**Sequential DRAT (Distributed Release Audit Tool)**

**What**

This is a simple utility which uses DRAT to scan multiple code repositories sequentially and dump its output to user defined directory.

**Why**

As we know that DRAT runs on single code repository and generates the output. But what if we have large number of repositories to be scanned and record their individual statistics. This utility can be leveraged to such large-scale tasks.

**How to use**

Presently, script ‘dratseq’ has two actions defined in it – run and help.

1. **run** – This command takes two arguments:
   1. Path to a file having list of repositories to traverse. Each line in the file represents one repository **absolute** path. Eg: Below file has path for Apache tika and nutch.

/apacheSvn/tika  
/apacheSvn/nutch

* 1. Path to the output directory where the contents of ${DRAT\_HOME}/data will be copied to, for each repository. Each folder in the output directory follow standard naming conventions i.e.
     1. Remove the first character which will always start with ‘/’
     2. All ‘/’ will be replaced with ‘\_’
     3. And it will be appended with current timestamp in the format YYYYmmddHHMMSS (YearMonthDateHourMinuteSeconds)

Example - An output directory of ‘/apacheSvn/tika’ repository can be written as apacheSvn\_tika\_20150919090532

1. **help** – This command takes no arguments and returns usage of the command ‘dratseq’. A sample output of the same is as below:

Usage: dratseq [run] in order to analyze multiple repositories.

run <path to list of repository URLs> <path to data directory> | start OODT and analyze the repositories

help | print this message

**How it Works**

The script follows below steps in sequence:

1. Check for correct number of arguments passed to **run** command. If false, exit.
2. Check if the first argument is file and second is a directory. If false, exit.
3. Check if environment variables (JAVA\_HOME and DRAT\_HOME) are set. If false, exit.
4. While read each repository from the list:
   1. Check if the repository path exists. If false, continue to next repository.
   2. Start OODT. If failed to start, exit.
   3. Run DRAT on the repository.
   4. Check if DRAT was successful. Prints error message.
   5. Stop OODT. If failed to stop, exit.
   6. Creates a sub-directory in the output directory to records DRAT statistics.
   7. Copy ${DRAT\_HOME}/data to the user-defined output directory.
   8. Reset DRAT.
5. Endwhile.