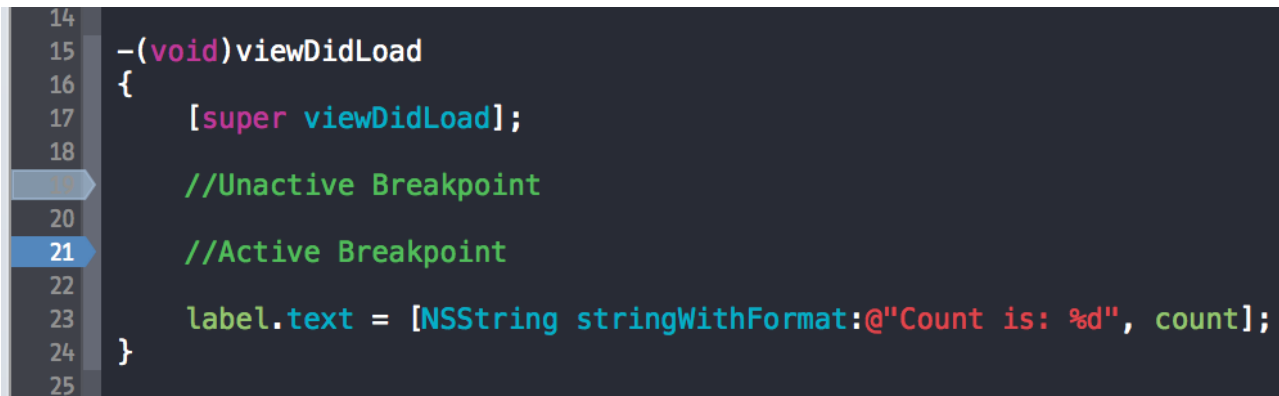


Breakpoints

Breakpoints are points in your code where execution will break when it reaches. They are great for finding out where exceptions occur and figuring out where logical errors are.

Basics

To create a simple breakpoint, click on the line number you want to add your breakpoint to. You can disable the breakpoint by clicking on it again, you can also click on it again to activate it.



```
14
15 -(void)viewDidLoad
16 {
17     [super viewDidLoad];
18
19     //Unactive Breakpoint
20
21     //Active Breakpoint
22
23     label.text = [NSString stringWithFormat:@"Count is: %d", count];
24 }
25
```

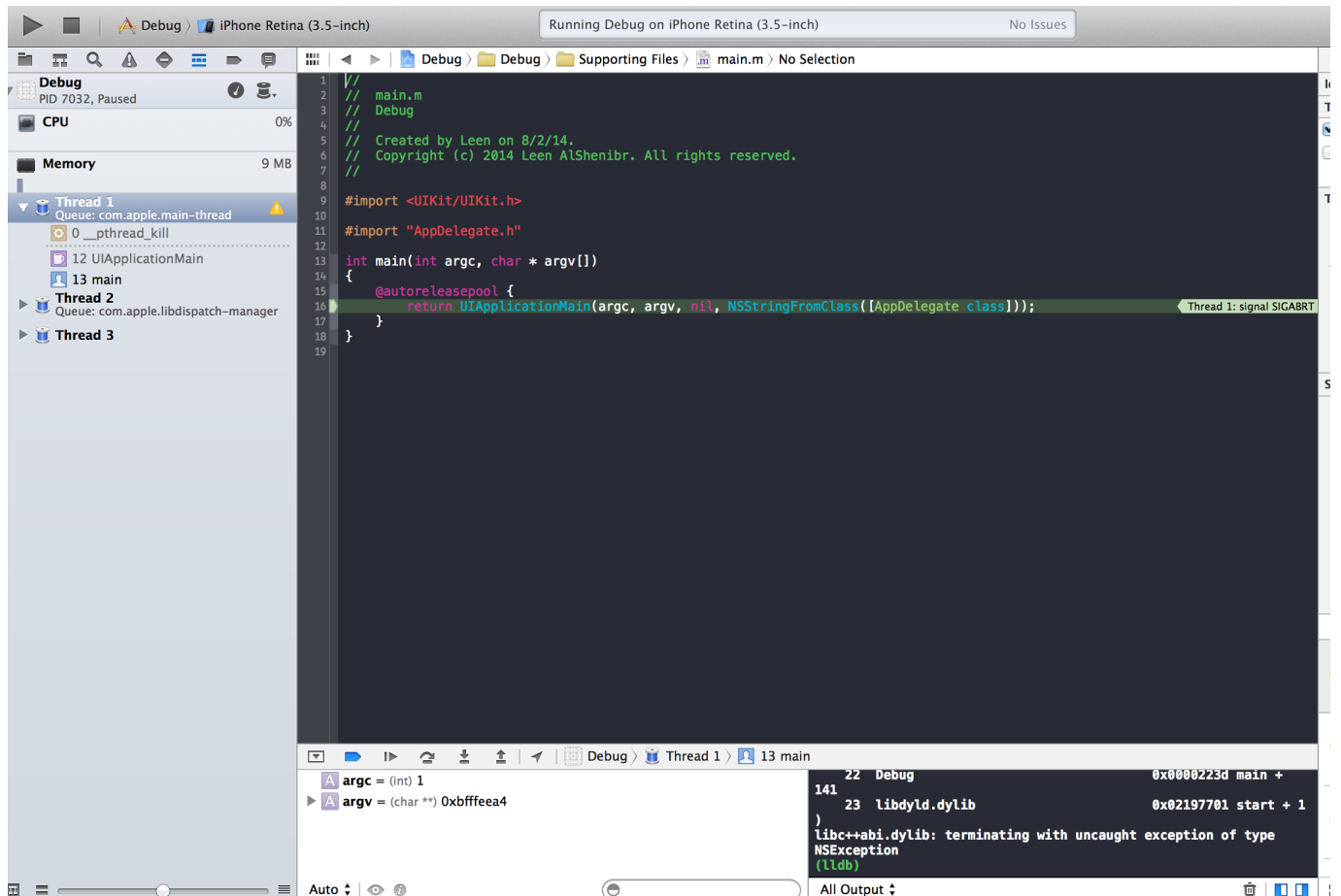
The screenshot shows a code editor with a dark background. On the left, a vertical line of numbers from 14 to 25 is visible. Line 19 has a light blue arrow pointing to it, indicating an unactive breakpoint. Line 21 has a dark blue arrow pointing to it, indicating an active breakpoint. The code is in Objective-C and shows a viewDidLoad method with a comment and a line of code that sets the text of a label.

You can delete the breakpoint by right clicking on the breakpoint and then "Delete Breakpoint". It's easier to use the Breakpoint Navigator to delete multiple breakpoints.

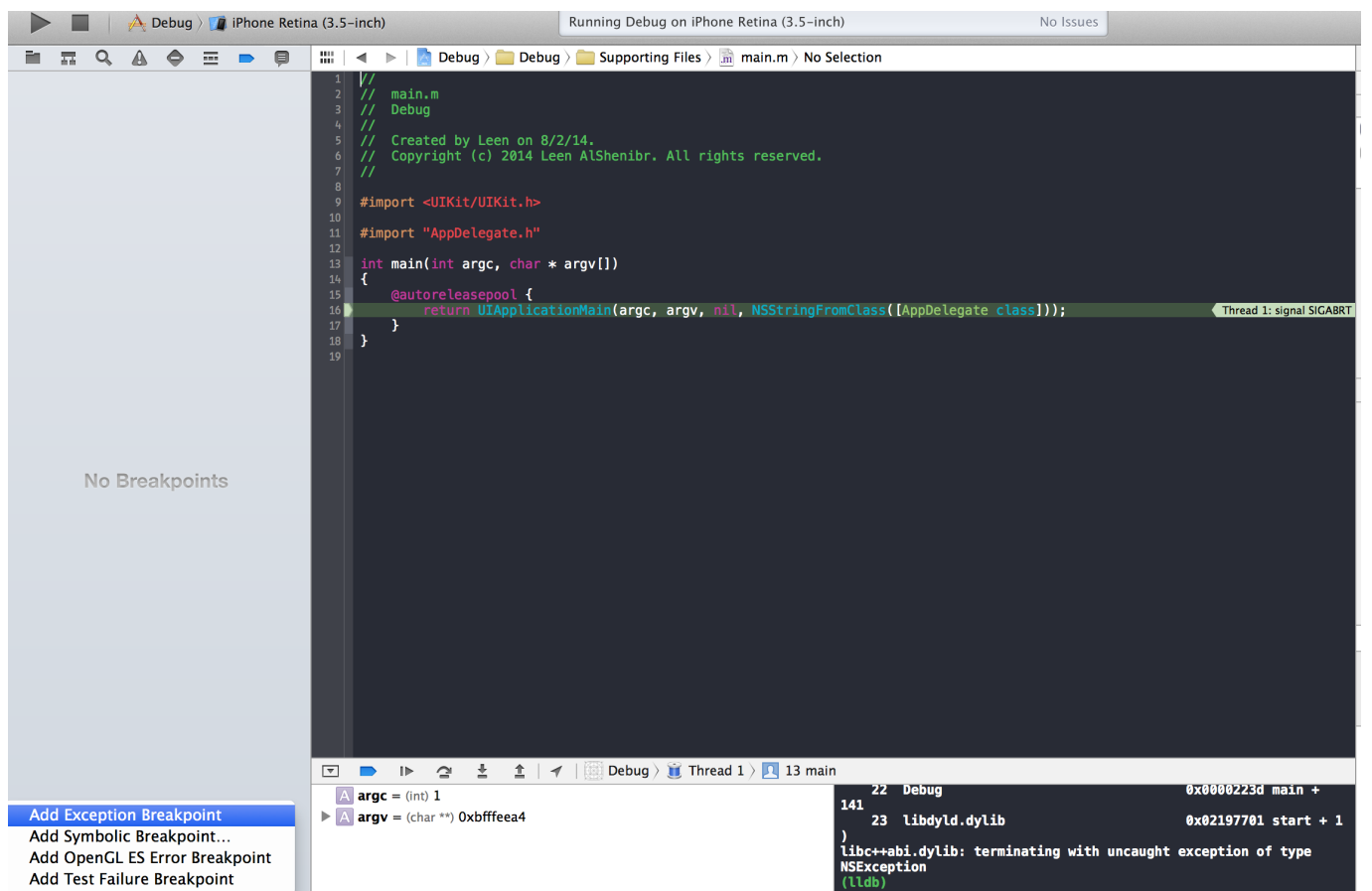
Exception Breakpoints

You can use this breakpoint to get some context on which line of code caused this exceptions.

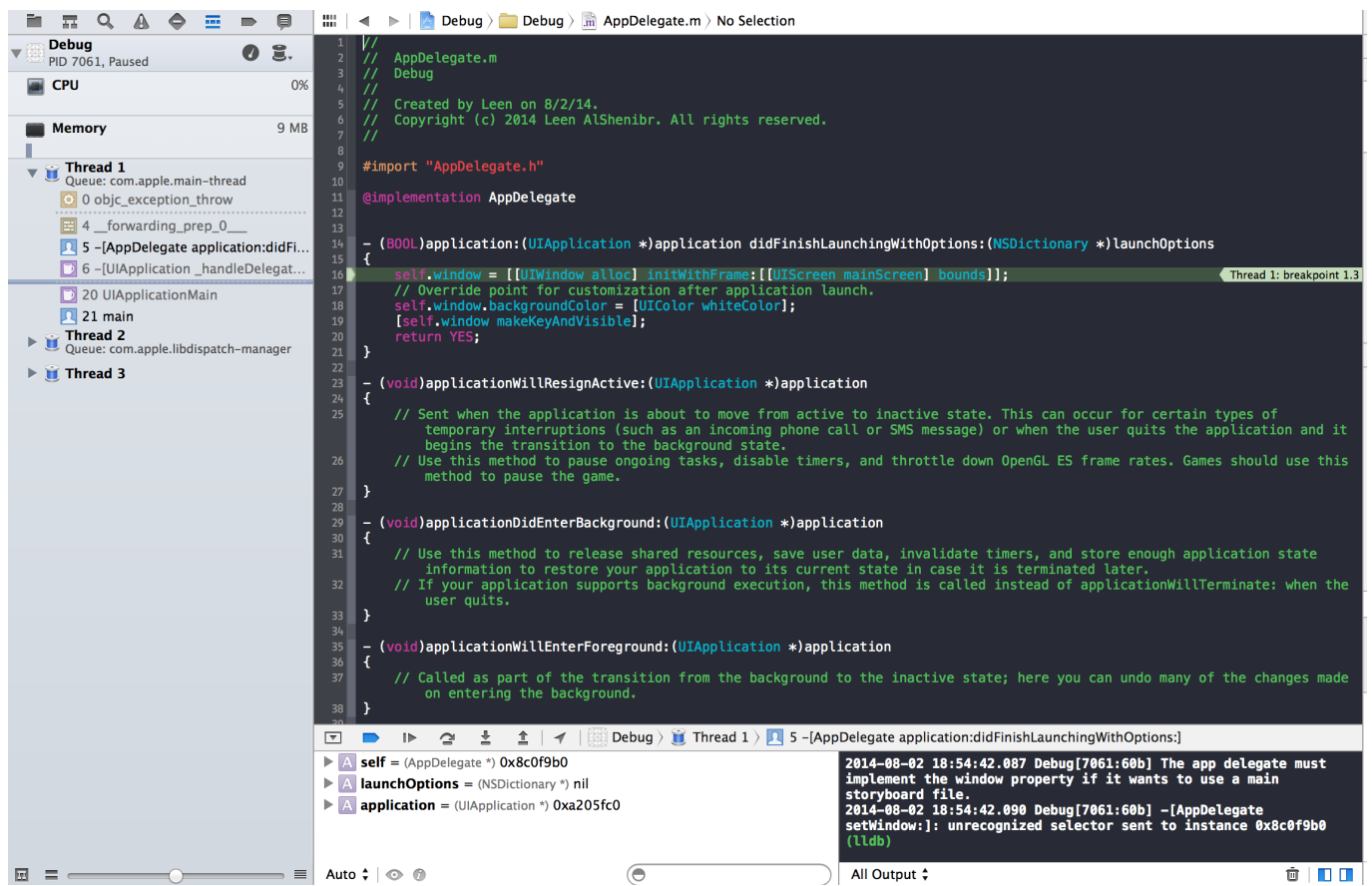
Instead of getting this screen:



Add an Exception: 1. Go to the Breakpoint Navigator 2. Click on add icon on the bottom of the page. 3. Click add Exception Breakpoint.



Now all the breakpoints will be caught before they are thrown, it will point to the line that was called right before the exception was about to be thrown.



If you were still shown that the usual exception screen, then the problem might involve outlets in the storyboard. Delete the Exception Breakpoint run the app again. Look at the first part of the Terminating app due to uncaught exception print out in the console.

Example:

```
*** Terminating app due to uncaught exception 'NSUnknownKeyException', reason:
'<ViewController 0x8c225c0> setValue:forUndefinedKey:]: this class is not key
value coding-compliant for the key button.'
*** First throw call stack:
(
    //Cut out to keep this short.
)
libc++abi.dylib: terminating with uncaught exception of type NSException
```

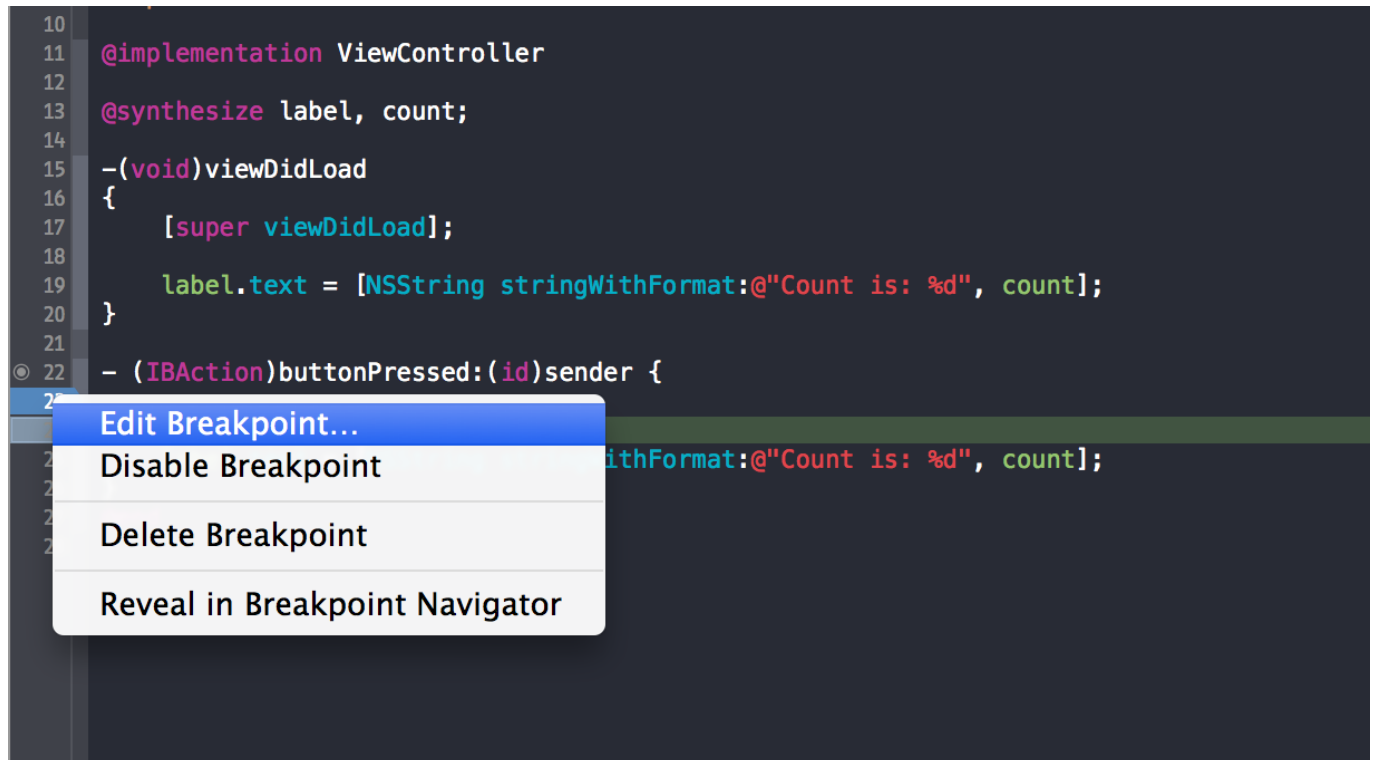
This log says that there was an outlet created in the class "ViewController" that made a connection with the name "button", but the connection can't be found in ViewController. This usually happens when you delete outlets you created but didn't delete the connection from the items "Connections Inspector" the arrow like icon on the left pane.

These types of inspections are usually hard to trace because the actual item doesn't need to be named "button" so you need to inspect most of the items inside that view to find the error.

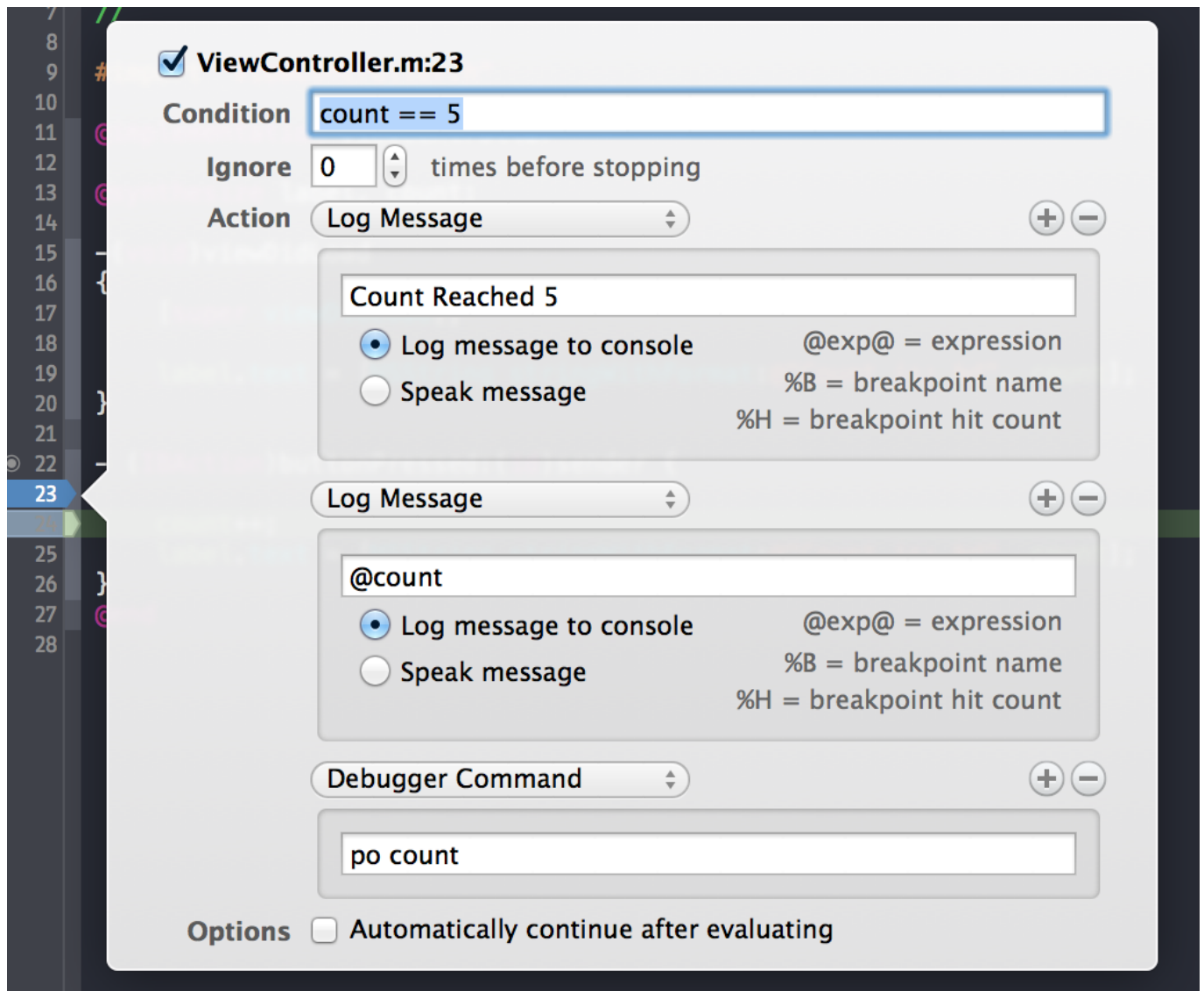
Conditional Breakpoints

Conditionals Breakpoints, are breakpoints that break when a condition evaluates to true. These can be used to do a wide array of things. Keep track of variables value, coupled with continued execution.

Click on the margin of the editor to create a breakpoint. Right click on the breakpoint and click "Edit Breakpoint"



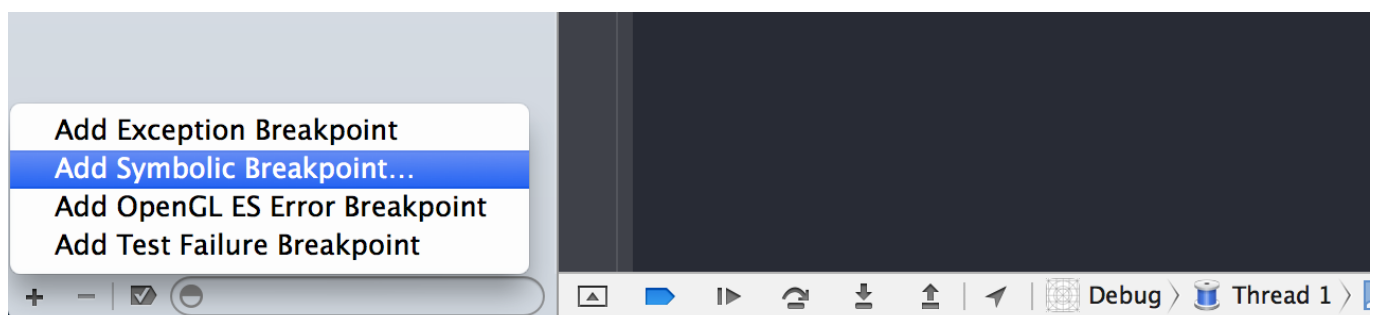
You can add a condition and multiple actions to the breakpoint as below. You can also let the app continue execution when it reaches the breakpoint so it doesn't interrupt execution every time it reaches that breakpoint by checking the box at the bottom.



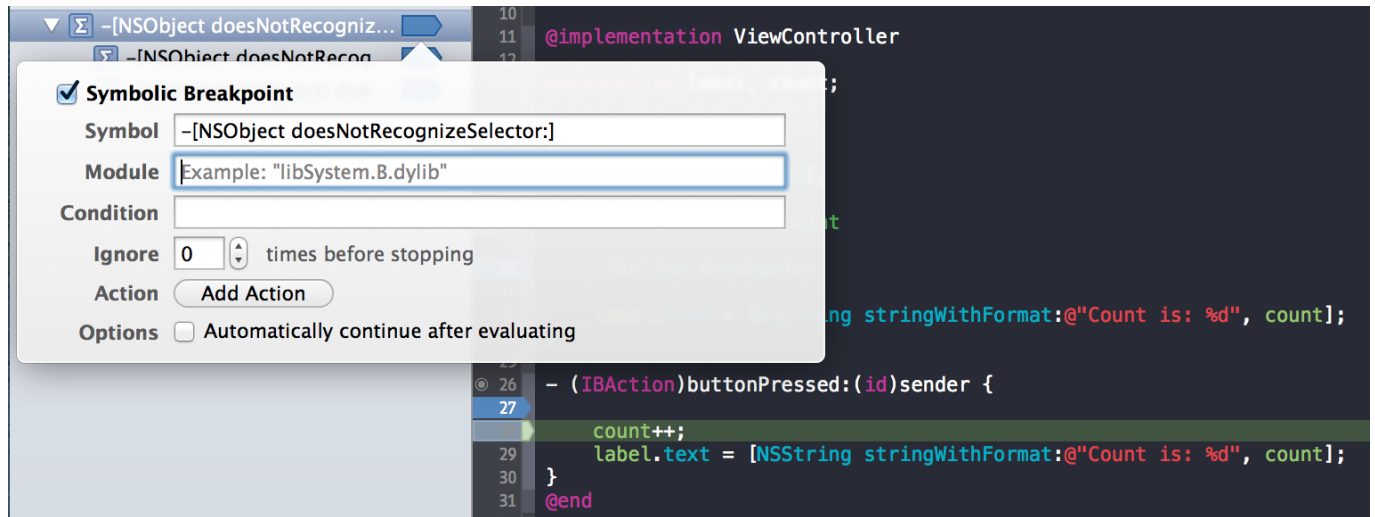
Symbolic Breakpoints

You can use these kind of breakpoints to stop the execution of code that you don't have the source code of. For Example, we are interested in catching calls to unrecognized selectors.

We can create add a new breakpoint by clicking the plus button in the Breakpoint Navigator.



Then choosing `Add Symbolic Breakpoint`. Add this `-[NSObject doesNotRecognizeSelector:]` selector in the symbol field.



Now the execution will stop whenever that selector is called.

Additional Links:

- [Apple's Breakpoint Navigator Help Documentation.](#)
- [Short Breakpoint Overview](#)