

Maestra 2.0

A free lightweight unit test environment

Release notes

Documentation revision 2.00



MacroExpressions
<http://www.macroexpressions.com>

Revision history

Revision 2.0 immediately follows the original 1.0

Added:

1. C++ support
2. (Recursive) MC/DC and condition coverage support
3. More robust instrumentation (w.r.t. commas) for compilers supporting variadic macros

Improved

1. Instrumentation for the `for` keyword
2. Much better instrumentation output (outputs what's written, i.e. before the instrumentation macros expansion)

What is Maestra?

Maestra is a reference implementation of the unit test environment described in the paper "C code unit testing on a shoestring" by Ark Khasin. You should be able to find it in the same distribution.

It is free for use and modification; however it comes without any warranty whatsoever.

The name stands for MacroExpressions Test Run Application; it is also an Italian term for a (female) elementary school teacher. That seems quite appropriate for a testing environment.

What is included?

The following files are included:

- Core Maestra files:
 - `maestra.c` – the common test harness implementation
 - `maestra.h` – a magic plug-in for a project-wide header (maps to `instrum.h` in the paper)
 - `maestra_instrum.h` – a reference instrumentation (maps to `instrum_uut.h` in the paper)
 - `maestra_uninstrum.h` – cleanup of instrumentation
 - `maestra_implem.h` – prototypes for reference instrumentation (maps to `instrum_implem.h` in the paper)
 - `maestra_instrum.c` – an implementation of functions used by common instrumentation in `maestra_instrum.h`
 - `maestra_print.c` – an implementation of misc. printing routines useful in the harness
 - `maestra_api.h` – an API useful for writing stubs and specific harnesses
- A simple application example:
 - `uut-sample.c` – a few useless functions demonstrating the instrumentation
 - `project.h` – a bogus project-wide header used in `uut-sample.c`
 - `stub-sample.c` – a sample stub for `uut-sample.c`
 - `harness-sample.c` – a specific harness for `uut-sample.c`
 - `harness-sample2.c` – a specific harness for `uut-sample.c` which employs including the UUT in the harness file
- A result of test execution – `sample.txt`

All files compile under C99.

To make them compile under C90, the following features must be removed:

- Use of `__FUNCTION__` (or define `__FUNCTION__` as "unknown", with quotes)

- Use of designated initializers (that would require better comments in lieu of designators in harness-sample.c)
- Use of the long long and long double types and the corresponding printf formats.
- Comment out the definition of MAESTRA_HAVE_VARIADIC_MACROS in `maestra_instrum.h`

To make the reference implementation useful for unit-testing C++ source files,

- Remove C99-specific features (unless your compiler supports them)
- (Optional) add instrumentation for ``throw'` in a fashion similar to other keywords.
- `maestra_instrum.c` and `maestra_print.c` must be compiled as C sources, the rest should be compiled as C++ (with explicit casts added as necessary).

To make the reference implementation useful for unit-testing in a mixed C/C++ setting, you will probably need two harness files: one for C, another for C++.

What isn't included?

More beefed-up implementation isn't. This version is very basic and doesn't support HTML output and possible improvements, such as specifying max numbers of iterations for loops. Comparisons of strings and of raw memory for the purpose of testing are omitted in this version.

It would also be a good idea to remove the path information (if any) from the filename passed as a parameter to an instrumentation function. This has not been done yet.