About LLDB and Debugging

Debugging refers to creating and using an analytical framework to isolate causal pathways and test hypotheses. The most important tool of debugging is the debugger, which helps you understand how your program behaves at runtime, without modifying the code.

LLDB provides the underlying debugging environment for developers on Apple platforms. You can use it from a Terminal window or an Xcode source editor to find and eliminate problems in your Swift, C, C++, and Objective-C code.

Note: This document focuses on the functionality of LLDB provided through its command-line prompt. For more information on using LLDB from Xcode, see *Debugging with Xcode*.

At a Glance

A debugger has two primary functions: *controlling execution flow* and *accessing state*.

You primarily control the execution of a program by setting *breakpoints* at different locations in code. Whenever a program hits a set breakpoint, the debugger temporarily stops execution of the program. While the execution is stopped, you can use the debugger to inspect or modify the current state of different variables, step over, into, or out of the next statement, and continue execution as needed.

See Also

- WWDC 2013: Advanced Debugging with LLDB
- *Debugging with Xcode*
- The Official LLDB Website