
unit

eliasku

Jun 03, 2022

CONTENTS:

1	API	1
2	Indices and tables	5
	Index	7

Defines

UNIT__NOOP

unit.h - v0.0.6 - Simple header-only testing library for C - <https://github.com/eliasku/unit>

Minimal example. Compile executable with `-D UNIT_TESTING` to enable tests.

```
#define UNIT_MAIN
#include "unit.h"

suite( you_should_define_the_suite_name_here ) {
    describe( optionally_add_the_subject ) {
        it( "describe the test behaviour" ) {
            check("use warn / check / require functions");
        }
    }
}
```

By default all test code is stripped away, to enable test-code you should pass `-D UNIT_TESTING` and build executable

In any translation unit you need to `#define UNIT_MAIN` before `#include <unit.h>` to implement the library and generate default main entry-point to run all tests.

If you need just to implement library, you `#define UNIT_IMPLEMENT` before `include <unit.h>` in any single translation unit Declare default main arguments

`UNIT__CONCAT_(a, b)`

`UNIT__CONCAT(a, b)`

`UNIT_SUITE(Name, ...)`

`UNIT_DESCRIBE(Name, ...)`

`UNIT_TEST(Description, ...)`

`UNIT_ECHO(...)`

`UNIT_WARN(x, ...)`

`UNIT_WARN_FALSE(x, ...)`

UNIT_WARN_EQ(a, b, ...)
UNIT_WARN_NE(a, b, ...)
UNIT_WARN_GT(a, b, ...)
UNIT_WARN_GE(a, b, ...)
UNIT_WARN_LT(a, b, ...)
UNIT_WARN_LE(a, b, ...)
UNIT_CHECK(x, ...)
UNIT_CHECK_FALSE(x, ...)
UNIT_CHECK_EQ(a, b, ...)
UNIT_CHECK_NE(a, b, ...)
UNIT_CHECK_GT(a, b, ...)
UNIT_CHECK_GE(a, b, ...)
UNIT_CHECK_LT(a, b, ...)
UNIT_CHECK_LE(a, b, ...)
UNIT_REQUIRE(x, ...)
UNIT_REQUIRE_FALSE(x, ...)
UNIT_REQUIRE_EQ(a, b, ...)
UNIT_REQUIRE_NE(a, b, ...)
UNIT_REQUIRE_GT(a, b, ...)
UNIT_REQUIRE_GE(a, b, ...)
UNIT_REQUIRE_LT(a, b, ...)
UNIT_REQUIRE_LE(a, b, ...)
UNIT_SKIP(...)
unit_main(...)
SUITE(...)
DESCRIBE(...)
IT(...)
TEST(...)
ECHO(...)
WARN(...)
WARN_FALSE(...)

WARN_EQ(...)

WARN_NE(...)

WARN_GT(...)

WARN_GE(...)

WARN_LT(...)

WARN_LE(...)

CHECK(...)

CHECK_FALSE(...)

CHECK_EQ(...)

CHECK_NE(...)

CHECK_GT(...)

CHECK_GE(...)

CHECK_LT(...)

CHECK_LE(...)

REQUIRE(...)

REQUIRE_FALSE(...)

REQUIRE_EQ(...)

REQUIRE_NE(...)

REQUIRE_GT(...)

REQUIRE_GE(...)

REQUIRE_LT(...)

REQUIRE_LE(...)

SKIP(...)

INDICES AND TABLES

- genindex
- modindex
- search

INDEX

C

CHECK (*C macro*), 3
CHECK_EQ (*C macro*), 3
CHECK_FALSE (*C macro*), 3
CHECK_GE (*C macro*), 3
CHECK_GT (*C macro*), 3
CHECK_LE (*C macro*), 3
CHECK_LT (*C macro*), 3
CHECK_NE (*C macro*), 3

D

DESCRIBE (*C macro*), 2

E

ECHO (*C macro*), 2

I

IT (*C macro*), 2

R

REQUIRE (*C macro*), 3
REQUIRE_EQ (*C macro*), 3
REQUIRE_FALSE (*C macro*), 3
REQUIRE_GE (*C macro*), 3
REQUIRE_GT (*C macro*), 3
REQUIRE_LE (*C macro*), 3
REQUIRE_LT (*C macro*), 3
REQUIRE_NE (*C macro*), 3

S

SKIP (*C macro*), 3
SUITE (*C macro*), 2

T

TEST (*C macro*), 2

U

UNIT__CONCAT (*C macro*), 1
UNIT__CONCAT_ (*C macro*), 1
UNIT__NOOP (*C macro*), 1
UNIT_CHECK (*C macro*), 2

UNIT_CHECK_EQ (*C macro*), 2
UNIT_CHECK_FALSE (*C macro*), 2
UNIT_CHECK_GE (*C macro*), 2
UNIT_CHECK_GT (*C macro*), 2
UNIT_CHECK_LE (*C macro*), 2
UNIT_CHECK_LT (*C macro*), 2
UNIT_CHECK_NE (*C macro*), 2
UNIT_DESCRIBE (*C macro*), 1
UNIT_ECHO (*C macro*), 1
unit_main (*C macro*), 2
UNIT_REQUIRE (*C macro*), 2
UNIT_REQUIRE_EQ (*C macro*), 2
UNIT_REQUIRE_FALSE (*C macro*), 2
UNIT_REQUIRE_GE (*C macro*), 2
UNIT_REQUIRE_GT (*C macro*), 2
UNIT_REQUIRE_LE (*C macro*), 2
UNIT_REQUIRE_LT (*C macro*), 2
UNIT_REQUIRE_NE (*C macro*), 2
UNIT_SKIP (*C macro*), 2
UNIT_SUITE (*C macro*), 1
UNIT_TEST (*C macro*), 1
UNIT_WARN (*C macro*), 1
UNIT_WARN_EQ (*C macro*), 1
UNIT_WARN_FALSE (*C macro*), 1
UNIT_WARN_GE (*C macro*), 2
UNIT_WARN_GT (*C macro*), 2
UNIT_WARN_LE (*C macro*), 2
UNIT_WARN_LT (*C macro*), 2
UNIT_WARN_NE (*C macro*), 2

W

WARN (*C macro*), 2
WARN_EQ (*C macro*), 2
WARN_FALSE (*C macro*), 2
WARN_GE (*C macro*), 3
WARN_GT (*C macro*), 3
WARN_LE (*C macro*), 3
WARN_LT (*C macro*), 3
WARN_NE (*C macro*), 3