NSNotificationCenter

In OS X 10.11 and iOS 9.0 NSNotificationCenter and NSDistributedNotificationCenter will no longer send notifications to registered observers that may be deallocated. If the observer is able to be stored as a zeroing-weak reference the underlying storage will store the observer as a zeroing weak reference, alternatively if the object cannot be stored weakly (i.e. it has a custom retain/release mechanism that would prevent the runtime from being able to store the object weakly) it will store the object as a non-weak zeroing reference. This means that observers are not required to un-register in their deallocation method. The next notification that would be routed to that observer will detect the zeroed reference and automatically un-register the observer. If an object can be weakly referenced notifications will no longer be sent to the observer during deallocation; the previous behavior of receiving notifications during dealloc is still present in the case of non-weakly zeroing reference observers. Block based observers via the -[NSNotificationCenter addObserverForName:object:queue:usingBlock] method still need to be un-registered when no longer in use since the system still holds a strong reference to these observers. Removing observers (either weakly referenced or zeroing referenced) prematurely is still supported. CFNotificationCenterAddObserver does not conform to this behavior since the observer may not be an object.

NSNotificationCenter and NSDistributedNotificationCenter will now provide a debug description when printing from the debugger that will list all registered observers including references that have been zeroed out for aiding in debugging notification registrations. This data is only valid per the duration of the breakpoint since the underlying store must account for multithreaded environments. Wildcard registrations for notifications where the name or object that was passed to the addObserver method family was null will be printed in the debug description as *.