The UHD logging facility

Contents

AN-001 by Michael Dickens

This application note describes the UHD logging facility: compile-time options, runtime options, application usage, and default values ... and how all of these interact.

UHD has 3 primary files associated with logging: log.hpp, log.cpp, and UHDLog.cmake. There are secondary logging files providing an interface to C and Python.

There are 7 logging levels; the name and number can be used interchangeably for any logging variable.

UHD Logging Levels

trace	debug	info	warning	error	fatal	off
0	1	2	3	4	5	6

There are 2 distinct times where logging can be controlled: during compilation per CMake settings, and during runtime via shell environment variables.

There are 10 CMake variables that control logging:

- 6 are boolean: `UHD_LOG_FASTPATH_DISABLE`, `UHD_LOG_CONSOLE_DISABLE`, `UHD_LOG_CONSOLE_COLOR`, `UHD_LOG_CONSOLE_TIME`, `UHD_LOG_CONSOLE_THREAD`, and `UHD_LOG_CONSOLE_SRC`;
- 3 set log levels: `UHD_LOG_MIN_LEVEL`, `UHD_LOG_FILE_LEVEL`, and `UHD_LOG_CONSOLE_LEVEL`;
- 1 sets the default log file: `UHD_LOG_FILE`.

Each of these variables can be set independently via the CMake commandline or other methods, and override each's default value as set in the UHD CMake scripts. The default values for 2 of these variables depends on whether the `CMAKE_BUILD_TYPE` is `Debug` or not, as follows:

UHD CMake Logging Variables and Default Values for different `CMAKE_BUILD_TYPE`

	Default Value	
Variable	*non-`Debug`*	*`Debug`*
`UHD_LOG_FILE_LEVEL`	`info`	`trace`
`UHD_LOG_CONSOLE_LEVEL`	`info`	`debug`

Here is a description of the 10 CMake variables the influence logging:

- The variable `UHD_LOG_CONSOLE_DISABLE` sets whether to disable (if `ON`) or enable (if `OFF`) console-based logging; the default is `OFF` such that console logging is enabled. When set to `ON`, console-based logging is *not* compiled into UHD at all, and so any `CONSOLE` log setting will have no effect.
- The variable `UHD_LOG_FASTPATH_DISABLE` sets whether to disable (if `ON`) or enable (if `OFF`) fastpath logging; the default is `OFF` such that fastpath logging is enabled. Fastpath logging is used for Extra-fast logging macro for when speed matters; no metadata is tracked, and only the message is displayed. This logging is used for printing the `UOSDL` characters during streaming to show underrun, overflow, and other situations.
- The variable `UHD_LOG_CONSOLE_COLOR` sets whether to disable (if `OFF`) or enable (if `ON`) colorization of logging text via special terminal character sequences. By default on Microsoft Windows or Cygwin, colorization is disabled; otherwise it is enabled.
- The variable `UHD_LOG_CONSOLE_TIME` sets whether to disable (if `OFF`) or enable (if `ON` the current timestamp as part of the log message. By default this variable is not set, which means it is `OFF`. This variable must be set to `ON` (or the equivalent) for this logging to be enabled.
- The variable `UHD_LOG_CONSOLE_THREAD` sets whether to disable (if `OFF`) or enable (if `ON` the thread ID displaying the log as part of the log message. By default this variable is not set, which means it is `OFF`. This variable must be set to `ON` (or the equivalent) for this logging to be enabled.
- The variable `UHD_LOG_CONSOLE_SRC` sets whether to disable (if `OFF`) or enable (if `ON` the source code filename and line displaying the log as part of the log message. By default this variable is not set, which means it is `OFF`. This variable must be set to `ON` (or the equivalent) for this logging to be enabled.
- The variable `UHD_LOG_MIN_LEVEL` sets the minimum logging level for *any* logging functionality. For example, when this variable is set to `debug` (the default), then `trace` level debugging will not be available for *any* logging; only `debug` and higher (up to `fatal`) will be available. Setting this variable to `off` will disable runtime logging, but all enabled logging will still be compiled as part of UHD.
- The variable `UHD_LOG_FILE_LEVEL` sets the default value for file-based logging. This value can be changed for runtime to any value `UHD_LOG_MIN_LEVEL` or higher.
- The variable `UHD_LOG_CONSOLE_LEVEL` sets the default value for console-based logging. This value can be changed for runtime to any value `UHD_LOG_MIN_LEVEL` or higher.
- The variable `UHD_LOG_FILE` sets the default file to use for file-based logging. By default this variable is not set, which means that there is no default file to write file-based logging into. If not set during compile-time, then to enable file-based logging the eponymous runtime variable must be set, as noted below.

There are 5 runtime environment variables that control logging. Each of these variables can be set independently via a shell environment, on the commandline, or other methods, and override each's default value as set by the aforementioned CMake variables.

- The variable `UHD_LOG_LEVEL` sets the minimum logging level for *any* logging functionality. For example, when this variable is set to `debug` (the default), then `trace` level debugging will not be available for *any* logging; only `debug` and higher (up to `fatal`) will be available.
- The variable `UHD_LOG_FILE_LEVEL` sets the default value for file-based logging. This value can be changed for runtime to any value `UHD_LOG_MIN_LEVEL` or higher.
- The variable `UHD_LOG_FILE` sets the file to use for file-based logging. If no log file is set at compile-time or runtime, then no file logging will take place. Any file specified for logging must be able to be opened for writing to (but not necessarily reading from by the UHD process); any existing file will be appended to.
- The variable `UHD_LOG_CONSOLE_LEVEL` sets the default value for console-based logging. This value can be changed for runtime to any value `UHD_LOG_MIN_LEVEL` or higher.
- The variable `UHD_LOG_FASTPATH_DISABLE` sets whether to disable fastpath logging, if it was not disabled at compile-time.