

# **Advanced DataTools Webcast Series**

## **Dostats and Update Statistics**

# **Dostats and Update Statistics**

**Art S. Kagel**

ASK Database Management Corp.

[www.askdbmgt.com](http://www.askdbmgt.com)

# Agenda

- A Review of all that dostats has to offer
- Using dostats to replace AUS
- Using drive\_dostats
- Best practice for Update Statistics

# Using Dostats

## General Overview

- Dostats implements the recommendations in the Performance Guide with modifications suggested by IBM engineers.
- Very flexible.
- The most powerful update statistics tool available.
- Coded in ESQL/C & runs external to the engine.
- Much less overhead and lesser effect on server processing than AUS.
- Better algorithm – Better results - \*my opinion B^)

# Using Dostats

## Important Options

- **-Q** <%PDQ>
  - Specifies the PDQPRIORITY to use when updating stats for tables.
- **-P** <%PDQ>
  - Specifies the PDQPRIORITY to use when compiling stored procedures (default=0 - recommended).
- -Q can use the AUS PDQPRIORITY threshold if -aus-thresholds is passed in.

# Using Dostats

## Important Options

- **-h** <serveralias>
  - Specifies the servername to use  
(Should be a network or stream-pipe connection type).
- **-S**
  - Enable using a shared memory connection  
(Only one database can be processed!)
- **-X**
  - SET EXPLAIN ON for the dostats session.

# Using Dostats

## Important Options

→ **-d <database spec>**

→ **-t <table spec>**

-d & -t take MATCHES wildcard specifications\* and all matching tables in all matching databases will be processed.

\* Note the behavior of -S!

# Using Dostats

## Important Options

### → **-p**

Process stored procedures! If -t is not specified then only procedures are processed.

If neither -p or -t are specified processing is equivalent to passing '-p -t "\*"'

### → **--include-owner**

- Include procedure owner when compiling SPL routines to avoid errors processing routines with names that clash with keywords.

# Using Dostats

## Important Options

- **-n** <num>

- Specifies a threshold on the number of acceptable errors before dostats aborts (default=0).

- **--error-count**

- Used with -n this causes dostats to return the number of errors encountered in the return code.

# Using Dostats

## Important Options

→ **-m**

- Enables processing stats for the system catalog tables.

→ **-w <#secs>**

- Set WAIT MODE TO <#secs>

# Using Dostats

## Important Options

- **-a**
  - Aging option. Only update stats that are older than `<n>` days old.
- **-A <#days>**
  - Specify the number of days for aging.
- Aging can use the AUS aging threshold if `-aus-thresholds` is passed in.

# Using Dostats

## Important Options

- Browsing is no longer needed if you enable `AUTO_STAT_MODE` and set an equivalent `STATCHANGE` level for the server or at the table level! In 11.70 and later the engine will no-op commands for tables that do not satisfy the table's `STATCHANGE` level. Also cannot be used with the `-S` option
- **-b** – Browsing option. Only update stats for tables that have gained or lost `<x%>` of their rows since distributions were last calculated.
- **-B** – Specify the browsing change percentage.

# Using Dostats

## Important Options

### → **--aus-thresholds**

- Dostats will use the settings that AUS uses for -Q, --small\_tables\_high, and -A and -B when -a and/or -b are specified. -A, -B, & -Q override the AUS settings if also specified.

### → Threshold names are:

- AUS\_AGE  
Default 30 days
- AUS\_CHANGE  
Default 10%
- AUS\_PDQ  
Default 10%
- AUS\_SMALL\_TABLES  
Default 100 rows

# Using Dostats

## Important Options

→ **-e**

- Print run timings per command. Report in fractional seconds.

→ **-E**

- Print run timings per command. Report in hours, minutes, and fractional seconds.

# Using Dostats

## Important Options

### → **--time-display**

- Displays the starting time for each command and block of commands in UTC. Used best with -e or -E.

### → **--display-local**

- Displays the starting time for each command and block of commands in the current timezone. Used best with -e or -E.

# Using Dostats

## Important Options

### → **--procedure=name**

- Create a stored procedure to process the UPDATE STATISTICS commands that dostats determines are required.

### → **-f <filepath>**

- Write the commands to <filepath> instead of executing them immediately.

### → **-v [-1|0|1|2]**

- Enable verbose mode.

-1 - No progress reports, 0 - default, 1 - expand options in force, 2 - additional verbosity

# Using Dostats

## Important Options

### → **--execute-local**

- Store and execute the procedure created by – procedure in each local database and execute it there.

### → **--proc-local**

- Create the table containing commands to be executed in each database rather than in sysadmin.  
(

→ These are required for mode ANSI and unlogged databases in servers earlier than 11.70, and always for servers earlier than 11.10.

# Using Dostats

## Important Options

### → **--schedule**

- Create a stored procedure to process the UPDATE STATISTICS commands that dostats determines are required and schedule it to run in the task manager. Defaults to run once only at midnight. The following options manage the schedule:

- **--frequency=d** - Run daily
- **--frequency=w<daynum>** - Run weekly on 0=Sunday, 1=Monday, etc.
- **--frequency=m<daynum>** - Run monthly on the first 0=Sunday, etc.
- **--first-time=<datetime>** - First run at the given time (hh:mm:ss)
- **--last-time=<datetime>** - Last time to run (hh:mm:ss)
- **--one-time** - Run the procedure once then disable it
- Scheduled tasks are registered in ph\_task under the task group 'DOSTATS'

# Using Dostats Esoteric Options

## → **--auto-run**

- Cause statistics to take consideration for the STATCHANGE level. Overrides AUTO\_STAT\_MODE.

## → **--force-run**

- Force statistics to be updated even if the STATCHANGE level has not been exceeded. Overrides AUTO\_STAT\_MODE.

# Using Dostats Esoteric Options

- **--isolation [c|d|l]** – Use:
  - **<c>**ommitted read,
  - **<d>**dirty read, or
  - **<l>**ast committed isolation during command execution.

# Using Dostats Esoteric Options

→ **-R** <res>

- Set the resolution for HIGH stats. <res> % of data in each HIGH bucket

# Using Dostats Esoteric Options

→ **-r** <res>

- Set the resolution for MEDIUM stats. <res> % of data in each MEDIUM bucket

→ **-c** <conf>

- Set the confidence level for MEDIUM stats. % probability that the MEDIUM distributions will be as good as HIGH. Range from 80 - 99%

→ **-Z** <rows|pct>

- Set the sampling size for MEDIUM stats. Settings less than 1.0 are treated as a percentage of rows to sample in each table. Values 1.0 or larger are treated as a number of rows to sample. (Not documented.)

# Using Dostats Esoteric Options

## → **--drop-distributions**

- Begin overall processing by deleting all existing distributions from the database for all tables  
DO NOT USE with drive\_dostats.

## → **--clean-distributions**

- Begin processing for each table by deleting distributions for that table only.

# Using Dostats Esoteric Options

## → -F

- Disable the single column index optimization. Causes single column indexes to get separate HIGH DISTRIBUTIONS ONLY and LOW commands issued.

# Using Dostats Esoteric Options

- **-i** <table>
- **-i** @<list file path>
- **-i** “!<where clause>”
- **-i** “!<select tablename ...>”

Process only the table names specified in -i arguments, the list file, where clause, or select statement.

The -t option is optional with -i. If specified, the specification passed to -t and the list of names generated by the -i options are OR'ed together. Use “-t none” or avoid -t to operate only on the tables contained in the inclusion lists.

# Using Dostats Esoteric Options

- **-x** <database>:
- **-x** <table>
- **-x** @<list file path>
- **-x** !<WHERE clause>

Force the exclusion of the named database(s) or tables listed in the -x arguments, list file, or where clause.

The -t option is optional. Use “-t '\*'” or avoid -t to apply exclusions to the list of all tables).

Can be combined with -i options but be careful!

\* Full SELECT support has been deprecated since the filters can easily be inverted and the SELECT passed to -i instead (Dec. 2011 release).

# Using Dostats Esoteric Options

## → -G

- Use count(\*) queries to get row counts rather than query SMI tables. On servers with many tables across many dbspaces this can be faster than the default method.

## → -W <secs>

- Set lock mode to wait <secs> seconds.

# Using Dostats Esoteric Options

→ **-f** <file>

- Write the commands that dostats calculates into the specified file instead of executing them immediately.

→ **-X**

- SET EXPLAIN ON during the run.

→ **-N**

- Do not use the SPECIFIC keyword for compiling stored procedures.

# Using Dostats Esoteric Options

## → -g

- Treat the columns named in fragmentation expressions as if they lead an index and update them to HIGH.

This can often improve the time it takes the optimizer to decide whether to do fragment elimination and whether to

# Using Dostats Esoteric Options

- **-L** – Run **only** LOW with DROP DISTRIBUTIONS.
- **-I** – Run **only** LOW keeping distributions.
- **-T** – Run **only** MEDIUM level statistics
- **-I** – Run **only** HIGH on the first column of indexes.
- **-C** – Run **only** LOW on index keys

Enabling any of these options disables the rest.  
By default run is equivalent of setting -T, -I, & -C.  
-L & -T are mutually exclusive. The last one passed prevails.

# Using Dostats Esoteric Options

## → **--small-tables-high**

- For small tables, just do HIGH on all columns for best quality distributions.

## → **--small-tables-threshold** <#rows>

- Specify the number of rows that constitute a “small” table.

Default: 0 (Only empty tables are processed as small.)

# Using Dostats Esoteric Options

## → **--distributions-high-file** <filepath>

- Declare a file containing columns for which HIGH distributions are to be gathered regardless of whether the column leads an index.
- Useful for columns that are used extensively in filters but are not indexed or do not lead an index especially if the column(s) are part of multiple index keys.

The file should contain one line per column with whitespace separated three fields:

- 1) database name
- 2) table name
- 3) column name

# Questions ?!?



# Using dostats to replace AUS or where AUS can't be used

→ Step one: disable the AUS tasks in the task manager.

Easy Peasey: You can do this in OAT or just go into dbaccess, OAT's SQL editor or whatever you use and run:

```
UPDATE sysadmin:ph_task  
  SET tk_enable = 'f'  
  WHERE tk_name MATCHES 'Auto *';
```

\* Obviously if you are still using a version that doesn't have the task manager, skip a few steps then get an upgrade to 11.70 or later! I will get to you in a bit.

# Using dostats to replace AUS or where AUS can't be used

→ Go to my BLOG and check out the May 5, 2011 post:

<http://informix-myview.blogspot.com>

→ You will find embedded there three ksh/bash scripts:

→ `setup_stats.ksh`

→ `evaluate_stats.ksh`

→ `initial_stats.ksh`

Mouse these into files on your system and install them in a bin directory in user informix's path.

# Using dostats to replace AUS or where AUS can't be used

- You will need to modify `setup_stats.ksh` so that it configures the run scripts
  - At the top are two variables:
    - `start_time` - the time of day to start the actual update statistics run
    - `estart_time` - the time of day to start the evaluator run
  - You want to set `estart_time` to run at the end of the normal processing day so that major changes in table contents are considered.
  - You set `start_time` to run at least an hour after that but sufficiently before the start of the next day's activities so it will complete without impacting production activity.

# Using dostats to replace AUS or where AUS can't be used

- Modify the path to find the evaluate\_stats.ksh script in the line that begins "let cmd = " to point to where you installed that script.
- Change the environment "dot" script to yours.
- You may also want to modify the -Q 100 that is passed to dostats if there will be other activity on the server at the time you run this script to set things up.
- Add --local-proc to the dostats command line if you are using v11.10 or 11.50 and your database is MODE ANSI or does not use logging.

# Using dostats to replace AUS or where AUS can't be used

→ Run `setup_stats.ksh` for each database that you want to process. It takes two arguments - database and a path to a directory into which to write log files:

```
setup_stats.ksh mydatabase /tmp/logfiles
```

```
setup_stats.ksh myotherdb /tmp/logfiles
```

# Using dostats to replace AUS or where AUS can't be used

- Make similar changes in `evaluate_stats.ksh` and `initial_stats.ksh` to the environment script lines and `dostats` command line as you did for the `setup_stats.ksh` script.
- Optionally initialize your stats to a known state by running `initial_stats.ksh` for each database:

```
initial_stats.ksh mydatabase /tmp/logfiles
```

```
initial_stats.ksh myotherdb /tmp/logfiles
```

# Using dostats to replace AUS or where AUS can't be used

- If you are running an unsupported version of Informix that does not have the task manager (so no sysadmin database) or you just prefer to use cron, you will need to use cron job scripts instead of running `setup_ksh.ksh`
  - Cron line to run the `evaluate_run_cron.ksh` script
    - Must add the `-local-proc` option to the `dostats` command line
  - Cron line to run the `dostats_run_cron.ksh` script a bit later
- \* Email me and I will send you the whole set of scripts.

# Why use dostats when we have AUS?

→ Dostats in evaluation mode (--procedure) is faster than AUS evaluator function and uses fewer resources  
On my system at home (v11.70, 4GB storage, 15 databases, >2800 tables):

- AUS run: 33.42 secs
- Dostats --procedure for all databases: 19.24 secs -- 30% faster!

→ This can result in the evaluator not completing its work before the refresh task kicks off. This can cause the refresh to fail or to not process all tables.

→ This is a MAJOR issue in v11.50 (before the evaluator task code was rewritten).

# Why use dostats when we have AUS?

→AUS uses server resources for all operations running in the CPU Vps.

→AUS is more intensive in the server: it executes thousands of independent queries (several for each table), and it runs multiple layers of stored procedures.

→Dostats is independent of the server and can even be run on a separate system.

→Dostats executes three tiny queries to configure itself to the server, one query to determine the list of databases to process, one for the list of tables & columns, and one for the list of stored procedures (one additional query if fragmentation expressions are to be examined).

# Why use dostats when we have AUS?

- AUS does not perform the LOW operation on each full index key.
  - Ongoing debate within IBM/HCL and between them and me as to whether this is required.
  - I just know that when there are performance problems and AUS is running I can often fix it just by running dostats in default mode!

# Questions ?!?



# Using drive\_dostats to make dostats runs faster

## What is drive\_dostats?

It is a shell script that breaks up the list of tables in a database into  $\langle N \rangle$  sublists and runs a copy of dostats, using the dostats **-i@listfile** option, for each sublist as a background task. The tables are sorted by size (descending by default, ascending optional) and assigned to the  $\langle N \rangle$  sublists round-robin so that the largest (smallest) tables, arguably the most significant, are processed first. Essentially  $\langle N \rangle$  tables at a time are processed in parallel.

# Using `drive_dostats` to make `dostats` runs faster

If you have enough horsepower (spare CPU cycles, memory and storage bandwidth for sorting, and storage bandwidth to read index and data pages) to process all of that data in parallel `drive-dostats` can run your stats many times faster than a single `dostats` session can.

# Using drive\_dostats to make dostats runs faster

```
$ drive_dostats
```

Usage:

```
drive_dostats nprocs database [tablespec] [-x@excl|-xexctbl] [dostats options]  
[-i@incl] [-iinctbl]
```

**tablespec** - a MATCHES style wildcard to select tables to include

**-a** - Reverse sort ordering of tables to smallest first

**-x@excl** - excl is a file containing tablenamees to ignore

**-xexctbl** - exctbl is a tablename to ignore

Multiple -x and -x@ options are accepted and can be mixed

**-i@incl** - incl is a file containing tablenamees to process

**-iinctbl** - inctbl is a tablename to process

Multiple -i and -i@ options are accepted and can be mixed

dostats options - Most dostats options are passed on and are valid.

Some (notably -i) have no meaning and will fail.

Mixing -x/-x@ and -i/-i@ options only tables which appear in the include list but do not appear in the exclude list will be processed. Mixing these options should be carefully considered and planned to avoid unexpected results.

# Using drive\_dostats to make dostats runs faster

On one of my databases on my 2 core machine:

dostats single run: 2.577 sec

drive\_dostats 4 run: 1.982 sec

That's a 30% improvement!

Another database:

dostats single run: 5.588 sec

drive\_dostats 10 run: 0.713 sec

That's a 683% improvement!

# Questions ?!?



# Review of UPDATE STATISTICS

- The Performance Guide suggests a protocol sequence of UPDATE STATISTICS commands designed to get a useful level of distributions while minimizing the time required to produce them.
- UPDATE STATISTICS HIGH with more than the default resolution will always be at least as good as this protocol and sometimes better but it will take much more time to execute!

# Review of UPDATE STATISTICS

The Performance Guide specifies that the following commands be executed for each table:

→UPDATE STATISTICS HIGH ... DISTRIBUTIONS ONLY for the first column of each index key.

→UPDATE STATISTICS HIGH ... DISTRIBUTIONS ONLY for the first column that is different if multiple index keys begin with the same column(s).

→UPDATE STATISTICS MEDIUM on all other columns.

→UPDATE STATISTICS LOW on the full key for each index.

I would add HIGH on frequently used filter columns and columns used in partitioning expressions that are not indexed.

# Review of UPDATE STATISTICS

- Informix Dynamic Server v11.70 introduced conditional update statistics processing.
  - New distributions are calculated only if certain thresholds were reached since the current distributions were calculated:
    - The number of rows changes by the threshold percentage
    - The number of inserts, deletes, and/or updates indicates that more than the threshold percentage of rows may have been modified.

# Review of UPDATE STATISTICS

- Informix Dynamic Server v11.70 introduced conditional update statistics processing.
  - Threshold is an attribute of each table (STATCHANGE)
    - Default threshold is in the ONCONFIG file (STATCHANGE = 10)
    - Feature enabled/disabled with AUTO\_STAT\_MODE 1/0

# Review of UPDATE STATISTICS

- As of 11.10 and later, CREATE INDEX produces data distributions (HIGH) on the lead column only of B+Tree indexes and updates the sysindexes statistical column values at creation time.
- This is better than nothing and if the remaining key columns already have current distributions then this is fine.
- Not good enough if the table was just created!

# Review of Auto Update Stats (AUS)

- Implemented starting in 11.10
- On by default
- Two tasks
  - AUS Evaluator
  - AUS Refresh

# Review of Auto Update Stats (AUS)

- The Evaluator was originally a very high overhead task. Fixed in 11.70+
- Evaluator and Refresh tasks scheduled one hour apart by default. Which is often not enough time for the evaluator to complete if you have many tables and/or many databases.
- If the evaluator does not complete before the refresh kicks off the refresh may not run or may only process a subset of your tables.

# Review of Auto Update Stats (AUS)

- By default:
  - AUS generates HIGH distributions for leading columns of indexes
  - AUS generates MEDIUM distributions for all other columns included in any index key
  - AUS does not generate any data distributions for any columns that are not included in an index key!
- If distributions already exist for a column the AUS evaluator will duplicate that level of distributions instead of the defaults – need dostats to prime the pump

# New Update Stats Features

New index scan algorithm is 3x faster for LOW stats.  
How?

Remove the RA\_ parameters from the ONCONFIG file and set AUTO\_READAHEAD to 1.

New index sampling for LOW stats is 20x faster. How?

Set USTLOW\_SAMPLE to 1 in your ONCONFIG file or run:  
Onmode -wm USTLOW\_SAMPLE 1

# Questions?



Send follow-up questions to  
[info@advanceddatatools.com](mailto:info@advanceddatatools.com)

# Next Webcast

- More Informix Webcasts coming soon...
- Please check our Website for Future webcasts here at:  
<http://advanceddatatools.com/Informix/NextWebcast.html>

# Informix Training in 2019

Attend classes online on the web, or in person at our training center in Virginia. All you need is a web browser to connect to our WebEx training system and an SSH client (like Putty) to connect to our training lab for hands-on exercises. Each student uses an 8-core Linux server, with 16GB RAM, SSD drives with Informix 12, and several large databases for benchmark exercises.

- **March 11-14, 2019 - Advanced Informix Performance Tuning**
  - This course is for database administrators and application developers who will be responsible for managing, optimizing, and tuning an Informix database server. The focus is on skills, procedures, and scripts to improve the performance of your database server. The course will provide a toolkit of scripts and utilities to start monitoring and optimizing your Informix database server. Our advanced course is taught together by Lester Knutsen and Art Kagel, two of the most experienced Informix DBA consultants in the world.
  
- **April 22-25, 2019 - Informix for Database Administrators**
  - This course is for new database administrators, programmers, and technical support personnel who will be setting up, managing, and tuning IBM Informix databases.
  
- **September 16-19, 2019 - Informix for Database Administrators**
  - This course is for new database administrators, programmers, and technical support personnel who will be setting up, managing, and tuning IBM Informix databases.

➤ **More Information and Registration at:**

<http://www.advanceddatatools.com/Training/InformixTraining.html>

***Advanced DataTools***

# Eight New Training Servers



© Lester Knutsen 2018

Each Student in class will have a server running Informix 12.10 with:

- 8 CPU Cores
- 16 GB RAM
- 1 SSD Disk
- 1- 4 Disks



## ***Informix Support and Training from the Informix Champions!***

Advanced DataTools is an Advanced Level IBM Informix Data Management Partner, and has been an authorized Informix partner since 1993. We have a long-term relationship with IBM, we have priority access to high-level support staff, technical information, and Beta programs. Our team has been working with Informix since its inception, and includes 8 Senior Informix Database Consultants, 4 IBM Champions, 2 IIUG Director's Award winners, and an IBM Gold Consultant. We have Informix specialists Lester Knutsen and Art Kagel available to support your Informix performance tuning and monitoring requirements!

- ***Informix Remote DBA Support Monitoring***
- ***Informix Performance Tuning***
- ***Informix Training***
- ***Informix Consulting***
- ***Informix Development***

**Free Informix Performance Tuning Webcast replays at:**

<http://advancedatools.com/Informix/Webcasts.html>

Email: [info@advancedatools.com](mailto:info@advancedatools.com)

Web: <http://www.advancedatools.com>



***Advanced DataTools***

**Thank You**

***Advanced DataTools Corporation***

For more information:

**info@advancedatools.com**

**http://www.advancedatools.com**

***Advanced DataTools***

Art S. Kagel  
ASK Database  
Management

[art@askdbmgt.com](mailto:art@askdbmgt.com)

[www.askdbmgt.com](http://www.askdbmgt.com)