NAME

`mrls` – list attributes of Oracle extended SQL trace files

SYNOPSIS

`mrls` [`--all` | `-a`] [`--csv`] [`--dashes`] [`--debug`] [`--exclude=`*list* | `-e` *list*] [`--format=`*string*] [`--head`] [`--help` | `-?`] [`--include=`*list* | `-i` *list*] [`--license`] [`--man`] [`--recursive` | `-R`] [`--reverse` | `-r`] [`--sort=`*colname*] [`--verbose` | `--verbose=`*level*] [`--version`] [*file* …]

DESCRIPTION

`mrls` prints an `ls(1)` style report about Oracle extended SQL trace files. If one or more *file* names is listed, `mrls` reports on only those files. If no *file* argument is given, `mrls` investigates each file in the current working directory. To report only on files with a `.trc` file name extension, use `mrls *.trc`.

OPTIONS

`--all`, `-a` Include files whose names begin with a dot. It is customary for such files to be considered hidden. The default is `--noall`.

`--csv` Use comma-separated value output format. Using `--csv` is shorthand for `--format=csv --nodashes`. If you wish to use different values for these options, then specify those options separately on the command line after the `--csv` option. For example, `--csv --dashes` will do the expected thing, but using `--dashes --csv` will print no dashes, because it is equivalent to specifying `--dashes --format=csv --nodashes`.

`--dashes` Print dashes between the header and the body. The default is `--nodashes`.

`--debug` Print debugging information to `stderr`. The default is `--nodebug`.

`--exclude=`*list*, `-e` *list* See `--include=`*list*.

`--format=`*string* Print in the specified format. Supported values are `tab` (table) and `csv` (comma-separated values). The default is `--format=tab`.

`--head` Print a heading at the beginning of the report. Use `--nohead` to suppress the heading. The default is `--head`.

`--help`, `-?` Print usage information and exit.

`--include=`*list*, `-i` *list*; `--exclude=`*list*, `-e` *list* Include or exclude the named columns in the output. Use commas or separate directives to include or exclude two or more columns, as in:

```
--include=LIO,PIO
--include=LIO --include=PIO
```

Contradictions between `--include` and `--exclude` are resolved with later option declarations overriding earlier ones. Thus,

```
--include=DEP --exclude=DEP   # excludes DEP
--exclude=DEP --include=DEP   # includes DEP
```

Columns you can include, exclude, or sort by (see `--sort`) are:

`:all` Shorthand for the list of all permissible column names. Use `--include=:all` to select all columns, which you can selectively pare away with `--exclude`. Or use `--exclude=:all` to empty the output column list so that you can add columns with `--include`.

`CPU` The number of CPU seconds consumed by the experience, calculated as the sum of `c` values for database calls executed at the minimum recursive depth reported within the file.

`DEP` The recursive depth value of the task's topmost database call in the call stack, calculated as the minimum `dep` value found in the file.

`DURATION` The duration represented by the file. You cannot exclude `DURATION` from the output.

`END` The end time of the experience represented by the file.

`END-TIM` The `tim` value corresponding to the `END` value.

`FILE` The name of the file. You cannot exclude `FILE` from the output.

`LIO` The number of database buffer cache accesses, calculated as the sum of `cr` and `cu` values for database calls executed at recursive depth 0.

`MIS` The number of database library cache misses, calculated as the sum of `mis` values for database calls executed at recursive depth 0.

`PIO` The number of database blocks obtained by OS read calls, calculated as the sum of `p` values for database calls executed at recursive depth 0.

`ROW` The number of rows returned by dbcalls, calculated as the sum of `r` values for database calls executed at recursive depth 0.

`SIZE` The size of the file in bytes.

`START` The begin time of the experience represented by the file.

`START-TIM` The `tim` value corresponding to the `START` value.

`STATUS` The status of the `mrls` call upon its file. Empty means no errors or warnings. The status value can contain the following flags:

Flag	Meaning
`e`	`DEP`, `START`, and `DURATION` are estimated
`t`	`START` not found

`TAGS` The tag values associated with the file.

`VERSION` The version of the Oracle Database that wrote the trace file.

The default is `--include=DURATION,SIZE,VERSION,START,FILE`.

`--license` Print license key information and exit.

`--man` Print the manual page and exit.

`--recursive, -R` Plunge directories recursively. The default is `--norecursive`. If you use the short `-R` form, you must specify the uppercase letter ‘R’ (to disambiguate from the `-r` “reverse” option).

`--reverse, -r` Sort in reverse order. The default is `--noreverse`.

`--sort=`*colname* Sort by the named column. Permissible column names are listed in the `--include` option entry. Using `--sort=`*colname* automatically implies `--include=`*colname*. Excluding a column after including it in a `--sort` argument (e.g., `--sort=CPU --exclude=CPU`) is prohibited. The default is `--sort=DURATION`. The secondary sort field is always `FILE`, in ascending order.

`--tags-dbfile=`*file* Use *file* (a SQLite database created by Method R Workbench) for retrieval of `TAGS` information. The default is `--tags-file=$HOME/.method-r/workbench/9.2.1.2/files.db`.

`--verbose, --verbose=`*level* Print supplemental information to `stdout`. The default is `--verbose=0`. Using `--verbose` without an argument is equivalent to using `--verbose=1`.

`--version` Print the version number and exit.

EXAMPLES

```
mrls                  # open all files in cwd, reporting only on trace files
mrls *.trc            # open only .trc files in cwd
mrls --include=CPU    # add the CPU column to the output (takes longer)
mrls --include=:all   # add all columns to the output (also takes longer)
mrls --sort=START     # sort oldest-start-time first
mrls --sort=ROW -r    # sort most-rows-manipulated first
```

TROUBLESHOOTING

When you report on the default mrls columns, mrls can often find the information it needs from just the first and last few lines of its input file. This feature makes mrls very fast, even for large files. In cases where mrls can't find everything it needs, it will throw a warning to stderr and set a flag in the STATUS column if you have included STATUS in your output.

The following columns force mrls to read the entire input file, which takes considerably longer for large files: CPU, DEP, LIO, MIS, PIO, and ROW. If mrls throws a warning that it is estimating or unable to find a value, you can include any of these columns, and mrls will print precise, non-estimated output. Or you can load the file into the Method R Workbench application.

If a file's DURATION value is larger than you expected, the file may contain trace data for more than one user experience. The total response time that mrls reports is the duration from the earliest *tim* value it finds near the top of the file to the latest *tim* value it finds near the bottom. If your trace file contains data for more than one user experience (for example, because of connection pooling, or because an Oracle process has reused an operating system process ID), then use mrcrop or the Oracle trcsess utility to chop the trace file into smaller chunks, each representing a single end-user experience.

EXIT STATUS

Exit status is 0 on successful completion, and > 0 if an error occurs.

BUGS AND DEFICIENCIES

mrls does not open zip archives. One could argue that it should treat a zip archive like it treats a directory, plunging and reporting on what's inside. However, to gain a significant speed advantage for large trace files (presumably the very ones you'd be most likely to zip), mrls reads a little bit from the top of the file and a little bit from the bottom. To do this would require mrls to inflate and temporarily store each archive. This would cause such a response time problem for mrls that we don't presently believe it would be worth the time to try to implement the feature.

mrls does not return *the* Oracle Database version for a given Oracle trace file; it returns the *first* Oracle Database version it finds the file. The two answers are different in situations wherein a single trace file contains segments of trace data collected from more than one version of Oracle, such as will happen when Oracle reuses OS process ID values.

AUTHORS

Cary Millsap, Jeff Holt

SUPPORT

`mrls` 9.2.1.2

For support, visit https://method-r.com/support.

COPYRIGHT AND LICENSE