

Google benchmark library relies
on static variables & static
initializers.

```
—// Helpers for generating unique variable names
#define BENCHMARK_PRIVATE_NAME(...) \
BENCHMARK_PRIVATE_CONCAT(benchmark_uniq_, BENCHMARK_PRIVATE_UNIQUE_ID, \
                         __VA_ARGS__)

#define BENCHMARK_PRIVATE_CONCAT(a, b, c) BENCHMARK_PRIVATE_CONCAT2(a, b, c)
#define BENCHMARK_PRIVATE_CONCAT2(a, b, c) a##b##c
// Helper for concatenation with macro name expansion
#define BENCHMARK_PRIVATE_CONCAT_NAME(BaseClass, Method) \
BaseClass##_##Method##_Benchmark

#define BENCHMARK_PRIVATE_DECLARE(n) \
/* NOLINTNEXTLINE(misc-use-anonymous-namespace) */ \
static ::benchmark::internal::Benchmark const* const BENCHMARK_PRIVATE_NAME( \
    n) BENCHMARK_UNUSED

#define BENCHMARK(...) \
BENCHMARK_PRIVATE_DECLARE(_benchmark_) = \
    (::benchmark::internal::RegisterBenchmarkInternal( \
        ::benchmark::internal::make_unique< \
            ::benchmark::internal::FunctionBenchmark>(#__VA_ARGS__, \
                __VA_ARGS__)))
```

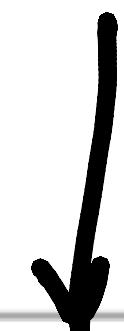
As explained in the following
note : , the

BENCHMARK(BM_EMPTY) creates a

static variable which results
in registration of Benchmark
(via RegisterBenchmarkInternal) at
runtime.

C++ guarantees order of static
variable initialization in a
single translation unit (.cc file),
but the order or even execution
nature (serial or parallel) is
not guaranteed across translation
units.

Imagine multiple static variables
across 2 translation units,
causing benchmark registration.



Will result in adding benchmarks
to BenchmarkFamilies.

↓ Hence, AddBenchmarks

of BenchmarkFamilies needs mutex.

For the same reason,

FindBenchmarks of BenchmarkFamilies
needs a mutex.

Consider 2 benchmarks with
executables A & B.

Both are executed at the
same time, A is still registering
benchmarks, while B is done
registering benchmarks. B starts
its execution :

which involves 'finding benchmarks'
Finding benchmarks also uses
BenchmarkFamilies which is
a shared object, & is still being

used by A. Hence, even Find -

- Benchmarks needs a mutex.

