CustomLogLevels

How to add or customize log levels

Introduction

By default, Lumberjack comes with 4 pre-defined log levels:

- Error
- Warn
- Info
- Verbose

The small number of levels helps those who are new to logging frameworks. (When there are 16 different log levels, it can be become difficult to choose...)

However, this might not be the proper choice for your project. Or perhaps you're familiar with another logging framework, and you'd like to use the same notations.

You're in luck because **you can customize the log levels however you want**! In fact Lumberjack has structured it's macros to make it easy to customize.

Details

The log levels are all defined in DDLog.h. All you have to do is create your own header (e.g. MYLog.h). Your project files will then import MYLog.h instead of DDLog.h.

Within MYLog.h you simply undefine the pre-defined stuff, and then set everything up however you want.

There is a sample Xcode project that comes with Lumberjack that does exactly this. It looks something like this:

MYLog.h:

```
#import "DDLog.h"

// We want to use the following log levels:
//
// Fatal
// Error
// Warn
// Notice
// Info
// Debug
//
// All we have to do is undefine the default values,
// and then simply define our own however we want.

// First undefine the default stuff we don't want to use.

#undef LOG_FLAG_ERROR
#undef LOG_FLAG_WARN
#undef LOG_FLAG_INFO
#undef LOG_FLAG_VERBOSE

#undef LOG_LEVEL_ERROR
#undef LOG_LEVEL_WARN
#undef LOG_LEVEL_INFO
#undef LOG_LEVEL_VERBOSE

#undef LOG_ERROR
#undef LOG_WARN
#undef LOG_INFO
#undef LOG_VERBOSE

#undef DDLogError(frmt, ...)
#undef DDLogWarn(frmt, ...)
#undef DDLogInfo(frmt, ...)
#undef DDLogVerbose(frmt, ...)

#undef DDLogCError(frmt, ...)
#undef DDLogCWarn(frmt, ...)
```

```
#undef DDLogCInfo(frmt, ...)
#undef DDLogCVerbose(frmt, ...)

// Now define everything how we want it

#define LOG_FLAG_FATAL   (1 << 0)  // 0...000001
#define LOG_FLAG_ERROR   (1 << 1)  // 0...000010
#define LOG_FLAG_WARN    (1 << 2)  // 0...000100
#define LOG_FLAG_NOTICE  (1 << 3)  // 0...001000
#define LOG_FLAG_INFO    (1 << 4)  // 0...010000
#define LOG_FLAG_DEBUG   (1 << 5)  // 0...100000

#define LOG_LEVEL_FATAL   (LOG_FLAG_FATAL)                    // 0...000001
#define LOG_LEVEL_ERROR   (LOG_FLAG_ERROR  | LOG_LEVEL_FATAL ) // 0...000011
#define LOG_LEVEL_WARN    (LOG_FLAG_WARN   | LOG_LEVEL_ERROR ) // 0...000111
#define LOG_LEVEL_NOTICE  (LOG_FLAG_NOTICE | LOG_LEVEL_WARN  ) // 0...001111
#define LOG_LEVEL_INFO    (LOG_FLAG_INFO   | LOG_LEVEL_NOTICE) // 0...011111
#define LOG_LEVEL_DEBUG   (LOG_FLAG_DEBUG  | LOG_LEVEL_INFO  ) // 0...111111

#define LOG_FATAL   (ddLogLevel & LOG_FLAG_FATAL )
#define LOG_ERROR   (ddLogLevel & LOG_FLAG_ERROR )
#define LOG_WARN    (ddLogLevel & LOG_FLAG_WARN  )
#define LOG_NOTICE  (ddLogLevel & LOG_FLAG_NOTICE)
#define LOG_INFO    (ddLogLevel & LOG_FLAG_INFO  )
#define LOG_DEBUG   (ddLogLevel & LOG_FLAG_DEBUG )

#define DDLogFatal(frmt, ...)    SYNC_LOG_OBJC_MAYBE(ddLogLevel, LOG_FLAG_FATAL,  frmt, ##__VA_ARGS__)
#define DDLogError(frmt, ...)    SYNC_LOG_OBJC_MAYBE(ddLogLevel, LOG_FLAG_ERROR,  frmt, ##__VA_ARGS__)
#define DDLogWarn(frmt, ...)    ASYNC_LOG_OBJC_MAYBE(ddLogLevel, LOG_FLAG_WARN,   frmt, ##__VA_ARGS__)
#define DDLogNotice(frmt, ...)  ASYNC_LOG_OBJC_MAYBE(ddLogLevel, LOG_FLAG_NOTICE, frmt, ##__VA_ARGS__)
#define DDLogInfo(frmt, ...)    ASYNC_LOG_OBJC_MAYBE(ddLogLevel, LOG_FLAG_INFO,   frmt, ##__VA_ARGS__)
#define DDLogDebug(frmt, ...)   ASYNC_LOG_OBJC_MAYBE(ddLogLevel, LOG_FLAG_DEBUG,  frmt, ##__VA_ARGS__)

#define DDLogCFatal(frmt, ...)   SYNC_LOG_C_MAYBE(ddLogLevel, LOG_FLAG_FATAL,  frmt, ##__VA_ARGS__)
#define DDLogCError(frmt, ...)   SYNC_LOG_C_MAYBE(ddLogLevel, LOG_FLAG_ERROR,  frmt, ##__VA_ARGS__)
#define DDLogCWarn(frmt, ...)   ASYNC_LOG_C_MAYBE(ddLogLevel, LOG_FLAG_WARN,   frmt, ##__VA_ARGS__)
#define DDLogCNotice(frmt, ...) ASYNC_LOG_C_MAYBE(ddLogLevel, LOG_FLAG_NOTICE, frmt, ##__VA_ARGS__)
#define DDLogCInfo(frmt, ...)   ASYNC_LOG_C_MAYBE(ddLogLevel, LOG_FLAG_INFO,   frmt, ##__VA_ARGS__)
#define DDLogCDebug(frmt, ...)  ASYNC_LOG_C_MAYBE(ddLogLevel, LOG_FLAG_DEBUG,  frmt, ##__VA_ARGS__)
```