

	[Comment deleted]					
	tm@gmail.com <tm@gmail.com><u>#5</u></tm@gmail.com>	Jul 14, 2016 03:18AM	:			
	We updated to 'r12b' and we can still reproduce this on our app. Your example works fine though. I have some hypothesis: - We don't link statically with libc++. If you change your example to c++_shared it doesn't link, undefined reference to log2. - That said, I'm building a library that gets loaded at runtime, that is probably why I don't see the linking errors. - The fact the runtime loader doesn't tell me about unresolved symbols is because I'm getting the symbol resolved somehow. Probably from somewhere not the libc++, that is why the weird results from log2 I'm getting. The log2 symbol not visible on the shared version of libc++ seems to be the root cause IMO.					
					da@google.com <da@google.com><u>#6</u></da@google.com>	Jul 16, 2016 05:36AM
Assigned to da@google.com.						
Confirmed that log2 is unresolved with c++_shared. I'll take a look.						
	da@google.com <da@google.com><u>#7</u></da@google.com>	Jul 16, 2016 08:44AM	:			
	Should be fixed with these two patches: https://android-review.googlesource.com/#/q/topic:android_support_ordering					
	da@google.com <da@google.com></da@google.com>	Aug 12, 2016 03:4	3AV			
	Marked as fixed.					
	be@gmail.com <be@gmail.com><u>#8</u></be@gmail.com>	Oct 11, 2016 01:00PM	:			
	When using cmake in the latest release of NDK & CMake with android studio 2.2.1 I get this error too. Undefined reference to log2. I'm also using c++_shared. Interesting regular std::log works.					
	Android Studio 2.2.1 Build #AI-145.3330264, built on October 6, 2016 JRE: 1.8.0_76-release-b03 amd64					