# `Benchmark_O` Tests

The `Benchmark_O` binary is used directly from command line as well as a subcomponent invoked from higher-level scripts (eg. `Benchmark_Driver`). These script therefore depend on the supported command line options and the format of its console output. The following `lit` tests also serve as a verification of this public API to prevent its accidental breakage.

Note: Following tests use *Existential.* as an example of a benchmarks that are excluded from the default "pre-commit" list because they are marked `skip` and the default skip-tags ( `unstable,skip` ) will exclude them. The *Ackermann* and *AngryPhonebook* are alphabetically the first two benchmarks in the test suite (used to verify running by index). If these assumptions change, the test must be adapted.

## List Format

```
RUN: %Benchmark_O --list | %FileCheck %s \
RUN:                      --check-prefix LISTPRECOMMIT \
RUN:                      --check-prefix LISTTAGS
LISTPRECOMMIT: #,Test,[Tags]
LISTPRECOMMIT-NOT: Existential.
LISTPRECOMMIT: {{[0-9]+}},AngryPhonebook
LISTTAGS-SAME: ,[
LISTTAGS-NOT: TestsUtils.BenchmarkCategory.
LISTTAGS-SAME: String, api, validation
LISTTAGS-SAME: ]
```

Verify `Existential.` benchmarks are listed when skip-tags are explicitly empty and that they are marked `skip` :

```
RUN: %Benchmark_O --list --skip-tags= | %FileCheck %s --check-prefix LISTALL
LISTALL: AngryPhonebook
LISTALL: Existential.
LISTALL-SAME: skip
```

## Benchmark Selection

The logic for filtering tests based on specified names, indices and tags is shared between the default "run" and `--list` commands. It is tested on the list command, which is much faster, because it runs no benchmarks. It provides us with ability to do a "dry run".

Run benchmark by name (even if its tags match the skip-tags) or test number:

```
RUN: %Benchmark_O Existential.Mutating.Ref1 --list \
RUN:              | %FileCheck %s --check-prefix NAMEDSKIP
NAMEDSKIP: Existential.Mutating.Ref1

RUN: %Benchmark_O 1 --list | %FileCheck %s --check-prefix RUNBYNUMBER
RUNBYNUMBER: Ackermann
```

Composition of `tags` and `skip-tags` :

```
RUN: %Benchmark_O --list --tags=Dictionary,Array \
RUN:             | %FileCheck %s --check-prefix ANDTAGS
ANDTAGS: TwoSum
ANDTAGS-NOT: Array2D
ANDTAGS-NOT: DictionarySwap

RUN: %Benchmark_O --list --tags=algorithm --skip-tags=validation \
RUN:             | %FileCheck %s --check-prefix TAGSANDSKIPTAGS
TAGSANDSKIPTAGS: Ackermann
TAGSANDSKIPTAGS: DictOfArraysToArrayOfDicts
TAGSANDSKIPTAGS: Fibonacci
TAGSANDSKIPTAGS: RomanNumbers

RUN: %Benchmark_O --list --tags=algorithm \
RUN:              --skip-tags=validation,Dictionary,String \
RUN:             | %FileCheck %s --check-prefix ORSKIPTAGS
ORSKIPTAGS: Ackermann
ORSKIPTAGS-NOT: DictOfArraysToArrayOfDicts
ORSKIPTAGS: Fibonacci
ORSKIPTAGS-NOT: RomanNumbers
```

Alphabetic sorting of tests

```
RUN: %Benchmark_O --list \
RUN:             | %FileCheck %s --check-prefix ALPHASORT
ALPHASORT: COWArrayGuaranteedParameterOverhead
ALPHASORT: COWTree
ALPHASORT: ChainedFilterMap
ALPHASORT: Chars
ALPHASORT: FatCompactMap
```

Substring filters using + and - prefix

```
RUN: %Benchmark_O --list -.A +Angry -Small AngryPhonebook.ASCII2.Small \
RUN:             | %FileCheck %s --check-prefix FILTERS
FILTERS: AngryPhonebook.ASCII2.Small
FILTERS-NOT: AngryPhonebook.Armenian
FILTERS-NOT: AngryPhonebook.Cyrillic.Small
FILTERS: AngryPhonebook.Cyrillic
FILTERS: AngryPhonebook.Strasse
```

## Running Benchmarks

By default, each real benchmark execution takes about a second per sample. To minimise the test time, multiple checks are combined into one run.

```
RUN: %Benchmark_O AngryPhonebook --num-iters=1 \
RUN:                             --sample-time=0.000001 --min-samples=7 \
RUN:              | %FileCheck %s --check-prefix NUMITERS1 \
RUN:                              --check-prefix LOGHEADER \
RUN:                              --check-prefix LOGBENCH
```

```
LOGHEADER-LABEL: #,TEST,SAMPLES,MIN(μs),MAX(μs),MEAN(μs),SD(μs),MEDIAN(μs)
LOGBENCH: {{[0-9]+}},
NUMITERS1: AngryPhonebook,7
NUMITERS1-NOT: 0,0,0,0,0
LOGBENCH-SAME: ,{{[0-9]+}},{{[0-9]+}},{{[0-9]+}},{{[0-9]+}},{{[0-9]+}}
```

### Reporting Quantiles

The default benchmark result reports statistics of a normal distribution — mean and standard deviation. Unfortunately the samples from our benchmarks are *not normally distributed*. To get a better picture of the underlying probability distribution, we support reporting quantiles.

```
RUN: %Benchmark_O 0 --quantile=4 | %FileCheck %s --check-prefix FIVENUMSUMMARY
FIVENUMSUMMARY: #,TEST,SAMPLES,MIN(μs),Q1(μs),Q2(μs),Q3(μs),MAX(μs)
RUN: %Benchmark_O 0 --quantile=20 | %FileCheck %s --check-prefix VENTILES
VENTILES: #,TEST,SAMPLES,MIN(μs),V1(μs),V2(μs),V3(μs),V4(μs),V5(μs),V6(μs),
VENTILES: V7(μs),V8(μs),V9(μs),VA(μs),VB(μs),VC(μs),VD(μs),VE(μs),VF(μs),VG(μs),
VENTILES: VH(μs),VI(μs),VJ(μs),MAX(μs)
```

### Reporting Measurement Metadata

The presence of optional argument `--meta`, controls logging of measurement metadata at the end of the benchmark summary.

- PAGES – number of memory pages used
- ICS – number of involuntary context switches
- YIELD – number of voluntary yields

```
RUN: %Benchmark_O 0 --quantile=1 --meta | %FileCheck %s --check-prefix META
META: #,TEST,SAMPLES,MIN(μs),MAX(μs),PAGES,ICS,YIELD
RUN: %Benchmark_O 0 --quantile=1 --meta --memory \
RUN:              | %FileCheck %s --check-prefix MEMMETA
MEMMETA: #,TEST,SAMPLES,MIN(μs),MAX(μs),MAX_RSS(B),PAGES,ICS,YIELD
```

### Verbose Mode

Reports detailed information during measurement, including configuration details, environmental statistics (memory used and number of context switches) and all individual samples. We'll reuse this test to check arguments that modify the reported columns: `--memory`, `--quantile` and `--delta` to end with *one less* number in the benchmark summary, compared to normal format. Given that we are taking only 2 samples, the MEDIAN and MAX will be the same number. With the `--delta` option this means that **Δ**MAX is zero, so the penultimate number will be omitted from the output, giving us 2 consecutive delimiters („).

```
RUN: %Benchmark_O 1 Ackermann 1 AngryPhonebook \
RUN:              --verbose --num-samples=2 --memory --quantile=2 --delta \
RUN:              | %FileCheck %s --check-prefix RUNJUSTONCE \
RUN:                              --check-prefix CONFIG \
RUN:                              --check-prefix LOGVERBOSE \
RUN:                              --check-prefix MEASUREENV \
RUN:                              --check-prefix LOGFORMAT \
RUN:                              --check-prefix YIELDCOUNT
CONFIG: NumSamples: 2
```

```
CONFIG: Tests Filter: ["1", "Ackermann", "1", "AngryPhonebook"]
CONFIG: Tests to run: Ackermann, AngryPhonebook
LOGFORMAT: #,TEST,SAMPLES,MIN(μs),𝚫MEDIAN,𝚫MAX,MAX_RSS(B)
LOGVERBOSE-LABEL: Running Ackermann
LOGVERBOSE: Collecting 2 samples.
LOGVERBOSE: Measuring with scale {{[0-9]+}}.
LOGVERBOSE: Sample 0,{{[0-9]+}}
LOGVERBOSE: Sample 1,{{[0-9]+}}
MEASUREENV: MAX_RSS {{[0-9]+}} - {{[0-9]+}} = {{[0-9]+}} ({{[0-9]+}} pages)
MEASUREENV: ICS {{[0-9]+}} - {{[0-9]+}} = {{[0-9]+}}
MEASUREENV: VCS {{[0-9]+}} - {{[0-9]+}} = {{[0-9]+}}
YIELDCOUNT: yieldCount 1
RUNJUSTONCE-LABEL: 1,Ackermann
RUNJUSTONCE-NOT: 1,Ackermann
LOGFORMAT: ,{{[0-9]+}},{{[0-9]+}},,{{[0-9]*}},{{[0-9]+}}
LOGVERBOSE-LABEL: Running AngryPhonebook
LOGVERBOSE: Collecting 2 samples.
```

Verify the specified delimiter is used when logging to console. The non-verbose variant of this invocation is used from `Benchmark_Driver` to get the list of all tests. That's why it is *crucial* to tests this integration point.

```
RUN: %Benchmark_O --list --skip-tags= --delim=$'\t' --verbose \
RUN:              | %FileCheck %s --check-prefix LOGVERBOSEDELIM
LOGVERBOSEDELIM: Delimiter: "\t"
LOGVERBOSEDELIM: #      Test    [Tags]
```

## Error Handling

```
RUN: not %Benchmark_O --bogus 2>&1 \
RUN:              | %FileCheck %s --check-prefix ARGPARSE
ARGPARSE: error: unsupported argument '--bogus'

RUN: not %Benchmark_O --sample-time \
RUN:         2>&1 | %FileCheck %s --check-prefix NOVALUE
NOVALUE: error: missing value for '--sample-time'

RUN: not %Benchmark_O --sample-time= \
RUN:         2>&1 | %FileCheck %s --check-prefix EMPTYVAL
EMPTYVAL: error: missing value for '--sample-time'

RUN: not %Benchmark_O --sample-time=NaN \
RUN:         2>&1 | %FileCheck %s --check-prefix NANVALUE
NANVALUE: error: 'NaN' is not a valid 'Double' for '--sample-time'

RUN: not %Benchmark_O --num-iters \
RUN:         2>&1 | %FileCheck %s --check-prefix NUMITERS
NUMITERS: error: missing value for '--num-iters'

RUN: not %Benchmark_O --num-samples \
RUN:         2>&1 | %FileCheck %s --check-prefix NUMSAMPLES
NUMSAMPLES: error: missing value for '--num-samples'
```

```
RUN: not %Benchmark_O --sleep \
RUN:         2>&1 | %FileCheck %s --check-prefix SLEEP
SLEEP: error: missing value for '--sleep'

RUN: not %Benchmark_O --delim \
RUN:         2>&1 | %FileCheck %s --check-prefix DELIM
DELIM: error: missing value for '--delim'

RUN: not %Benchmark_O --tags=bogus \
RUN:         2>&1 | %FileCheck %s --check-prefix BADTAG
BADTAG: error: 'bogus' is not a valid 'BenchmarkCategory'

RUN: not %Benchmark_O --skip-tags=bogus \
RUN:         2>&1 | %FileCheck %s --check-prefix BADSKIPTAG
BADSKIPTAG: error: 'bogus' is not a valid 'BenchmarkCategory'
```

Measuring memory use of a test with our method is valid only for single test.

```
RUN: %Benchmark_O 1 2 --memory --list \
RUN:         2>&1 | %FileCheck %s --check-prefix WARNMEMORY
WARNMEMORY: warning:
```

## Usage

```
RUN: %Benchmark_O --help | %FileCheck %s --check-prefix OPTIONS
OPTIONS: usage: Benchmark_O [--argument=VALUE] [TEST [TEST ...]]
OPTIONS: optional arguments:
OPTIONS: --help
OPTIONS-SAME: show this help message and exit
OPTIONS: --verbose
OPTIONS: --delim
OPTIONS: --tags
OPTIONS: --list
```