete Measure elapsed time to run each statement where prepare, execute, and fetch times are reported separately.

SQL files

- First line in the SQL will be a set schema statement
- A good practice is to repeat the SQL being tested three times in the SQL file.
- The SQL will be exactly the same.
- This is done so that the first execution with handle all the caching. Data will be put into bufferpools and if the SQL needs to be compiled and put into package cache that will be done.

Literals in SQL

- db2batch -d TEST -f test.sql -r test.sql.out -o f 100 p 4 -msw table on -mss dbase_tables
- test.sql contains SQL that only has literal values

Parameter Markers in SQL

- *-m parameters_file*
- Specifies an input file with parameter values to bind to the SQL statement parameter markers before executing a statement. The default is to not bind parameters.
- Separate parameter values with white space NOT comma (,) or semi-colon (;)
- Need to have a value for each parameter marker in the SQL

Parameter Markers in SQL

- db2batch -d RE1 -f table.sql -r table.sql.out -m table_parm.txt -o p 4 -msw table on -mss dbase_tables

- In this example,
- table_parm.txt contains the values for parameter markers that are in SQL that is in table.sql file.

Parameter Markers in SQL

- If instead #PARAM directives are used, multiple values and even parameter ranges are specified in advance for each parameter of each statement, and on each iteration of the block a random value is chosen from the specified sets for each parameter.

Parameter Markers in SQL

Values in parameter file are separated by white space (spaces) NOT commas or semi-colons

CHAR, VARCHAR, DATE, TIMESTAMP values must be enclosed in single quotes

Number types like INTEGER are not enclosed in single quotes

Parameter Markers in SQL

Here is an example of what would be in a parameter file

200000003 '0'

If you use the wrong type you will get this error:

*** CLI error, attempt to bind statement parameter marker 0 to data of a different type

27

Parameter Markers in SQL

The set schema statement needs to be first statement in SQL file.

Approach #1

In SQL file

set schema SAPECC;

Hard-code schema name

First line in the parameter file must be blank

Parameter Markers in SQL

Approach #2

In SQL file

set schema ?;

Parameter marker style

First line in the parameter file must have the schema name in quotes like 'SAPECC'

Parameter Markers in SQL

This is what you will see in the output file

Approach #1

SQL Statement Number 1:

```
set schema SAPECC;
```

Parameter Markers in SQL

This is what you will see in the output file

Approach #2
SQL Statement Number 1:

```
set schema ?;

* List of 1 supplied parameter(s):
*   'SAPECC'
```

Interpreting db2batch output

Contents of output file vary depending on options chosen.

Snapshots

-mss dbase_tables

-mss requests a snapshot and type of snapshot, like

dbase_tables determines what snapshot data is in output

Interpreting db2batch output

Data

-o options control how much data is fetched and returned

-o r 500

Specifies fetch all rows and return 500 rows

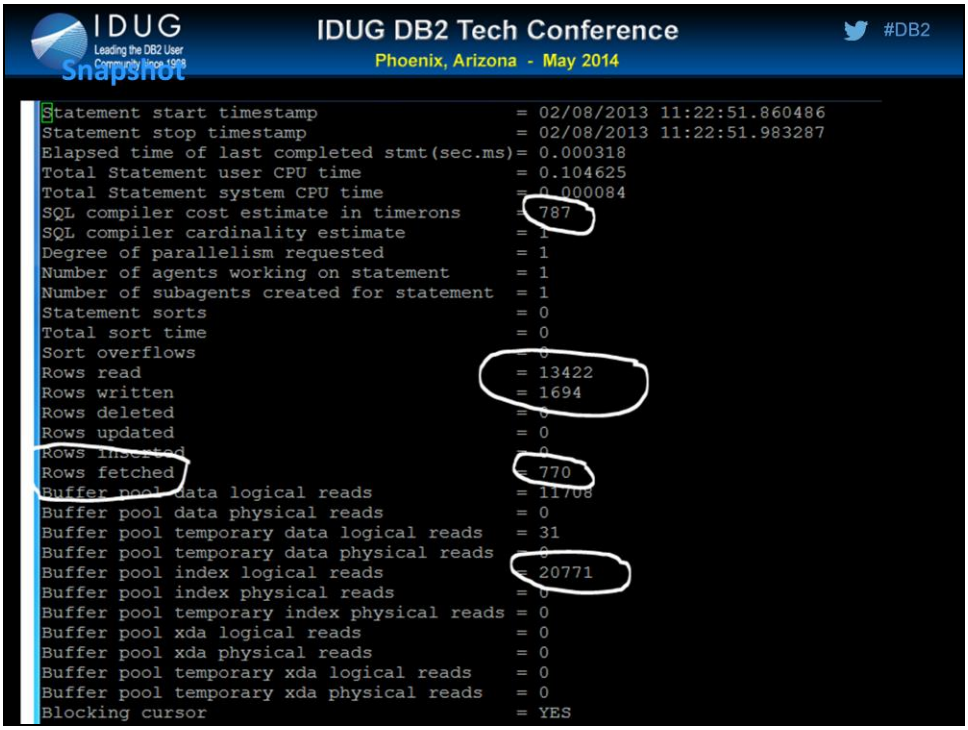
Summary Table

Always in output

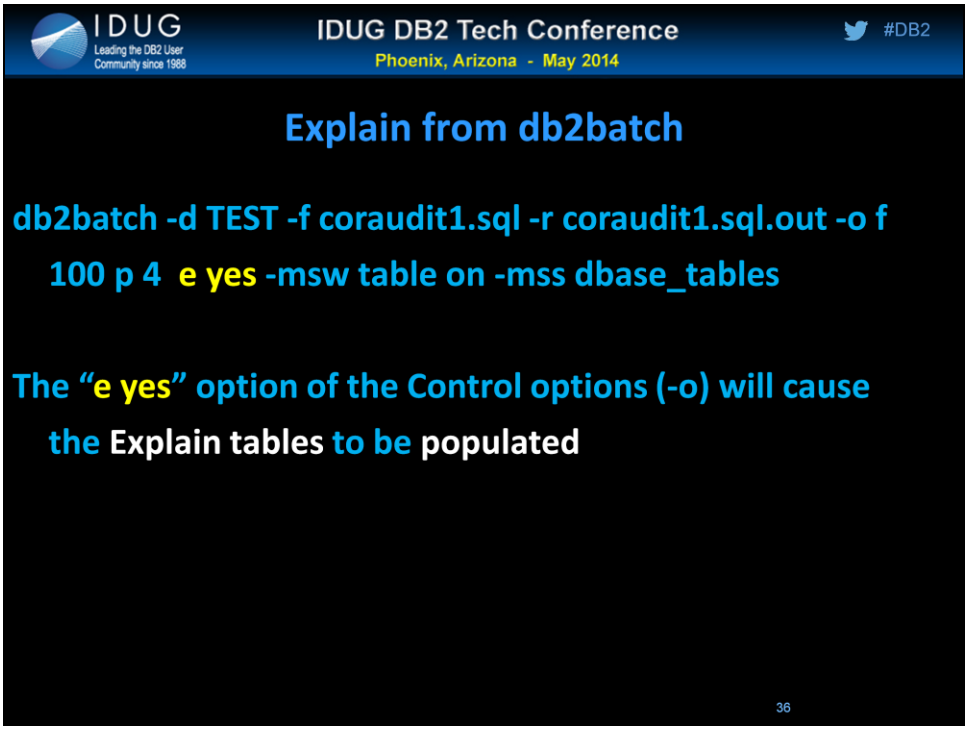
Summary table

* Summary Table:

Type	Number	Repetitions	Total Time (s)	Min Time (s)	Max Time (s)	Arithmetic Mean	Geometric Mean	Row(s) Fetched	Row(s) Output
Statement	1	1	0.000108	0.000108	0.000108	0.000108	0.000108	0	0
Statement	2	1	0.513722	0.513722	0.513722	0.513722	0.513722	318	318
Statement	3	1	0.727552	0.727552	0.727552	0.727552	0.727552	318	318
Statement	4	1	0.612861	0.612861	0.612861	0.612861	0.612861	318	318



Only selecting 5.7% of the rows that are fetch



If you use e 1 (an invalid option) it will cause strange problems – set schema statement will not work and you will get 0 for the execution times.

Explain from db2batch

Run this immediately after the db2batch command, to format the explain data

```
db2exfmt -d RE1 -g TIC -w -1 -n % -s % -# 0 -o
   coraudit1_August13.out
```

Locks snapshot

db2batch -d TEST -f coraudit1.sql -r coraudit1.sql.out -o f 100 p 4 e yes -msw table on -mss dbase_tables dbase_locks

dbase_locks will cause a lock snapshot to be taken

Search for Database Lock Snapshot in the output file to locate the locks information

38

All snapshot

db2batch -d RE1 -f coraudit1.sql -r coraudit1.sql.out -o f 100 p 4 e yes -msw table on -mss all

All will cause all snapshots to be taken. Be careful because the output will be VERY large

What We Covered

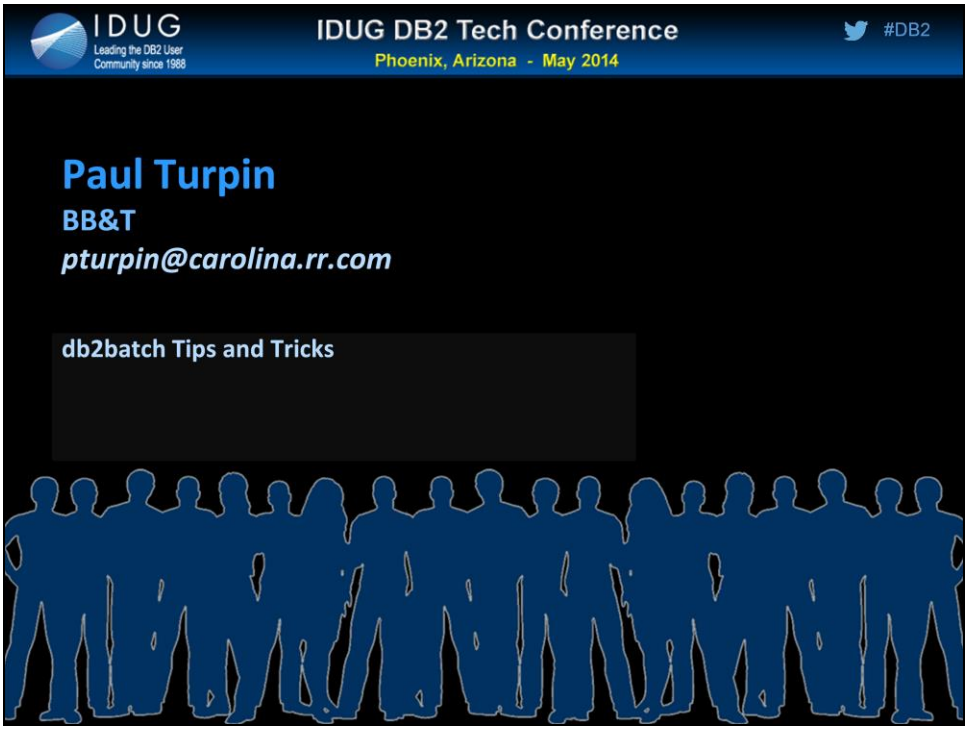
- Important db2batch options
- How to do a snapshot in db2batch
- How to use with literal SQL queries
- How to use with parameter marker queries
- How to do an explain in db2batch
- How to interpret db2batch output

db2batch – Additional Resources

http://www.dbisoftware.com/db2nightshow/20130308eric.pdf

Nice db2batch presentation done by Éric Lopes de Castro during the DB2 Got Talent competition

THANK YOU!



Paul Turpin is a consulting database administrator supporting SAP for a large financial institution. He specializes in DB2 for LUW on large systems. He enjoys exploring new features and functionality in DB2. He has spoken at IDUG North America, IDUG Europe, SHARE, IBM's Information on Demand conference, and several RUGs. Paul currently serves on the IDUG Board of Directors. He was one of IBM's inaugural Information Champions.