

clog: C-style (a-la printf) logging library

License BSD 2-Clause "Simplified" License

C-style library for logging errors, warnings, information notes, and debug information.

Features

- printf-style interface for formatting variadic parameters.
- Separate functions for logging errors, warnings, information notes, and debug information.
- Independent logging settings for different modules.
- Logging to logcat on Android and stderr/stdout on other platforms.
- Compatible with C99 and C++.
- Covered with unit tests.

Example

```
#include <clog.h>

#ifndef MYMODULE_LOG_LEVEL
#define MYMODULE_LOG_LEVEL CLOG_DEBUG
#endif

CLOG_DEFINE_LOG_DEBUG(mymodule_, "My Module", MYMODULE_LOG_LEVEL);
CLOG_DEFINE_LOG_INFO(mymodule_, "My Module", MYMODULE_LOG_LEVEL);
CLOG_DEFINE_LOG_WARNING(mymodule_, "My Module", MYMODULE_LOG_LEVEL);
CLOG_DEFINE_LOG_ERROR(mymodule_, "My Module", MYMODULE_LOG_LEVEL);

...

void some_function(...) {
    int status = ...
    if (status != 0) {
        mymodule_log_error(
            "something really bad happened: "
            "operation failed with status %d", status);
    }

    uint32_t expected_zero = ...
    if (expected_zero != 0) {
        mymodule_log_warning(
            "something suspicious happened (var = %\"PRIu32\")", "
            "fall back to generic implementation", expected_zero);
    }

    void* usually_non_null = ...
    if (usually_non_null == NULL) {
        mymodule_log_info(
            "something unusual, but common, happened: "
```

```
        "enabling work-around");  
    }  
  
    float a = ...  
    mymodule_log_debug("computed a = %.7f", a);  
}
```