

# Debugging in Xcode

Session 407

**Troy Koelling**

Xcode Debugger UI Engineer

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

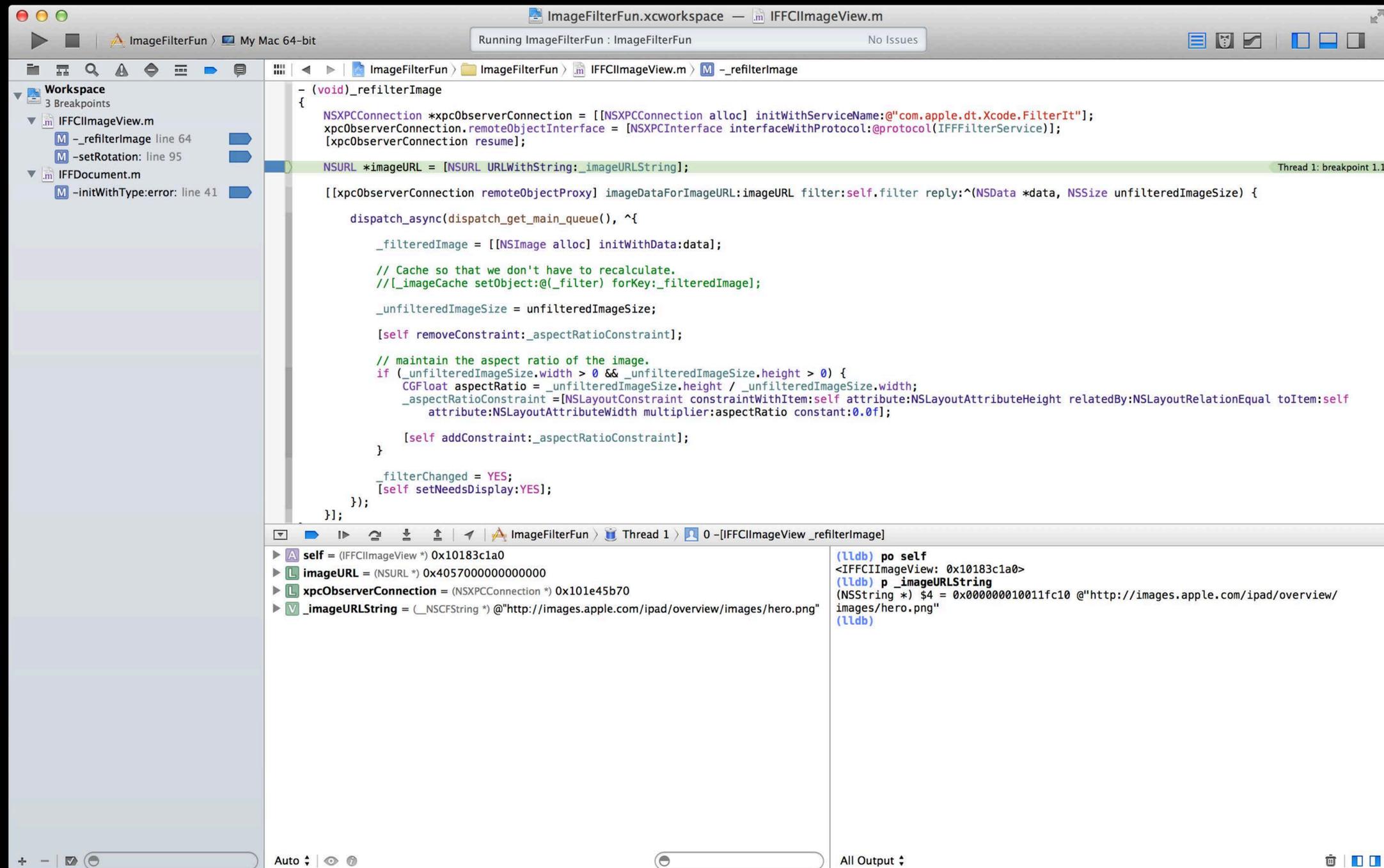


“Everyone knows that **debugging is twice as hard** as writing a program in the first place. So if you’re as clever as you can be when you write it, how will you ever debug it?”

Brian Kernighan

# Debugging in Xcode 5

# Debugging in Xcode 5







```

[[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, NSSize unfi

dispatch_async(dispatch_get_main_queue(), ^{

    _filteredImage = [[UIImage alloc] initWithData:data];

    // Cache so that we don't have to recalculate.
    //[_imageCache setObject:@(_filter) forKey:_filteredImage];

    _unfilteredImageSize = unfilteredImageSize;

    [self removeConstraint:_aspectRatioConstraint];

    // maintain the aspect ratio of the image.
    if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
        CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
        _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:
            attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

        [self addConstraint:_aspectRatioConstraint];
    }

    _filterChanged = YES;
    [self setNeedsDisplay:YES];
});
}];

```

ImageFilterFun > Thread 1 > 0 -[IFFCImageView \_refilterImage]

- ▶ **A** self = (IFFCImageView \*) 0x10183c1a0
- ▶ **L** imageURL = (NSURL \*) 0x4057000000000000
- ▶ **L** xpcObserverConnection = (NSXPCConnection \*) 0x101e45b70
- ▶ **V** \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

```

(lldb) po self
<IFFCImageView: 0x10183c1a0>
(lldb) p _imageURLString
(NSString *) $4 = 0x000000010011fc10 @"http://
images/hero.png"
(lldb)

```

```

[[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, NSSize unfi

dispatch_async(dispatch_get_main_queue(), ^{

    _filteredImage = [[UIImage alloc] initWithData:data];

    // Cache so that we don't have to recalculate.
    //[_imageCache setObject:@(_filter) forKey:_filteredImage];

    _unfilteredImageSize = unfilteredImageSize;

    [self removeConstraint:_aspectRatioConstraint];

    // maintain the aspect ratio of the image.
    if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
        CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
        _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:M
            attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

        [self addConstraint:_aspectRatioConstraint];
    }

    _filterChanged = YES;
    [self setNeedsDisplay:YES];
});
}];

```

ImageFilterFun > Thread 1 > 0 -[IFFCImageView \_refilterImage]

- ▶ self = (IFFCImageView \*) 0x10183c1a0
- ▶ imageURL = (NSURL \*) 0x4057000000000000
- ▶ xpcObserverConnection = (NSXPCConnection \*) 0x101e45b70
- ▶ \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

```

(lldb) po self
<IFFCImageView: 0x10183c1a0>
(lldb) p _imageURLString
(NSString *) $4 = 0x000000010011fc10 @"http://
images/hero.png"
(lldb)

```

```

[[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, NSSize unfi

dispatch_async(dispatch_get_main_queue(), ^{

    _filteredImage = [[UIImage alloc] initWithData:data];

    // Cache so that we don't have to recalculate.
    //[_imageCache setObject:@(_filter) forKey:_filteredImage];

    _unfilteredImageSize = unfilteredImageSize;

    [self removeConstraint:_aspectRatioConstraint];

    // maintain the aspect ratio of the image.
    if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
        CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
        _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:M
            attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

        [self addConstraint:_aspectRatioConstraint];
    }

    _filterChanged = YES;
    [self setNeedsDisplay:YES];
});
}];

```

ImageFilterFun > Thread 1 > 0 -[IFFCImageView \_refilterImage]

- ▶ **A** self = (IFFCImageView \*) 0x10183c1a0
- ▶ **L** imageURL = (NSURL \*) 0x4057000000000000
- ▶ **L** xpcObserverConnection = (NSXPCConnection \*) 0x101e45b70
- ▶ **V** \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

```

(lldb) po self
<IFFCImageView: 0x10183c1a0>
(lldb) p _imageURLString
(NSString *) $4 = 0x000000010011fc10 @"http://
images/hero.png"
(lldb)

```

```

[[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, NSSize unfi

dispatch_async(dispatch_get_main_queue(), ^{

    _filteredImage = [[UIImage alloc] initWithData:data];

    // Cache so that we don't have to recalculate.
    //[_imageCache setObject:@(_filter) forKey:_filteredImage];

    _unfilteredImageSize = unfilteredImageSize;

    [self removeConstraint:_aspectRatioConstraint];

    // maintain the aspect ratio of the image.
    if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
        CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
        _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:
            attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

        [self addConstraint:_aspectRatioConstraint];
    }

    _filterChanged = YES;
    [self setNeedsDisplay:YES];
});
}];

```

ImageFilterFun > Thread 1 > 0 -[IFFCImageView \_refilterImage]

- ▶ **A** self = (IFFCImageView \*) 0x10183c1a0
- ▶ **L** imageURL = (NSURL \*) 0x4057000000000000
- ▶ **L** xpcObserverConnection = (NSXPCConnection \*) 0x101e45b70
- ▶ **V** \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

```

(lldb) po self
<IFFCImageView: 0x10183c1a0>
(lldb) p _imageURLString
(NSString *) $4 = 0x000000010011fc10 @"http://
images/hero.png"
(lldb)

```

```

[[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, NSSize unfi

dispatch_async(dispatch_get_main_queue(), ^{

    _filteredImage = [[UIImage alloc] initWithData:data];

    // Cache so that we don't have to recalculate.
    //[_imageCache setObject:@(_filter) forKey:_filteredImage];

    _unfilteredImageSize = unfilteredImageSize;

    [self removeConstraint:_aspectRatioConstraint];

    // maintain the aspect ratio of the image.
    if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
        CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
        _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:M
            attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

        [self addConstraint:_aspectRatioConstraint];
    }

    _filterChanged = YES;
    [self setNeedsDisplay:YES];
});
}];

```

ImageFilterFun > Thread 1 > 0 -[IFFCImageView \_refilterImage]

- ▶ **A** self = (IFFCImageView \*) 0x10183c1a0
- ▶ **L** imageURL = (NSURL \*) 0x4057000000000000
- ▶ **L** xpcObserverConnection = (NSXPCConnection \*) 0x101e45b70
- ▶ **V** \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

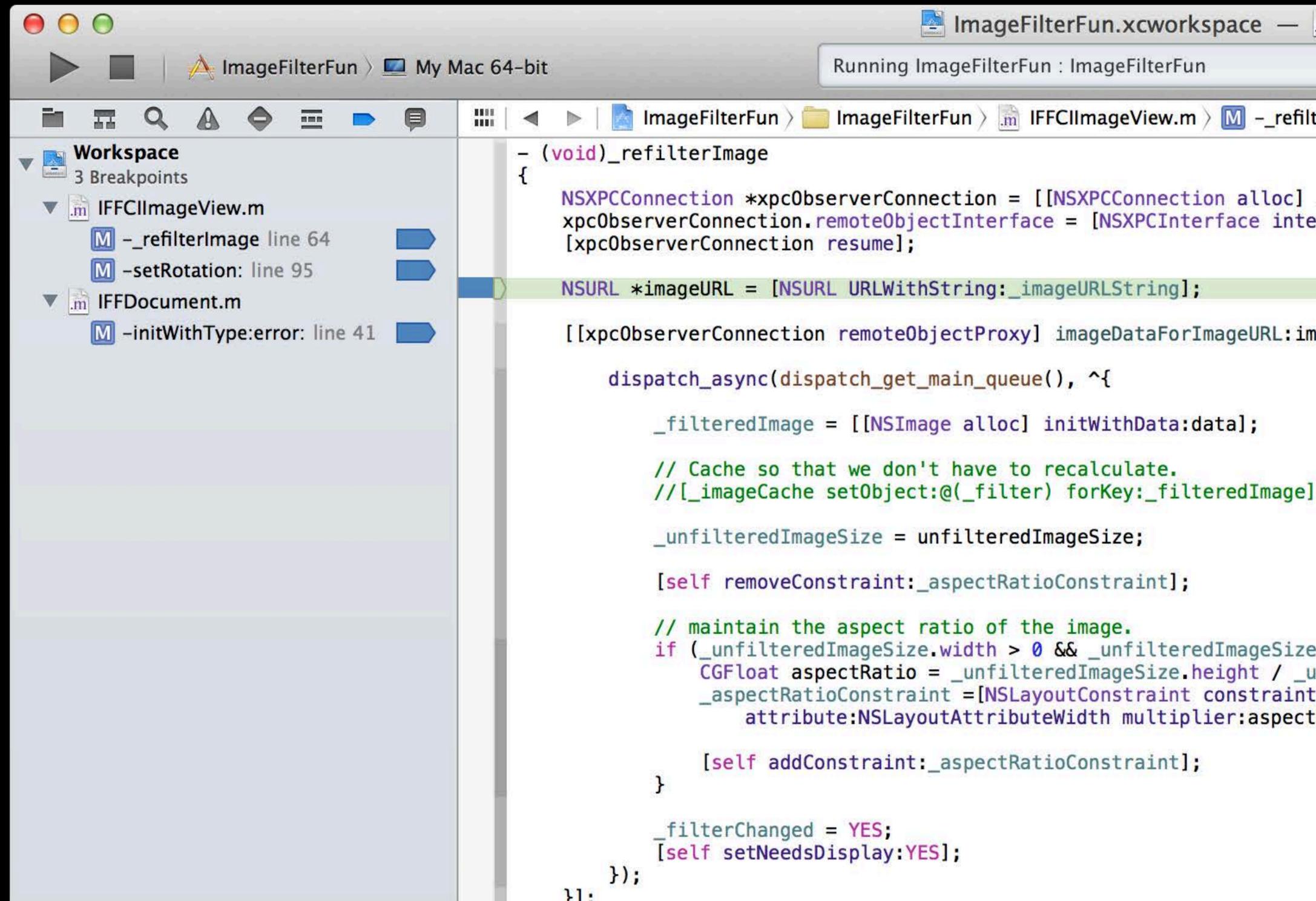
```

(lldb) po self
<IFFCImageView: 0x10183c1a0>
(lldb) p _imageURLString
(NSString *) $4 = 0x000000010011fc10 @"http://
images/hero.png"
(lldb)

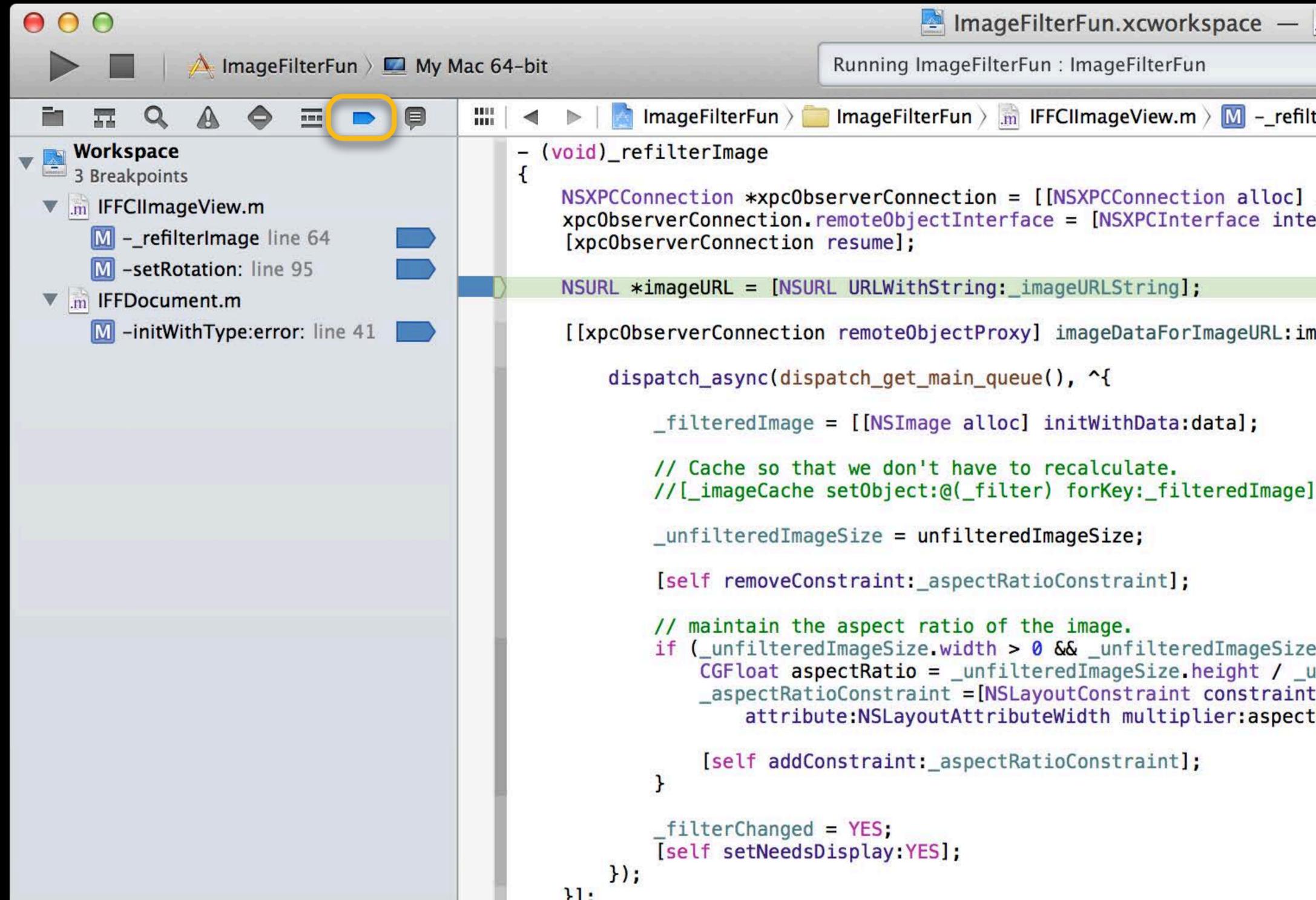
```



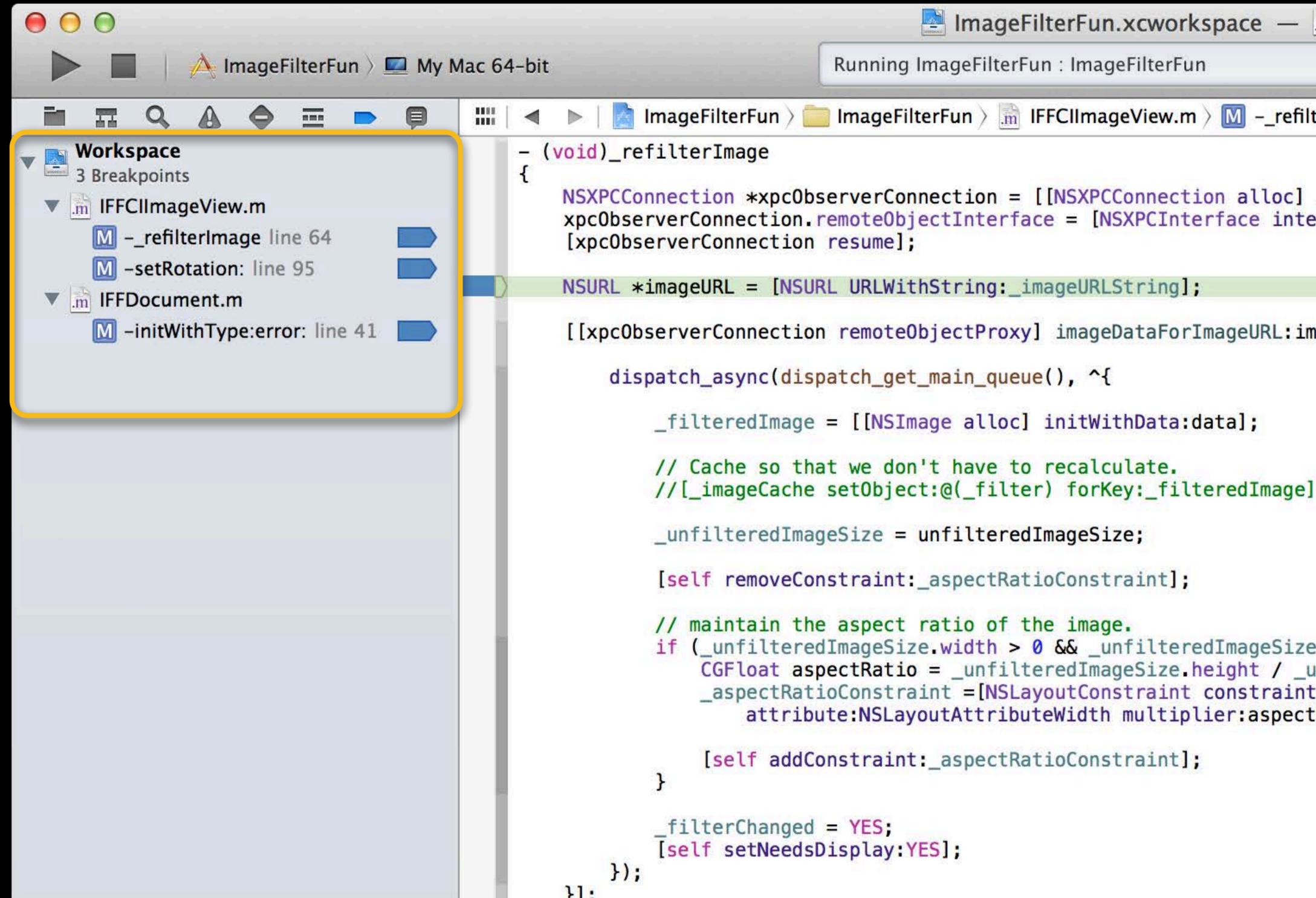
# Breakpoints Navigator



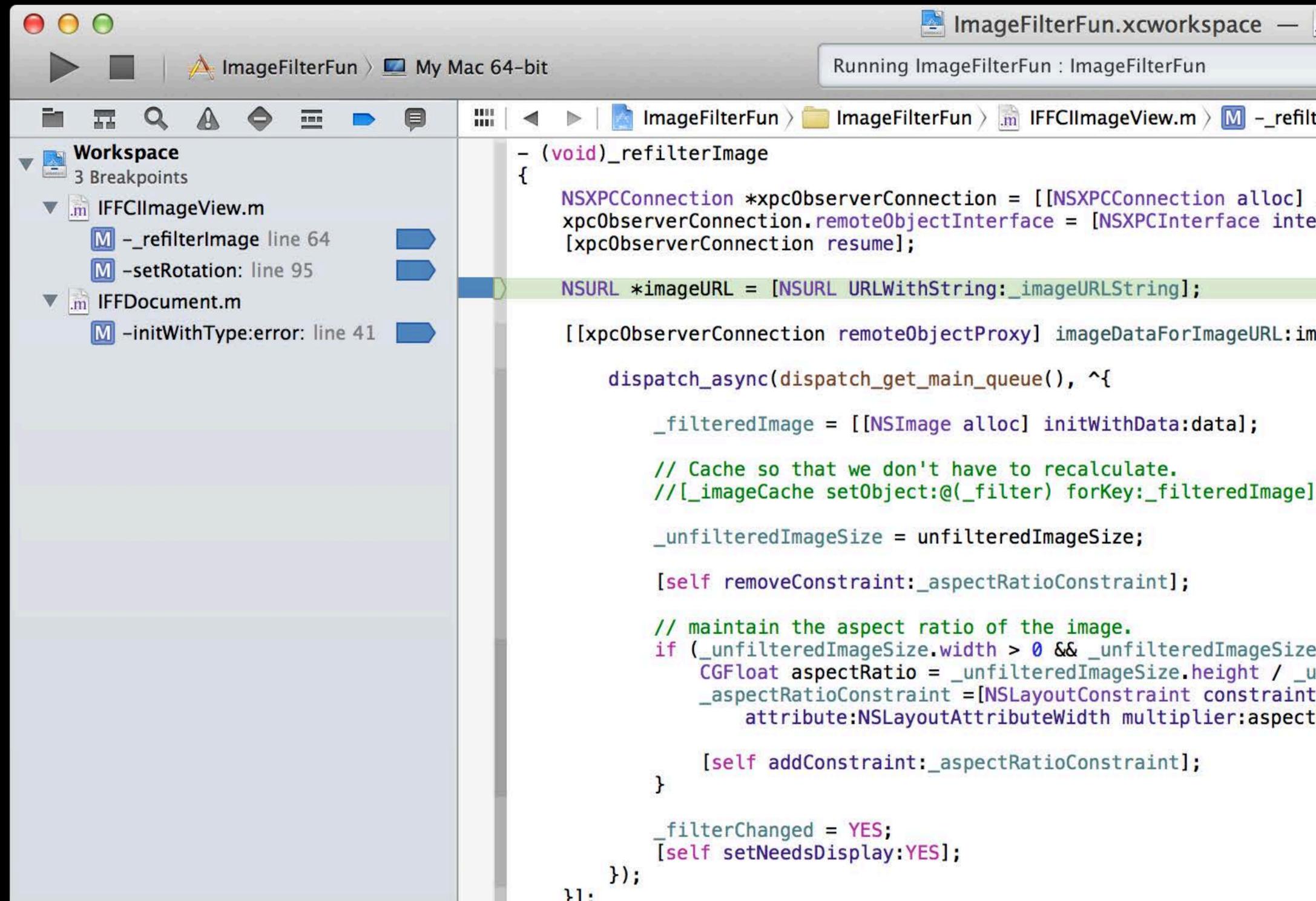
# Breakpoints Navigator



# Breakpoints Navigator



# Breakpoints Navigator



```
        _filterChanged = YES;  
        [self setNeedsDisplay:YES];  
    });  
};
```

ImageFilterFun > Thread 1 > 0 -[IFFCImage

- ▶ **A** self = (IFFCImageView \*) 0x10183c1a0
- ▶ **L** imageURL = (NSURL \*) 0x4057000000000000
- ▶ **L** xpcObserverConnection = (NSXPCConnection \*) 0x101e45b70
- ▶ **V** \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/image"

+ - | ✓ ○

Auto ⇅ | 🔍 ⓘ

○

```
        _filterChanged = YES;  
        [self setNeedsDisplay:YES];  
    });  
};
```

ImageFilterFun > Thread 1 > 0 -[IFFCI...

- ▶ **A** self = (IFFCImageView \*) 0x10183c1a0
- ▶ **L** imageURL = (NSURL \*) 0x4057000000000000
- ▶ **L** xpcObserverConnection = (NSXPCConnection \*) 0x101e45b70
- ▶ **V** \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/image"

+ - [ ]

Auto [ ] [ ]

[ ]

```
        _filterChanged = YES;  
        [self setNeedsDisplay:YES];  
    });  
};
```

ImageFilterFun > Thread 1 > 0 -[IFFCI...

- ▶ **A** self = (IFFCImageView \*) 0x10183c1a0
- ▶ **L** imageURL = (NSURL \*) 0x4057000000000000
- ▶ **L** xpcObserverConnection = (NSXPCConnection \*) 0x101e45b70
- ▶ **V** \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/image"

+ - | [ ]

Auto | [ ]

- Add Exception Breakpoint...
- Add OpenGL ES Error Breakpoint...
- Add Symbolic Breakpoint...
- Add Test Failure Breakpoint...

```
        _filterChanged = YES;  
        [self setNeedsDisplay:YES];  
    });  
};
```

ImageFilterFun > Thread 1 > 0 -[IFFCI

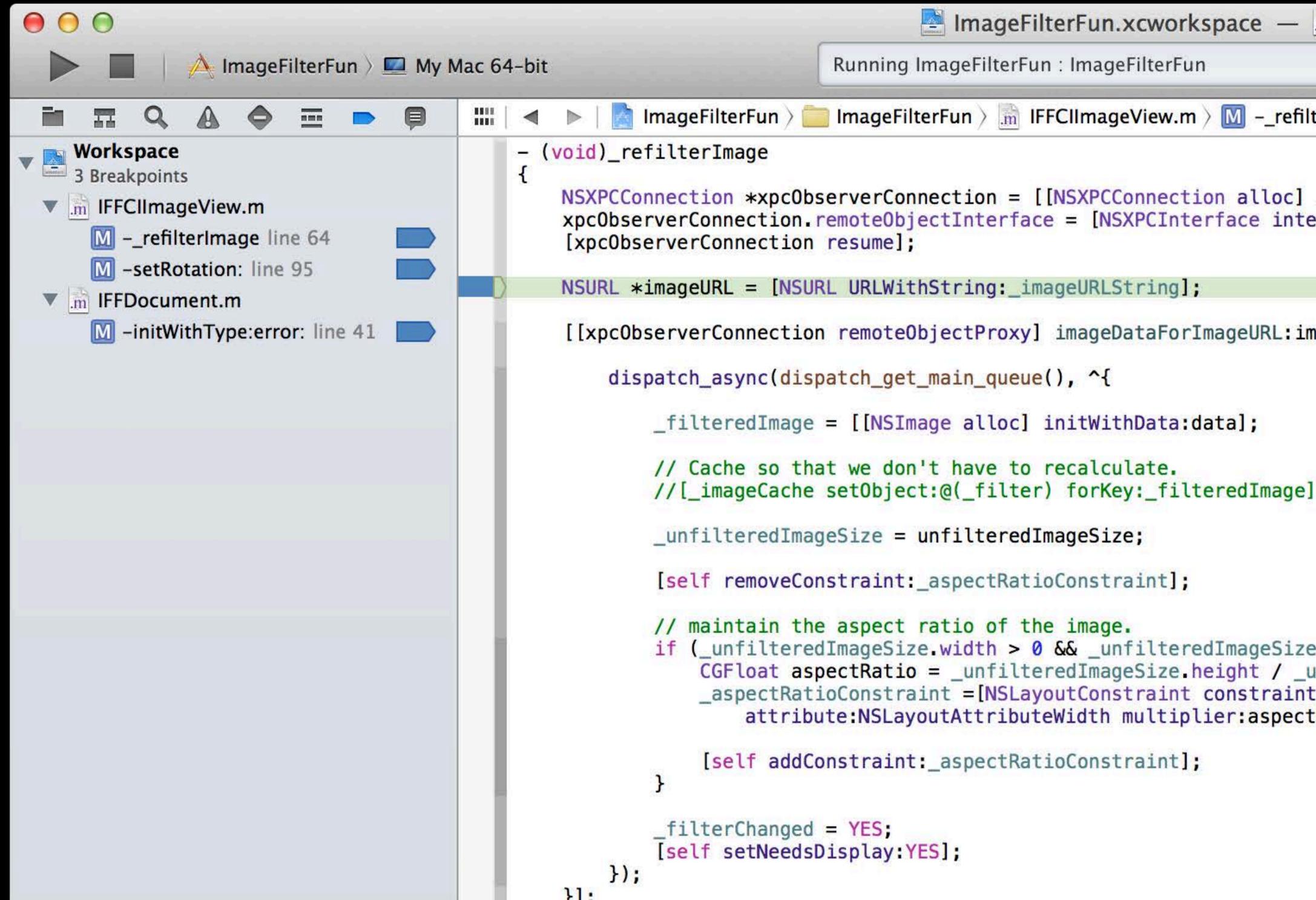
- ▶ **A** self = (IFFCImageView \*) 0x10183c1a0
- ▶ **L** imageURL = (NSURL \*) 0x4057000000000000
- ▶ **L** xpcObserverConnection = (NSXPCConnection \*) 0x101e45b70
- ▶ **V** \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/image

+ - | ✓ ○

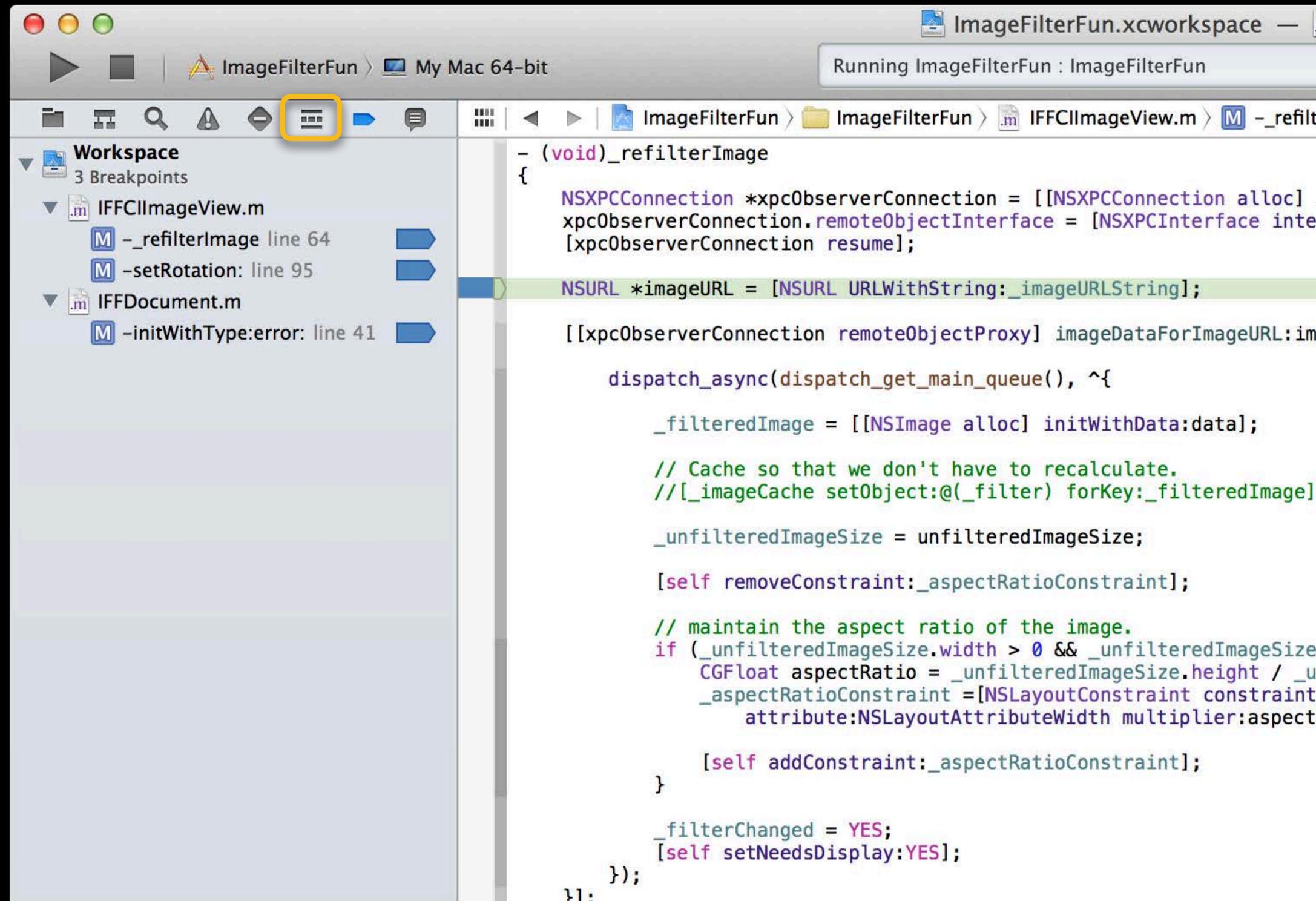
Auto ⬆️ | 🔍 ⓘ

○

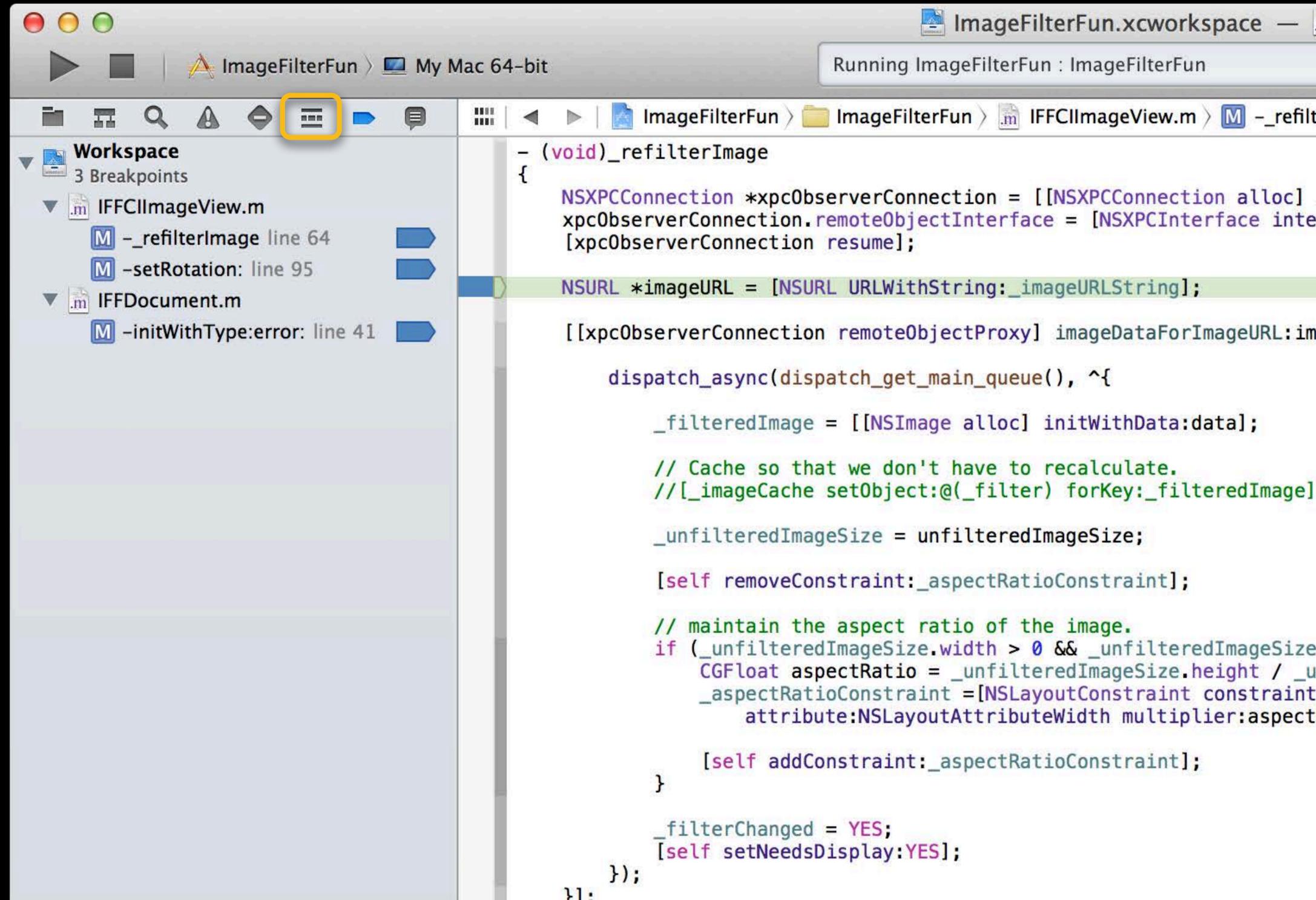
# Breakpoints Navigator



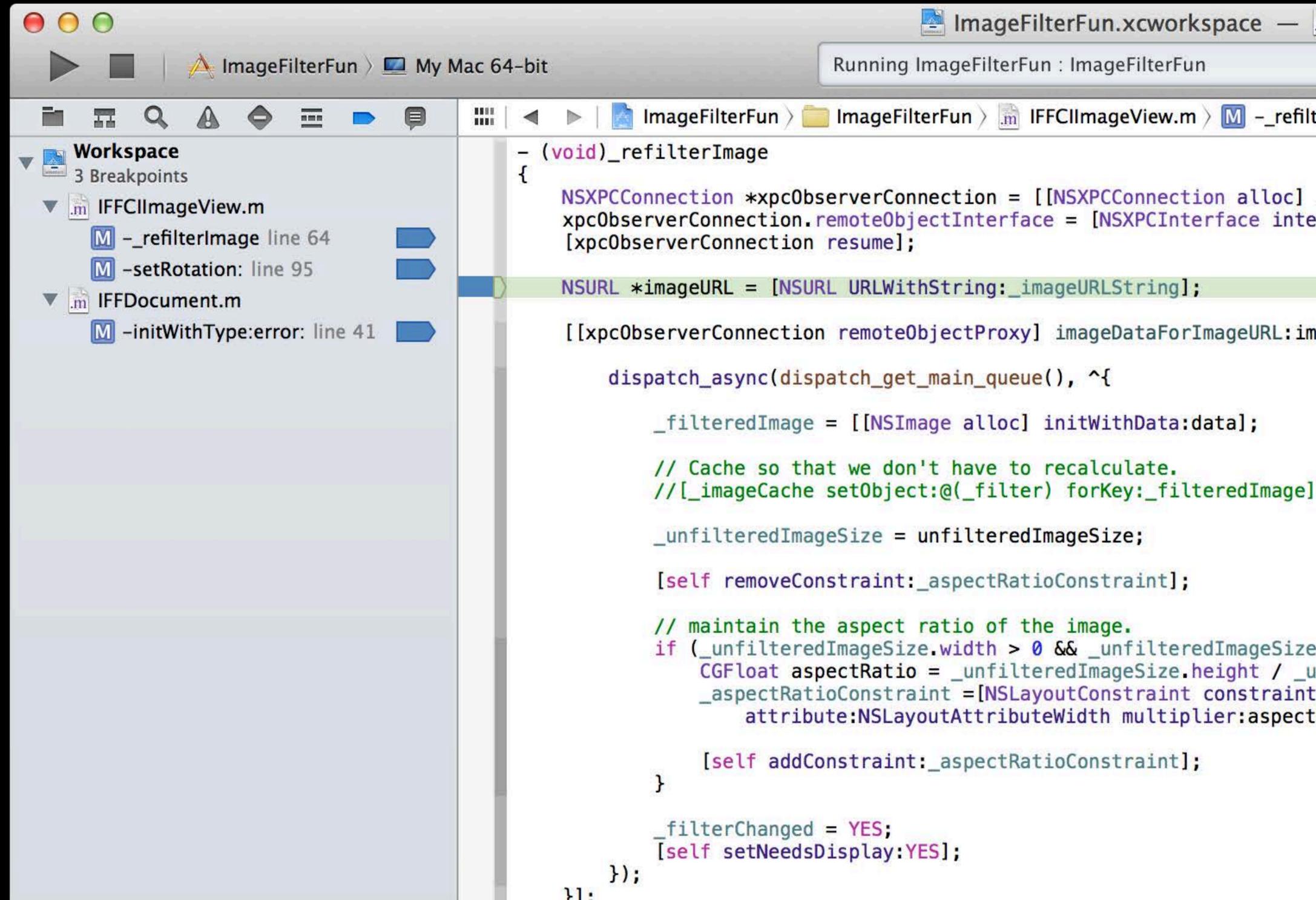
# Breakpoints Navigator



# Debug Navigator



# Debug Navigator



# Debug Navigator

The screenshot shows the Xcode interface for a project named "ImageFilterFun". The top status bar indicates the application is running on a "My Mac 64-bit" and is paused. The left sidebar displays performance metrics: CPU at 0%, Memory at 36 MB, and iCloud. Below these is the "Thread 1" section, which is expanded to show a call stack. The top of the stack is highlighted, showing the method `0 -[IFFCImageView _refilterImage]`. The right pane shows the source code for `IFFCImageView.m`, with the line `NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];` highlighted in green, corresponding to the selected stack frame.

```
ImageFilterFun.xcworkspace —
Running ImageFilterFun : ImageFilterFun

ImageFilterFun > ImageFilterFun > IFFCImageView.m > No Selection

- (void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc]
xpcObserverConnection.remoteObjectInterface = [NSXPCInterface inte
[xpcObserverConnection resume];

    NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:im

        dispatch_async(dispatch_get_main_queue(), ^{

            _filteredImage = [[UIImage alloc] initWithData:data];

            // Cache so that we don't have to recalculate.
            //[_imageCache setObject:@(_filter) forKey:_filteredImage]

            _unfilteredImageSize = unfilteredImageSize;

            [self removeConstraint:_aspectRatioConstraint];

            // maintain the aspect ratio of the image.
            if (_unfilteredImageSize.width > 0 && _unfilteredImageSize
                CGFloat aspectRatio = _unfilteredImageSize.height / _u
                _aspectRatioConstraint = [NSLayoutConstraint constraint
                    attribute:NSLayoutAttributeWidth multiplier:aspect

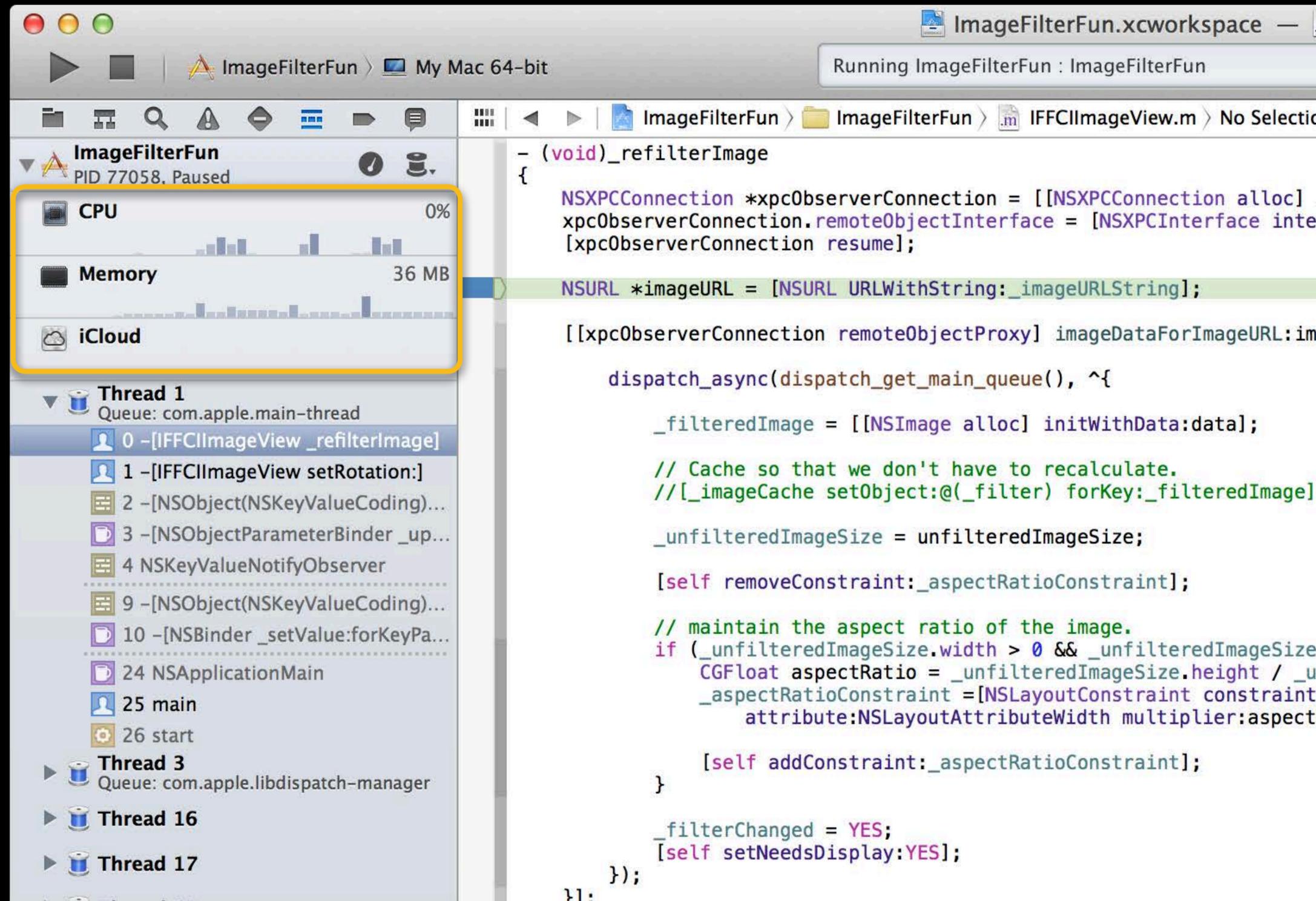
            [self addConstraint:_aspectRatioConstraint];

        }

        _filterChanged = YES;
        [self setNeedsDisplay:YES];

    });
}
```

# Debug Navigator



The screenshot displays the Xcode interface for a project named "ImageFilterFun". The top status bar shows the application is running on a "My Mac 64-bit" device. The left sidebar, titled "ImageFilterFun", shows the process is "Paused" (PID 77058). A yellow box highlights the performance monitoring section, which includes:

- CPU:** 0% usage.
- Memory:** 36 MB usage.
- iCloud:** No usage shown.

Below the performance section, the "Thread 1" list is visible, with the current thread being "0 -[IFFCIIImageView \_refilterImage]". The right pane shows the source code for "IFFCIIImageView.m", with the following code snippet highlighted in green:

```
(void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc] initWithRemoteObjectInterface:[NSXPCInterface interfaceWithName:@"ImageFilterFunImageFilter"]];
    xpcObserverConnection.remoteObjectInterface = [NSXPCInterface interfaceWithName:@"ImageFilterFunImageFilter"];
    [xpcObserverConnection resume];

    NSURL *imageURL = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL];

    dispatch_async(dispatch_get_main_queue(), ^{
        _filteredImage = [[UIImage alloc] initWithData:data];

        // Cache so that we don't have to recalculate.
        //[_imageCache setObject:@(_filter) forKey:_filteredImage];

        _unfilteredImageSize = unfilteredImageSize;

        [self removeConstraint:_aspectRatioConstraint];

        // maintain the aspect ratio of the image.
        if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
            CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
            _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeWidth multiplier:aspectRatio options:0 startingPoint:(CGPoint){0,0} endingPoint:(CGPoint){0,0}];

            [self addConstraint:_aspectRatioConstraint];
        }

        _filterChanged = YES;
        [self setNeedsDisplay:YES];
    });
}
```

# Debug Navigator

The screenshot shows the Xcode interface for a project named "ImageFilterFun". The top status bar indicates the application is running on a "My Mac 64-bit" and is paused. The left sidebar displays performance metrics: CPU at 0%, Memory at 36 MB, and iCloud. Below these, the "Thread 1" section is expanded, showing a call stack for the method `-[IFFCImageView _refilterImage]`. The stack includes various system and application methods, with the current method at the top. The right pane shows the source code for `IFFCImageView.m`, with the line `NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];` highlighted in green. The code is in Objective-C and shows the implementation of the `_refilterImage` method, which uses `NSURLSession` to fetch image data and then processes it.

```
- (void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc] initWithRemoteObjectInterface:[NSXPCInterface interfaceWithName:@"ImageFilterFunImageFilterFun" protocolName:@"ImageFilterFunImageFilterFunProtocol"]];
    xpcObserverConnection.remoteObjectInterface = [NSXPCInterface interfaceWithName:@"ImageFilterFunImageFilterFun" protocolName:@"ImageFilterFunImageFilterFunProtocol"];
    [xpcObserverConnection resume];

    NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageUrl];

    dispatch_async(dispatch_get_main_queue(), ^{
        _filteredImage = [[UIImage alloc] initWithData:data];

        // Cache so that we don't have to recalculate.
        //[_imageCache setObject:@(_filter) forKey:_filteredImage];

        _unfilteredImageSize = unfilteredImageSize;

        [self removeConstraint:_aspectRatioConstraint];

        // maintain the aspect ratio of the image.
        if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
            CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
            _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeWidth multiplier:aspectRatio options:0];

            [self addConstraint:_aspectRatioConstraint];
        }

        _filterChanged = YES;
        [self setNeedsDisplay:YES];
    });
}
```

# Debug Navigator

The screenshot shows the Xcode interface with the Debug Navigator on the left and the Source Editor on the right. The Debug Navigator displays the following information:

- ImageFilterFun** (PID 77058, Paused) - A yellow circle highlights the pause icon.
- CPU**: 0%
- Memory**: 36 MB
- iCloud**
- Thread 1** (Queue: com.apple.main-thread)
  - 0 -[IFFCImageView \_refilterImage] (highlighted)
  - 1 -[IFFCImageView setRotation:]
  - 2 -[NSObject(NSKeyValueCoding)...
  - 3 -[NSObjectParameterBinder \_up...
  - 4 NSKeyValueNotifyObserver
  - 9 -[NSObject(NSKeyValueCoding)...
  - 10 -[NSBinder \_setValue:forKeyPa...
  - 24 NSApplicationMain
  - 25 main
  - 26 start
- Thread 3** (Queue: com.apple.libdispatch-manager)
- Thread 16**
- Thread 17**

The Source Editor on the right shows the following Objective-C code:

```
- (void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc]
xpcObserverConnection.remoteObjectInterface = [NSXPCInterface inte
[xpcObserverConnection resume];

    NSURL *imageURL = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:im

        dispatch_async(dispatch_get_main_queue(), ^{

            _filteredImage = [[UIImage alloc] initWithData:data];

            // Cache so that we don't have to recalculate.
            //[_imageCache setObject:@(_filter) forKey:_filteredImage]

            _unfilteredImageSize = unfilteredImageSize;

            [self removeConstraint:_aspectRatioConstraint];

            // maintain the aspect ratio of the image.
            if (_unfilteredImageSize.width > 0 && _unfilteredImageSize
                CGFloat aspectRatio = _unfilteredImageSize.height / _u
                _aspectRatioConstraint = [NSLayoutConstraint constraint
                    attribute:NSLayoutAttributeWidth multiplier:aspect

            [self addConstraint:_aspectRatioConstraint];
        }

        _filterChanged = YES;
        [self setNeedsDisplay:YES];
    });
};
```

# Debug Navigator

The screenshot displays the Xcode interface for debugging an application named "ImageFilterFun". The top status bar indicates the application is running on a "My Mac 64-bit" device. The left sidebar shows the "ImageFilterFun" process (PID 77058, Paused) with performance metrics for CPU (0%), Memory (36 MB), and iCloud. Below this, the "Thread 1" stack is visible, with the current method `0 -[IFFCImageView _refilterImage]` selected. The right pane shows the source code for `IFFCImageView.m`, with the line `NSURL *imageURL = [NSURL URLWithString:_imageUrlString];` highlighted in green, corresponding to the selected method in the thread stack.

```
- (void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc] initWithXPCServiceName:@"com.apple.ImageFilterFun"];
    xpcObserverConnection.remoteObjectInterface = [NSXPCInterface interfaceWithName:@"com.apple.ImageFilterFun.IFFCImageView"];
    [xpcObserverConnection resume];

    NSURL *imageURL = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL];

    dispatch_async(dispatch_get_main_queue(), ^{
        _filteredImage = [[UIImage alloc] initWithData:data];

        // Cache so that we don't have to recalculate.
        //[_imageCache setObject:@(_filter) forKey:_filteredImage];

        _unfilteredImageSize = unfilteredImageSize;

        [self removeConstraint:_aspectRatioConstraint];

        // maintain the aspect ratio of the image.
        if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
            CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
            _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self
                attribute:NSLayoutAttributeWidth multiplier:aspectRatio
                relationship:NSLayoutRelationEqual toView:nil
                toItem:nil options:0];

            [self addConstraint:_aspectRatioConstraint];
        }

        _filterChanged = YES;
        [self setNeedsDisplay:YES];
    });
}
```

# Debug Navigator

ImageFilterFun.xcworkspace — Running ImageFilterFun : ImageFilterFun

ImageFilterFun > My Mac 64-bit

ImageFilterFun > ImageFilterFun > IFFCImageView.m > No Selection

**ImageFilterFun**  
PID 77058, Paused

**CPU** 0%

**Memory** 36 MB

**iCloud**

**com.apple.main-thread (serial)**  
1 Thread

- Thread 1
  - 0 -[IFFCImageView \_refilterImage]
  - 1 -[IFFCImageView setRotation:]
  - 2 -[NSObject(NSKeyValueCodin...]
  - 3 -[NSObjectParameterBinder \_...]
  - 4 NSKeyValueNotifyObserver
  - 9 -[NSObject(NSKeyValueCodin...]
  - 10 -[NSBinder \_setValue:forKe...]
  - 24 NSApplicationMain
  - 25 main
  - 26 start

**com.apple.libdispatch-manager (s...)**  
1 Thread

- Thread 16

```
- (void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc]
xpcObserverConnection.remoteObjectInterface = [NSXPCInterface inte
[xpcObserverConnection resume];

    NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:im

        dispatch_async(dispatch_get_main_queue(), ^{

            _filteredImage = [[UIImage alloc] initWithData:data];

            // Cache so that we don't have to recalculate.
            //[_imageCache setObject:@(_filter) forKey:_filteredImage]

            _unfilteredImageSize = unfilteredImageSize;

            [self removeConstraint:_aspectRatioConstraint];

            // maintain the aspect ratio of the image.
            if (_unfilteredImageSize.width > 0 && _unfilteredImageSize
                CGFloat aspectRatio = _unfilteredImageSize.height / _u
                _aspectRatioConstraint = [NSLayoutConstraint constraint
                    attribute:NSLayoutAttributeWidth multiplier:aspect

            [self addConstraint:_aspectRatioConstraint];

        }

        _filterChanged = YES;
        [self setNeedsDisplay:YES];

    });
}
```

# Debug Navigator

The screenshot shows the Xcode interface with the Debug Navigator on the left and the Source Editor on the right. The Debug Navigator displays the process 'ImageFilterFun' (PID 77058, Paused) with a yellow circle around the 'Show Call Stack' icon. Below it, the 'com.apple.main-thread (serial)' is expanded to show 'Thread 1' with a list of 27 stack frames. The top frame is '-[IFFCImageView \_refilterImage]'. The Source Editor on the right shows the implementation of this method in 'IFFCImageView.m', with the line `NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];` highlighted in green. The code includes logic for fetching image data via XPC, caching, and maintaining the aspect ratio of the image.

```
ImageFilterFun.xcworkspace —
Running ImageFilterFun : ImageFilterFun

ImageFilterFun > ImageFilterFun > IFFCImageView.m > No Selection

- (void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc]
xpcObserverConnection.remoteObjectInterface = [NSXPCInterface inte
[xpcObserverConnection resume];

    NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:im

        dispatch_async(dispatch_get_main_queue(), ^{

            _filteredImage = [[UIImage alloc] initWithData:data];

            // Cache so that we don't have to recalculate.
            //[_imageCache setObject:@(_filter) forKey:_filteredImage]

            _unfilteredImageSize = unfilteredImageSize;

            [self removeConstraint:_aspectRatioConstraint];

            // maintain the aspect ratio of the image.
            if (_unfilteredImageSize.width > 0 && _unfilteredImageSize
                CGFloat aspectRatio = _unfilteredImageSize.height / _u
                _aspectRatioConstraint = [NSLayoutConstraint constraint
                    attribute:NSLayoutAttributeWidth multiplier:aspect

            [self addConstraint:_aspectRatioConstraint];

        }

        _filterChanged = YES;
        [self setNeedsDisplay:YES];

    });
}
```

# Debug Navigator

The screenshot shows the Xcode interface with the Debug Navigator on the left and the Source Editor on the right. The Debug Navigator displays the process 'ImageFilterFun' (PID 77058, Paused) with CPU usage at 0% and Memory usage at 36 MB. Below this, the thread hierarchy is shown, including 'com.apple.main-thread (serial)' with 1 thread, and 'Thread 1' containing several methods such as '-[IFFCImageView \_refilterImage]', '-[IFFCImageView setRotation:]', and '-[NSObject(NSKeyValueCodin...)'].

The Source Editor on the right shows the implementation of the `-(void)_refilterImage` method in `IFFCImageView.m`. The code is as follows:

```
- (void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc]
xpcObserverConnection.remoteObjectInterface = [NSXPCInterface inte
[xpcObserverConnection resume];

    NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:im

        dispatch_async(dispatch_get_main_queue(), ^{

            _filteredImage = [[UIImage alloc] initWithData:data];

            // Cache so that we don't have to recalculate.
            //[_imageCache setObject:@(_filter) forKey:_filteredImage]

            _unfilteredImageSize = unfilteredImageSize;

            [self removeConstraint:_aspectRatioConstraint];

            // maintain the aspect ratio of the image.
            if (_unfilteredImageSize.width > 0 && _unfilteredImageSize
                CGFloat aspectRatio = _unfilteredImageSize.height / _u
                _aspectRatioConstraint = [NSLayoutConstraint constraint
                    attribute:NSLayoutAttributeWidth multiplier:aspect

            [self addConstraint:_aspectRatioConstraint];

        }

        _filterChanged = YES;
        [self setNeedsDisplay:YES];

    });
};
```

# Debug Navigator

ImageFilterFun.xcworkspace — Running ImageFilterFun : ImageFilterFun

ImageFilterFun > My Mac 64-bit

ImageFilterFun > ImageFilterFun > IFFCImageView.m > No Selection

**ImageFilterFun**  
PID 77058, Paused

CPU 0%

Memory 36 MB

iCloud

**com.apple.main-thread (serial)**  
1 Thread

- Thread 1
  - 0 -[IFFCImageView \_refilterImage]
  - 1 -[IFFCImageView setRotation:]
  - 2 -[NSObject(NSKeyValueCodin...]
  - 3 -[NSObjectParameterBinder \_...]
  - 4 NSKeyValueNotifyObserver
  - 9 -[NSObject(NSKeyValueCodin...]
  - 10 -[NSBinder \_setValue:forKe...]
  - 24 NSApplicationMain
  - 25 main
  - 26 start
- com.apple.libdispatch-manager (s...)  
1 Thread
- Thread 16

```
- (void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc]
xpcObserverConnection.remoteObjectInterface = [NSXPCInterface inte
[xpcObserverConnection resume];

    NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:im

    dispatch_async(dispatch_get_main_queue(), ^{

        _filteredImage = [[NSImage alloc] initWithData:data];

        // Cache so that we don't have to recalculate.
        //[_imageCache setObject:@(_filter) forKey:_filteredImage]

        _unfilteredImageSize = unfilteredImageSize;

        [self removeConstraint:_aspectRatioConstraint];

        // maintain the aspect ratio of the image.
        if (_unfilteredImageSize.width > 0 && _unfilteredImageSize
            CGFloat aspectRatio = _unfilteredImageSize.height / _u
            _aspectRatioConstraint = [NSLayoutConstraint constraint
                attribute:NSLayoutAttributeWidth multiplier:aspect

        [self addConstraint:_aspectRatioConstraint];
    }

    _filterChanged = YES;
    [self setNeedsDisplay:YES];
});
};
```

# Debug Navigator

ImageFilterFun.xcworkspace — Running ImageFilterFun : ImageFilterFun

ImageFilterFun > My Mac 64-bit

ImageFilterFun > ImageFilterFun > IFFCImageView.m > No Selection

**ImageFilterFun**  
PID 77058, Paused

**CPU** 0%

**Memory** 36 MB

**iCloud**

**com.apple.main-thread (serial)**  
1 Thread

- Thread 1
  - 0 -[IFFCImageView \_refilterImage]
  - 1 -[IFFCImageView setRotation:]
  - 2 -[NSObject(NSKeyValueCodin...]
  - 3 -[NSObjectParameterBinder \_...]
  - 4 NSKeyValueNotifyObserver
  - 9 -[NSObject(NSKeyValueCodin...]
  - 10 -[NSBinder \_setValue:forKe...]
  - 24 NSApplicationMain
  - 25 main
  - 26 start

**com.apple.libdispatch-manager (s...)**  
1 Thread

- Thread 16

```
- (void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc]
xpcObserverConnection.remoteObjectInterface = [NSXPCInterface inte
[xpcObserverConnection resume];

    NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:im

    dispatch_async(dispatch_get_main_queue(), ^{

        _filteredImage = [[UIImage alloc] initWithData:data];

        // Cache so that we don't have to recalculate.
        //[_imageCache setObject:@(_filter) forKey:_filteredImage]

        _unfilteredImageSize = unfilteredImageSize;

        [self removeConstraint:_aspectRatioConstraint];

        // maintain the aspect ratio of the image.
        if (_unfilteredImageSize.width > 0 && _unfilteredImageSize
            CGFloat aspectRatio = _unfilteredImageSize.height / _u
            _aspectRatioConstraint = [NSLayoutConstraint constraint
                attribute:NSLayoutAttributeWidth multiplier:aspect

        [self addConstraint:_aspectRatioConstraint];
    }

    _filterChanged = YES;
    [self setNeedsDisplay:YES];
});
};
```

# Variables View

The screenshot displays the Xcode IDE interface with a breakpoint set in the `-(void)_refilterImage` method of `IFFCIImageView.m`. The Variables View is open, showing the state of variables at the breakpoint.

**Variables View:**

- `self` = (IFFCIImageView \*) 0x10052b380
- `imageURL` = (NSURL \*) 0x7fff5fbfe6c0
- `xpcObserverConnection` = (NSXPCConnection \*) 0x100540410
- `imageURLString` = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

**lldb Output:**

```
(lldb) po self
<IFFCIImageView: 0x10052b380>
(lldb) p _imageURLString
(NSString *) $8 = 0x0000000100519f10 @"http://images.apple.com/ipad/overview/
images/hero.png"
(lldb) |
```

**Code Snippet:**

```
-(void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc] initWithServiceName:@"com.apple.dt.Xcode.FilterIt"];
    xpcObserverConnection.remoteObjectInterface = [NSXPCInterface interfaceWithProtocol:@protocol(IFFFilterService)];
    [xpcObserverConnection resume];

    NSURL *imageURL = [NSURL URLWithString:_imageURLString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, NSSize unfilteredImageSize) {
        dispatch_async(dispatch_get_main_queue(), ^{
            _filteredImage = [[UIImage alloc] initWithData:data];

            // Cache so that we don't have to recalculate.
            //[_imageCache setObject:@(_filter) forKey:_filteredImage];

            _unfilteredImageSize = unfilteredImageSize;

            [self removeConstraint:_aspectRatioConstraint];

            // maintain the aspect ratio of the image.
            if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
                CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
                _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutRelationEqual toItem:self
                    attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

                [self addConstraint:_aspectRatioConstraint];
            }

            _filterChanged = YES;
            [self setNeedsDisplay:YES];
        });
    }];
}
```

- 1 -[IFFCIImageView setRotation:]
- 2 -[NSObject(NSKeyValueCoding)...
- 3 -[NSObjectParameterBinder \_up...
- 4 NSKeyValueNotifyObserver
- 9 -[NSObject(NSKeyValueCoding)...
- 10 -[NSBinder \_setValue:forKeyPa...
- 24 NSApplicationMain
- 25 main
- 26 start
- ▶ Thread 3  
Queue: com.apple.libdispatch-manager
- ▶ Thread 16
- ▶ Thread 17
- ▶ Thread 18

```
// Cache so that we don't have to recalculate.  
//[_imageCache setObject:@(_filter) forKey:_filteredImage];  
  
_unfilteredImageSize = unfilteredImageSize;  
  
[self removeConstraint:_aspectRatioConstraint];  
  
// maintain the aspect ratio of the image.  
if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {  
    CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;  
    _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];  
  
    [self addConstraint:_aspectRatioConstraint];  
}  
  
_filterChanged = YES;  
[self setNeedsDisplay:YES];  
});  
});
```

ImageFilterFun > Thread 1 > 0 -[IFFCIImageView \_refilterImage]

- ▶ **A** self = (IFFCIImageView \*) 0x10052b380
- ▶ **L** imageURL = (NSURL \*) 0x7fff5fbfe6c0
- ▶ **L** xpcObserverConnection = (NSXPCConnection \*) 0x100540410
- ▶ **V** \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

```
(lldb) po self  
<IFFCIImageView: 0x10052b380>  
(lldb) p _imageURLString  
(NSString *) $8 = 0x0000000100519f10 @"http://images.ap  
images/hero.png"  
(lldb) |
```



Auto



All Output

- 1 -[IFFCImageView setRotation:]
- 2 -[NSObject(NSKeyValueCoding)...
- 3 -[NSObjectParameterBinder \_up...
- 4 NSKeyValueNotifyObserver
- 9 -[NSObject(NSKeyValueCoding)...
- 10 -[NSBinder \_setValue:forKeyPa...
- 24 NSApplicationMain
- 25 main
- 26 start
- Thread 3  
Queue: com.apple.libdispatch-manager
- Thread 16
- Thread 17
- Thread 18

```
// Cache so that we don't have to recalculate.  
//[_imageCache setObject:@(_filter) forKey:_filteredImage];  
  
_unfilteredImageSize = unfilteredImageSize;  
  
[self removeConstraint:_aspectRatioConstraint];  
  
// maintain the aspect ratio of the image.  
if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {  
    CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;  
    _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];  
    [self addConstraint:_aspectRatioConstraint];  
}  
  
_filterChanged = YES;  
[self setNeedsDisplay:YES];  
});  
});
```

ImageFilterFun > Thread 1 > 0 -[IFFCImageView \_refilterImage]

- self = (IFFCImageView \*) 0x10052b380
- imageURL = (NSURL \*) 0x7fff5fbfe6c0
- xpcObserverConnection = (NSXPCConnection \*) 0x100540410
- \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

```
(lldb) po self  
<IFFCImageView: 0x10052b380>  
(lldb) p _imageURLString  
(NSString *) $8 = 0x0000000100519f10 @"http://images.ap  
images/hero.png"  
(lldb) |
```

Auto

All Output

- 1 -[IFFCIImageView setRotation:]
- 2 -[NSObject(NSKeyValueCoding)...
- 3 -[NSObjectParameterBinder \_up...
- 4 NSKeyValueNotifyObserver
- 9 -[NSObject(NSKeyValueCoding)...
- 10 -[NSBinder \_setValue:forKeyPa...
- 24 NSApplicationMain
- 25 main
- 26 start
- ▶ Thread 3  
Queue: com.apple.libdispatch-manager
- ▶ Thread 16
- ▶ Thread 17
- ▶ Thread 18

```
// Cache so that we don't have to recalculate.  
//[_imageCache setObject:@(_filter) forKey:_filteredImage];  
  
_unfilteredImageSize = unfilteredImageSize;  
  
[self removeConstraint:_aspectRatioConstraint];  
  
// maintain the aspect ratio of the image.  
if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {  
    CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;  
    _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];  
  
    [self addConstraint:_aspectRatioConstraint];  
}  
  
_filterChanged = YES;  
[self setNeedsDisplay:YES];  
});  
});
```

ImageFilterFun > Thread 1 > 0 -[IFFCIImageView \_refilterImage]

- ▶ **A** self = (IFFCIImageView \*) 0x10052b380
- ▶ **L** imageURL = (NSURL \*) 0x7fff5fbfe6c0
- ▶ **L** xpcObserverConnection = (NSXPCConnection \*) 0x100540410
- ▶ **V** \_imageURLString = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

```
(lldb) po self  
<IFFCIImageView: 0x10052b380>  
(lldb) p _imageURLString  
(NSString *) $8 = 0x0000000100519f10 @"http://images.ap  
images/hero.png"  
(lldb) |
```



Auto



All Output

- 1 -[IFFCIImageView setRotation:]
- 2 -[NSObject(NSKeyValueCoding)...
- 3 -[NSObjectParameterBinder \_up...
- 4 NSKeyValueNotifyObserver
- 9 -[NSObject(NSKeyValueCoding)...
- 10 -[NSBinder \_setValue:forKeyPa...
- 24 NSApplicationMain
- 25 main
- 26 start
- ▶ Thread 3  
Queue: com.apple.libdispatch-manager
- ▶ Thread 16
- ▶ Thread 17
- ▶ Thread 18

```
// Cache so that we don't have to recalculate.  
//[_imageCache setObject:@(_filter) forKey:_filteredImage];  
  
_unfilteredImageSize = unfilteredImageSize;  
  
[self removeConstraint:_aspectRatioConstraint];  
  
// maintain the aspect ratio of the image.  
if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {  
    CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;  
    _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutRelationEqual toItem:self attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];  
  
    [self addConstraint:_aspectRatioConstraint];  
}  
  
_filterChanged = YES;  
[self setNeedsDisplay:YES];  
});  
});
```

ImageFilterFun > Thread 1 > 0 -[IFFCIImageView \_refilterImage]

- ▶ self = (IFFCIImageView \*) 0x10052b380
- ▶ imageURL = (NSURL \*) 0x7fff5fbfe6c0
- ▶ xpcObserverConnection = (NSXPCConnection \*) 0x100540410
- ▶ **imageURLString** = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

```
(lldb) po self  
<IFFCIImageView: 0x10052b380>  
(lldb) p _imageURLString  
(NSString *) $8 = 0x0000000100519f10 @"http://images.ap  
images/hero.png"  
(lldb) |
```

Auto

All Output

- 1 -[IFFCIImageView setRotation:]
- 2 -[NSObject(NSKeyValueCoding)...
- 3 -[NSObjectParameterBinder \_up...
- 4 NSKeyValueNotifyObserver
- 9 -[NSObject(NSKeyValueCoding)...
- 10 -[NSBinder \_setValue:forKeyPa...
- 24 NSApplicationMain
- 25 main
- 26 start
- ▶ Thread 3  
Queue: com.apple.libdispatch-manager
- ▶ Thread 16
- ▶ Thread 17
- ▶ Thread 18

```

// Cache so that we don't have to recalculate.
//[_imageCache setObject:@(_filter) forKey:_filteredImage];

_unfilteredImageSize = unfilteredImageSize;

[self removeConstraint:_aspectRatioConstraint];

// maintain the aspect ratio of the image.
if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
    CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
    _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

    [self addConstraint:_aspectRatioConstraint];
}

_filterChanged = YES;
[self setNeedsDisplay:YES];
});
});

```

ImageFilterFun > Thread 1 > 0 -[IFFCIImageView \_refilterImage]

- ▶ **A** self = (IFFCIImageView \*) 0x10052b380
- ▶ **A** \_cmd = (SEL) "\_refilterImage"
- ▶ **L** xpcObserverConnection = (NSXPCConnection \*) 0x100540410
- ▶ **L** imageURL = (NSURL \*) 0x7fff5fbfe6c0
- ▶ **R** Exception State Registers
- ▶ **R** Floating Point Registers
- ▶ **R** General Purpose Registers

```

(lldb) po self
<IFFCIImageView: 0x10052b380>
(lldb) p _imageURLString
(NSString *) $8 = 0x0000000100519f10 @"http://images.ap
images/hero.png"
(lldb) |

```



All ▾ |



All Output ▾

- 1 -[IFFCImageView setRotation:]
- 2 -[NSObject(NSKeyValueCoding)...
- 3 -[NSObjectParameterBinder \_up...
- 4 NSKeyValueNotifyObserver
- 9 -[NSObject(NSKeyValueCoding)...
- 10 -[NSBinder \_setValue:forKeyPa...
- 24 NSApplicationMain
- 25 main
- 26 start
- Thread 3  
Queue: com.apple.libdispatch-manager
- Thread 16
- Thread 17
- Thread 18

```
// Cache so that we don't have to recalculate.
//[_imageCache setObject:@(_filter) forKey:_filteredImage];

_unfilteredImageSize = unfilteredImageSize;

[self removeConstraint:_aspectRatioConstraint];

// maintain the aspect ratio of the image.
if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
    CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
    _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

    [self addConstraint:_aspectRatioConstraint];
}

_filterChanged = YES;
[self setNeedsDisplay:YES];
});
});
```

ImageFilterFun > Thread 1 > 0 -[IFFCImageView \_refilterImage]

- self = (IFFCImageView \*) 0x10052b380
- \_cmd = (SEL) "\_refilterImage"
- xpcObserverConnection = (NSXPCConnection \*) 0x100540410
- imageURL = (NSURL \*) 0x7fff5fbfe6c0
- Exception State Registers
- Floating Point Registers
- General Purpose Registers

```
(lldb) po self
<IFFCImageView: 0x10052b380>
(lldb) p _imageURLString
(NSString *) $8 = 0x0000000100519f10 @"http://images.ap
images/hero.png"
(lldb) |
```



All

All Output

# Debug Console

The screenshot shows the Xcode IDE with a breakpoint set in the `-(void)_refilterImage` method of `IFFCIImageView.m`. The breakpoint is triggered at the line `NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];`. The left sidebar shows the project structure and a list of threads, with Thread 1 selected. The bottom pane displays the current state of variables and registers.

```
-(void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc] initWithServiceName:@"com.apple.dt.Xcode.FilterIt"];
    xpcObserverConnection.remoteObjectInterface = [NSXPCInterface interfaceWithProtocol:@protocol(IFFFilterService)];
    [xpcObserverConnection resume];

    NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, NSUInteger unfilteredImageSize) {

        dispatch_async(dispatch_get_main_queue(), ^{

            _filteredImage = [[UIImage alloc] initWithData:data];

            // Cache so that we don't have to recalculate.
            //[_imageCache setObject:@(_filter) forKey:_filteredImage];

            _unfilteredImageSize = unfilteredImageSize;

            [self removeConstraint:_aspectRatioConstraint];

            // maintain the aspect ratio of the image.
            if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
                CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
                _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutRelationEqual toItem:self
                    attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

                [self addConstraint:_aspectRatioConstraint];
            }

            _filterChanged = YES;
            [self setNeedsDisplay:YES];
        });
    }];
}
```

Thread 1: breakpoint 1.1

self = (IFFCIImageView \*) 0x10052b380  
\_cmd = (SEL) "\_refilterImage"  
xpcObserverConnection = (NSXPCConnection \*) 0x100540410  
imageUrl = (NSURL \*) 0x7fff5fbfe6c0

(lldb) po self  
<IFFCIImageView: 0x10052b380>  
(lldb) p \_imageUrlString  
(NSString \*) \$8 = 0x0000000100519f10 @"http://images.apple.com/ipad/overview/images/hero.png"  
(lldb) |

```
_filteredImage = [[UIImage alloc] initWithData:data];  
  
// Cache so that we don't have to recalculate.  
//[_imageCache setObject:@(_filter) forKey:_filteredImage];  
  
_unfilteredImageSize = unfilteredImageSize;  
  
[self removeConstraint:_aspectRatioConstraint];  
  
// maintain the aspect ratio of the image.  
if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {  
    CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;  
    _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutRelationEqual toItem:self  
        attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];  
  
    [self addConstraint:_aspectRatioConstraint];  
}  
  
_filterChanged = YES;  
[self setNeedsDisplay:YES];
```

ImageFilterFun > Thread 1 > 0 -[IFFCIIImageView \_refilterImage]

ImageView \*) 0x10052b380

"\_refilterImage"

Connection = (NSXPConnection \*) 0x100540410

(NSURL \*) 0x7fff5fbfe6c0

ate Registers

at Registers

ose Registers

(lldb) po self

<IFFCIIImageView: 0x10052b380>

(lldb) p \_imageURLString

(NSString \*) \$8 = 0x0000000100519f10 @"http://images.apple.com/ipad/overview/  
images/hero.png"

(lldb) |

All Output ⌵



```
_filteredImage = [[UIImage alloc] initWithData:data];  
  
// Cache so that we don't have to recalculate.  
//[_imageCache setObject:@(_filter) forKey:_filteredImage];  
  
_unfilteredImageSize = unfilteredImageSize;  
  
[self removeConstraint:_aspectRatioConstraint];  
  
// maintain the aspect ratio of the image.  
if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {  
    CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;  
    _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutRelationEqual toItem:self  
        attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];  
  
    [self addConstraint:_aspectRatioConstraint];  
}  
  
_filterChanged = YES;  
[self setNeedsDisplay:YES];
```

ImageFilterFun > Thread 1 > 0 -[IFFCIIImageView \_refilterImage]

ImageView \*) 0x10052b380

"\_refilterImage"

Connection = (NSXPConnection \*) 0x100540410

(NSURL \*) 0x7fff5fbfe6c0

ate Registers

at Registers

ose Registers

(lldb) po self

<IFFCIIImageView: 0x10052b380>

(lldb) p \_imageURLString

(NSString \*) \$8 = 0x0000000100519f10 @"http://images.apple.com/ipad/overview/  
images/hero.png"

(lldb) |

All Output ↕



```
_filteredImage = [[UIImage alloc] initWithData:data];  
  
// Cache so that we don't have to recalculate.  
//[_imageCache setObject:@(_filter) forKey:_filteredImage];  
  
_unfilteredImageSize = unfilteredImageSize;  
  
[self removeConstraint:_aspectRatioConstraint];  
  
// maintain the aspect ratio of the image.  
if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {  
    CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;  
    _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutRelationEqual toItem:self  
        attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];  
  
    [self addConstraint:_aspectRatioConstraint];  
}  
  
_filterChanged = YES;  
[self setNeedsDisplay:YES];
```

ImageFilterFun > Thread 1 > 0 -[IFFCIIImageView \_refilterImage]

ImageView \*) 0x10052b380

"\_refilterImage"

Connection = (NSXPConnection \*) 0x100540410

(NSURL \*) 0x7fff5fbfe6c0

ate Registers

at Registers

ose Registers

(lldb) po self

<IFFCIIImageView: 0x10052b380>

(lldb) p \_imageURLString

(NSString \*) \$8 = 0x0000000100519f10 @"http://images.apple.com/ipad/overview/  
images/hero.png"

(lldb) |

All Output ⌵



# Debug Console

The screenshot shows the Xcode IDE with a breakpoint set in the `-(void)_refilterImage` method of `IFFCIImageView.m`. The breakpoint is triggered, and the console displays the following information:

```
self = (IFFCIImageView *) 0x10052b380
_cmd = (SEL) "_refilterImage"
xpcObserverConnection = (NSXPCConnection *) 0x100540410
imageUrl = (NSURL *) 0x7fff5fbfe6c0
```

The console also shows the state of the registers:

```
(lldb) po self
<IFFCIImageView: 0x10052b380>
(lldb) p _imageUrlString
(NSString *) $8 = 0x0000000100519f10 @"http://images.apple.com/ipad/overview/
images/hero.png"
(lldb) |
```

The code in the editor shows the following snippet:

```
-(void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc] initWithServiceName:@"com.apple.dt.Xcode.FilterIt"];
    xpcObserverConnection.remoteObjectInterface = [NSXPCInterface interfaceWithProtocol:@protocol(IFFFilterService)];
    [xpcObserverConnection resume];

    NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, CGSize unfilteredImageSize) {

        dispatch_async(dispatch_get_main_queue(), ^{

            _filteredImage = [[UIImage alloc] initWithData:data];

            // Cache so that we don't have to recalculate.
            //[_imageCache setObject:@(_filter) forKey:_filteredImage];

            _unfilteredImageSize = unfilteredImageSize;

            [self removeConstraint:_aspectRatioConstraint];

            // maintain the aspect ratio of the image.
            if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
                CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
                _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutRelationEqual toItem:self
                attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

                [self addConstraint:_aspectRatioConstraint];
            }

            _filterChanged = YES;
            [self setNeedsDisplay:YES];
        });
    }];
}
```

# Xcode 5 Makes It Easier to Fix Bugs

- Debug gauges
- Debugging workflow
- XPC services

# Understanding Performance

“We should forget about small efficiencies, say about 97% of the time: **premature optimization** is the root of all evil.”

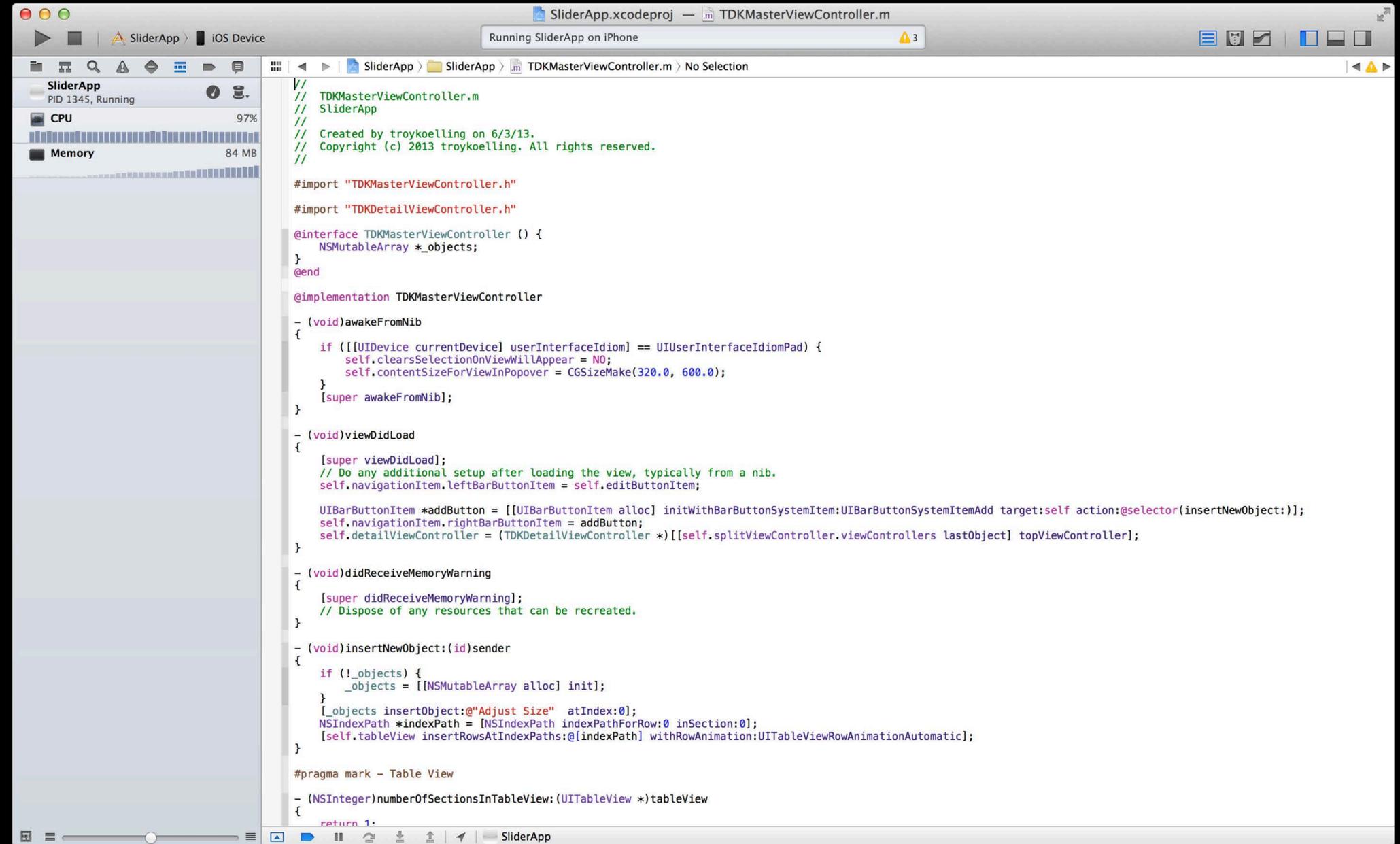
Donald Knuth



**Optimize when you  
measure a problem**

# Debug Gauges

- CPU
- Memory
- iCloud
- Energy
- OpenGL ES



The screenshot shows the Xcode IDE with a debug gauge on the left and a code editor on the right. The debug gauge displays the following information:

Category	Value
CPU	97%
Memory	84 MB

The code editor shows the following code for `TDKMasterViewController.m`:

```
//
// TDKMasterViewController.m
// SliderApp
//
// Created by troykoelling on 6/3/13.
// Copyright (c) 2013 troykoelling. All rights reserved.
//

#import "TDKMasterViewController.h"
#import "TDKDetailViewController.h"

@interface TDKMasterViewController () {
    NSMutableArray *_objects;
}
@end

@implementation TDKMasterViewController

- (void)awakeFromNib
{
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) {
        self.clearsSelectionOnViewWillAppear = NO;
        self.contentSizeForViewInPopover = CGSizeMake(320.0, 600.0);
    }
    [super awakeFromNib];
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    self.navigationItem.leftBarButtonItem = self.editButtonItem;

    UIBarButtonItem *addButton = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemAdd target:self action:@selector(insertNewObject:)];
    self.navigationItem.rightBarButtonItem = addButton;
    self.detailViewController = (TDKDetailViewController *)[self.splitViewController.viewControllers lastObject] topViewController;
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

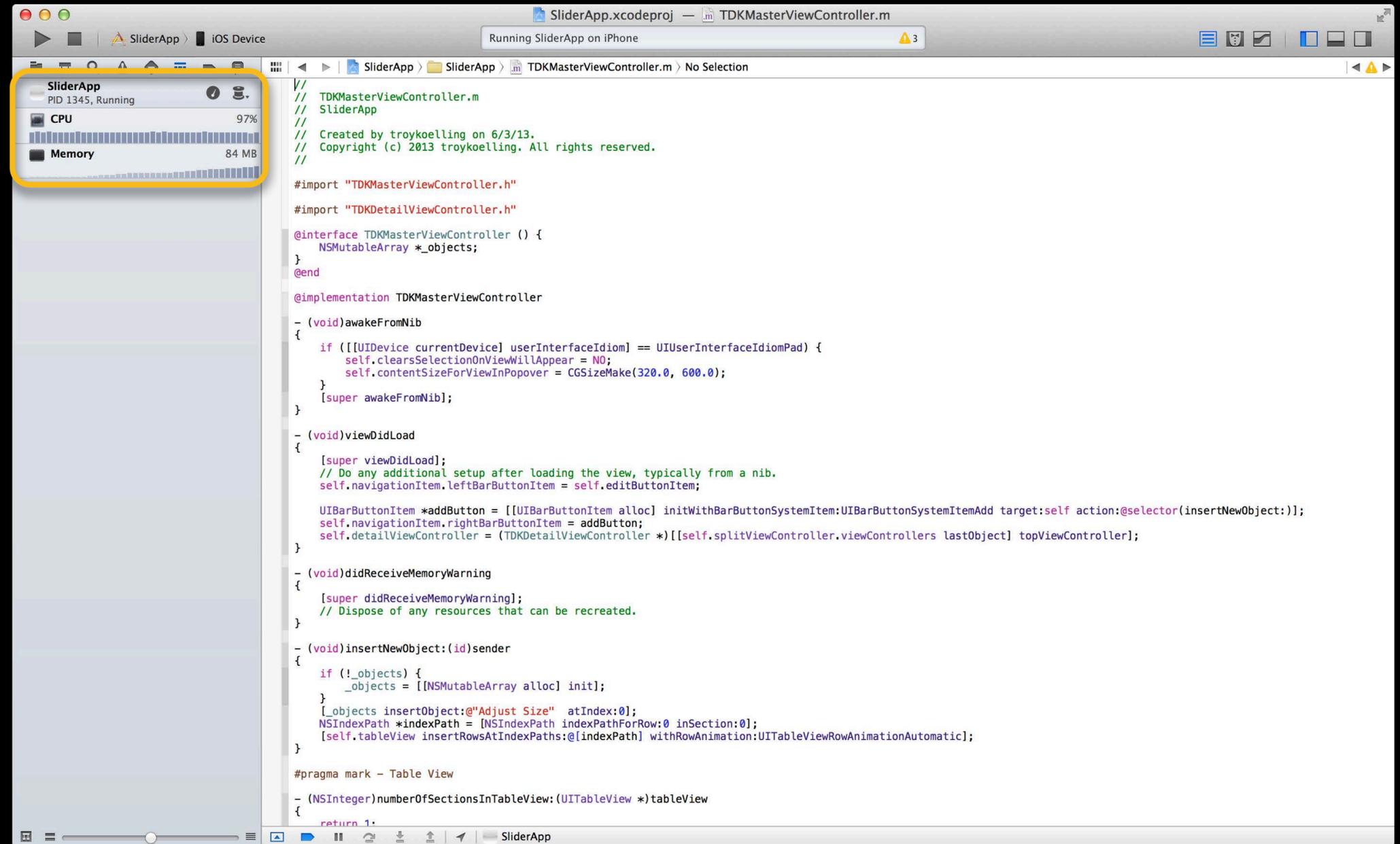
- (void)insertNewObject:(id)sender
{
    if (!_objects) {
        _objects = [[NSMutableArray alloc] init];
    }
    [_objects insertObject:@"Adjust Size" atIndex:0];
    NSIndexPath *indexPath = [NSIndexPath indexPathForRow:0 inSection:0];
    [self.tableView insertRowsAtIndexPaths:@[indexPath] withRowAnimation:UITableViewRowAnimationAutomatic];
}

#pragma mark - Table View

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}
```

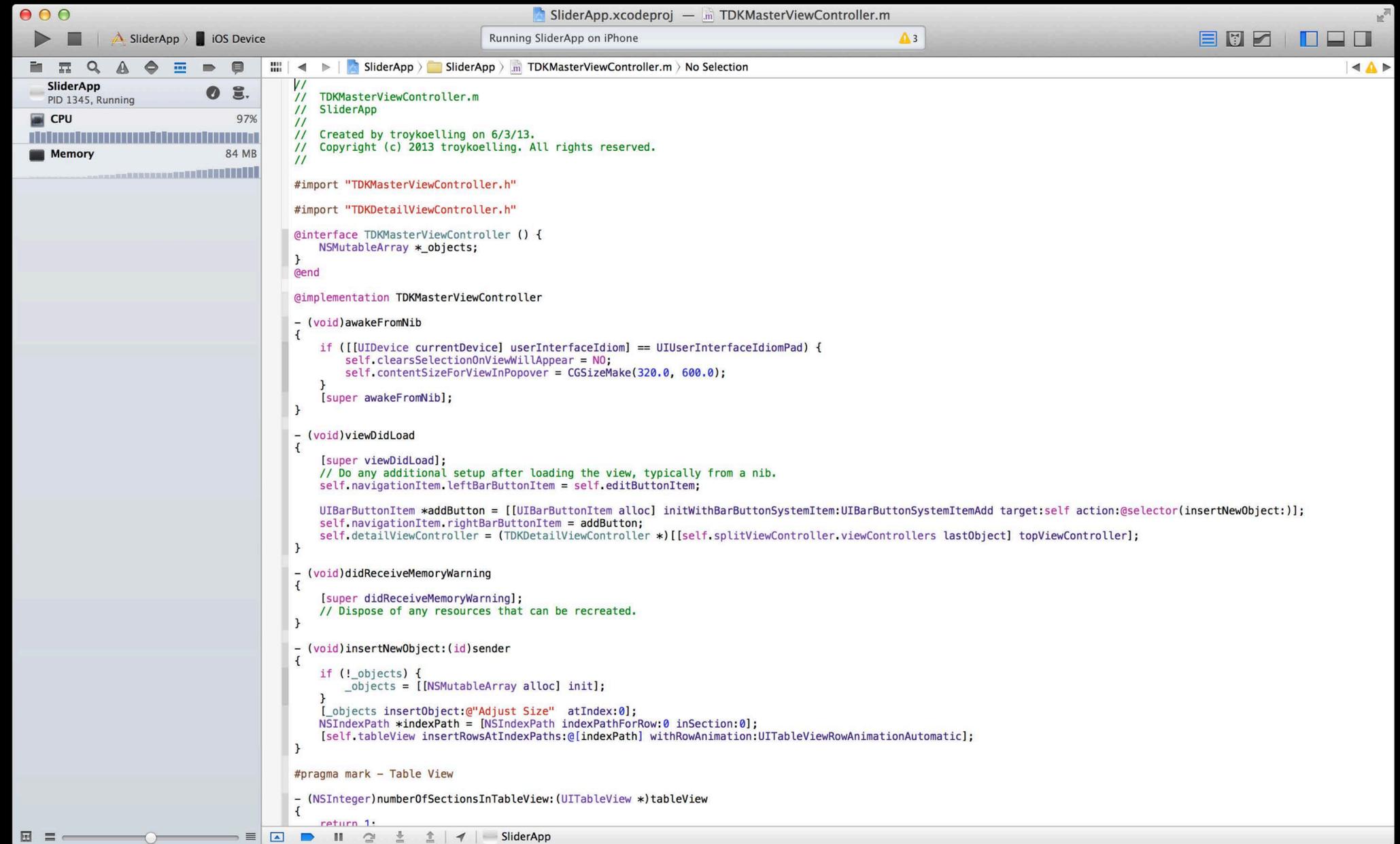
# Debug Gauges

- CPU
- Memory
- iCloud
- Energy
- OpenGL ES



# Debug Gauges

- CPU
- Memory
- iCloud
- Energy
- OpenGL ES



The screenshot shows the Xcode IDE with a debug gauge on the left and a code editor on the right. The debug gauge displays the following information:

Category	Value
CPU	97%
Memory	84 MB

The code editor shows the following code for `TDKMasterViewController.m`:

```
//
// TDKMasterViewController.m
// SliderApp
//
// Created by troykoelling on 6/3/13.
// Copyright (c) 2013 troykoelling. All rights reserved.
//

#import "TDKMasterViewController.h"
#import "TDKDetailViewController.h"

@interface TDKMasterViewController () {
    NSMutableArray *_objects;
}
@end

@implementation TDKMasterViewController

- (void)awakeFromNib
{
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) {
        self.clearsSelectionOnViewWillAppear = NO;
        self.contentSizeForViewInPopover = CGSizeMake(320.0, 600.0);
    }
    [super awakeFromNib];
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    self.navigationItem.rightBarButtonItem = self.editButtonItem;

    UIBarButtonItem *addButton = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemAdd target:self action:@selector(insertNewObject:)];
    self.navigationItem.rightBarButtonItem = addButton;
    self.detailViewController = (TDKDetailViewController *)[self.splitViewController.viewControllers lastObject] topViewController;
}

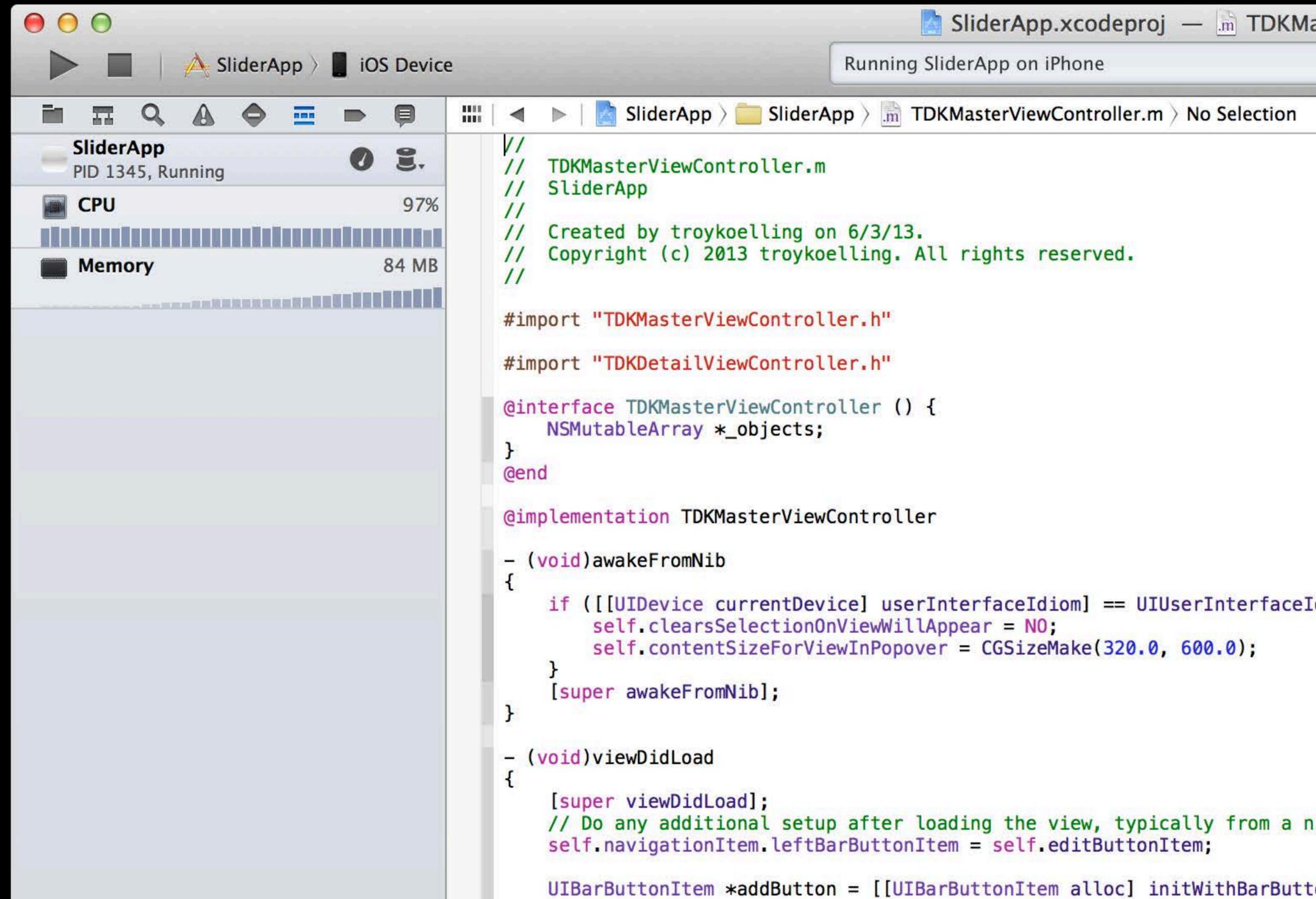
- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (void)insertNewObject:(id)sender
{
    if (!_objects) {
        _objects = [[NSMutableArray alloc] init];
    }
    [_objects insertObject:@"Adjust Size" atIndex:0];
    NSIndexPath *indexPath = [NSIndexPath indexPathForRow:0 inSection:0];
    [self.tableView insertRowsAtIndexPaths:@[indexPath] withRowAnimation:UITableViewRowAnimationAutomatic];
}

#pragma mark - Table View

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}
```

# Understanding Performance



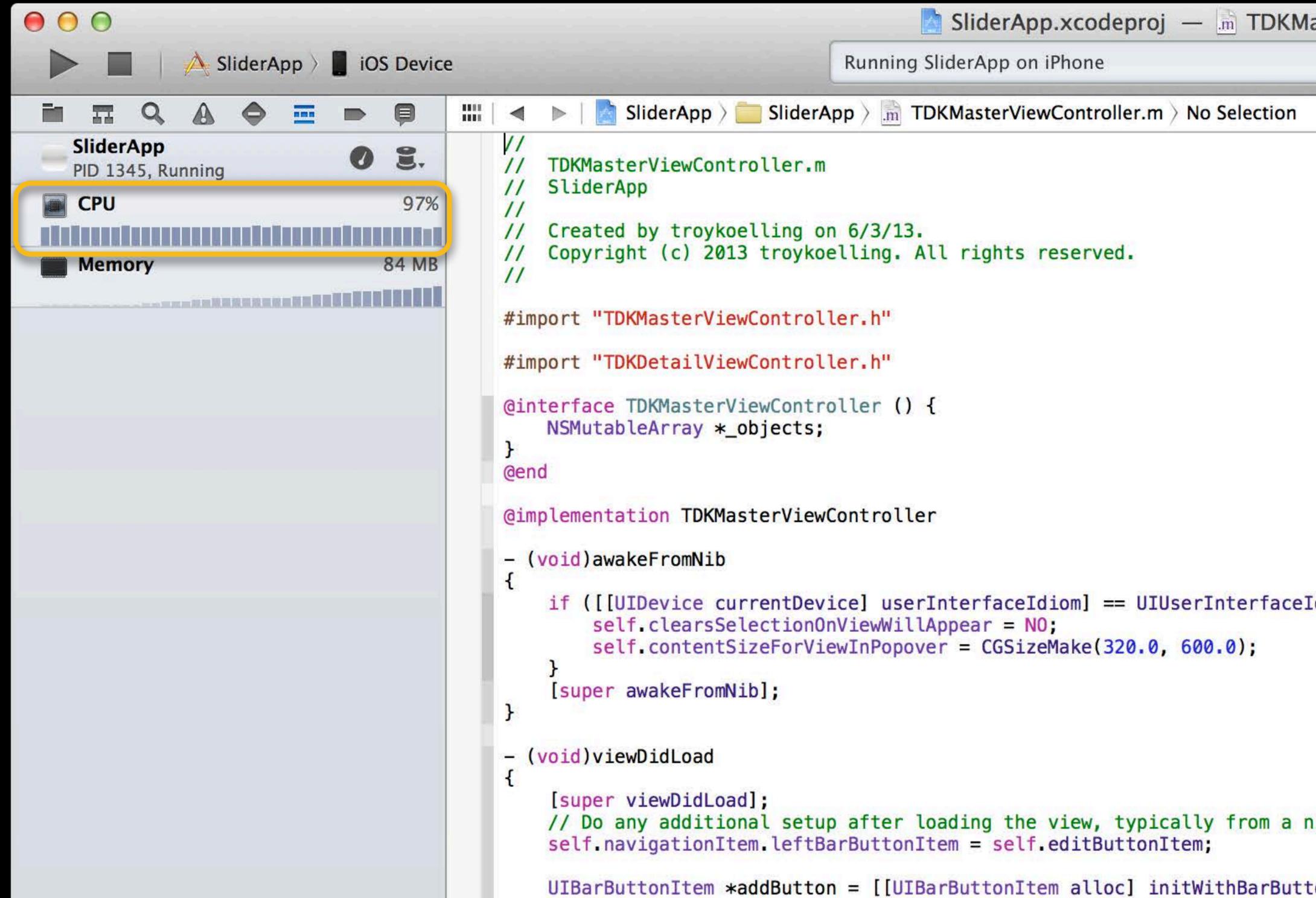
The screenshot displays the Xcode interface for a project named "SliderApp" running on an iPhone. The top status bar indicates "Running SliderApp on iPhone". The left sidebar shows the project structure and performance metrics for the running application:

- SliderApp**: PID 1345, Running
- CPU**: 97%
- Memory**: 84 MB

The main editor area shows the source code for `TDKMasterViewController.m`. The code includes comments and implementation details:

```
//  
// TDKMasterViewController.m  
// SliderApp  
//  
// Created by troykoelling on 6/3/13.  
// Copyright (c) 2013 troykoelling. All rights reserved.  
//  
  
#import "TDKMasterViewController.h"  
  
#import "TDKDetailViewController.h"  
  
@interface TDKMasterViewController () {  
    NSMutableArray *_objects;  
}  
@end  
  
@implementation TDKMasterViewController  
  
- (void)awakeFromNib  
{  
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPhone)  
        self.clearsSelectionOnViewWillAppear = NO;  
        self.contentSizeForViewInPopover = CGSizeMake(320.0, 600.0);  
    }  
    [super awakeFromNib];  
}  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib  
    self.navigationItem.leftBarButtonItem = self.editButtonItem;  
  
    UIBarButtonItem *addButton = [[UIBarButtonItem alloc] initWithBarButton
```

# CPU Report

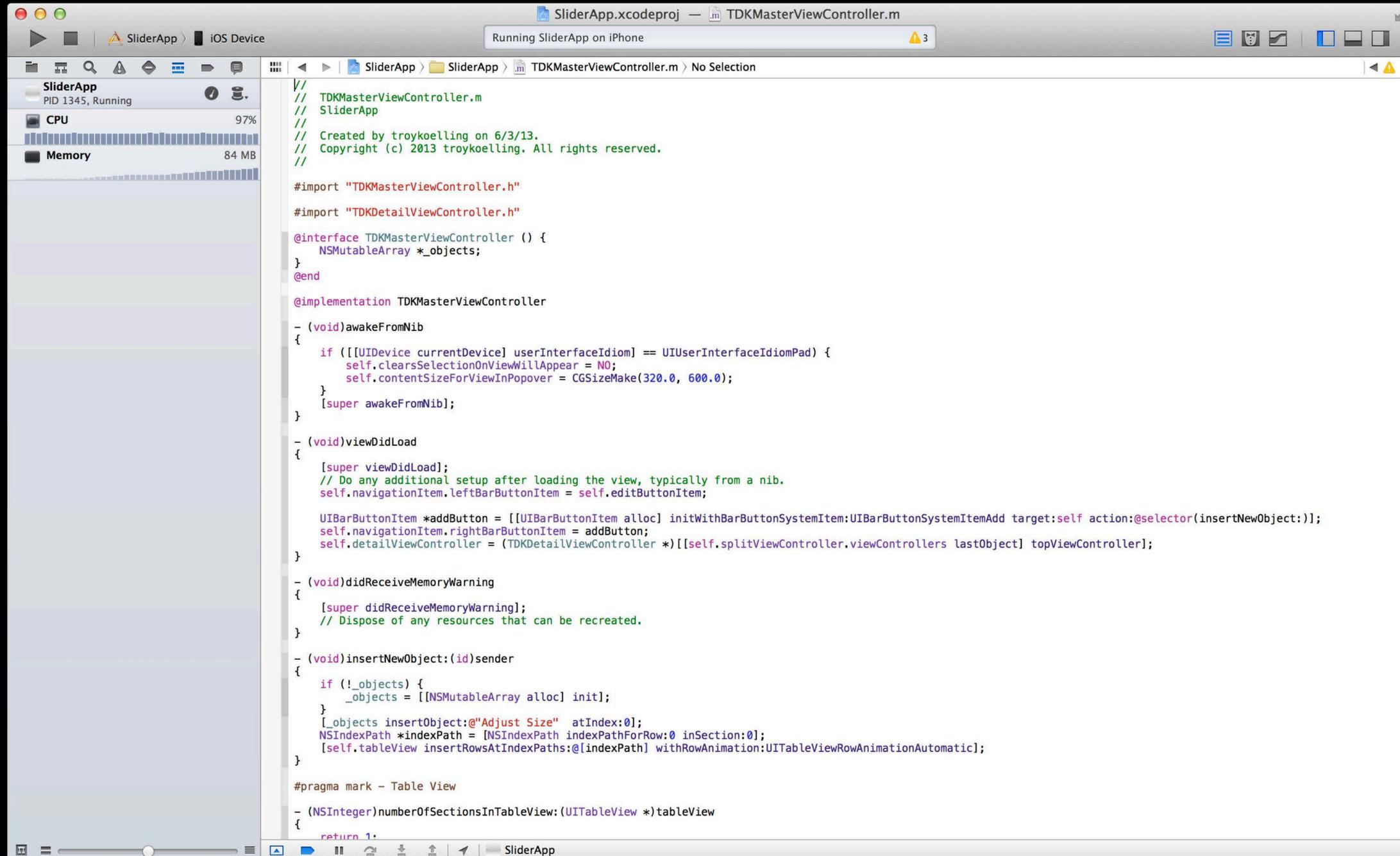


The screenshot shows the Xcode interface for a project named "SliderApp" running on an iPhone. The interface is divided into three main sections:

- Top Bar:** Shows the project name "SliderApp" and the target device "iOS Device". A status bar on the right indicates "Running SliderApp on iPhone".
- Left Panel (Organizer):** Displays the application's resource usage. The "CPU" section is highlighted with a yellow box, showing a usage of 97%. The "Memory" section shows a usage of 84 MB.
- Right Panel (Code Editor):** Shows the source code for "TDKMasterViewController.m". The code includes comments and implementation details for the class.

```
//  
// TDKMasterViewController.m  
// SliderApp  
//  
// Created by troykoelling on 6/3/13.  
// Copyright (c) 2013 troykoelling. All rights reserved.  
//  
  
#import "TDKMasterViewController.h"  
  
#import "TDKDetailViewController.h"  
  
@interface TDKMasterViewController () {  
    NSMutableArray *_objects;  
}  
@end  
  
@implementation TDKMasterViewController  
  
- (void)awakeFromNib  
{  
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPhone)  
        self.clearsSelectionOnViewWillAppear = NO;  
        self.contentSizeForViewInPopover = CGSizeMake(320.0, 600.0);  
    }  
    [super awakeFromNib];  
}  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib  
    self.navigationItem.leftBarButtonItem = self.editButtonItem;  
  
    UIBarButtonItem *addButton = [[UIBarButtonItem alloc] initWithBarButtonImageItem:  
    self.navigationItem.rightBarButtonItem = addButton;  
}
```

# CPU Report



The screenshot shows the Xcode IDE with the following components:

- Top Bar:** SliderApp.xcodeproj — TDKMasterViewController.m. Running SliderApp on iPhone.
- Left Panel:** SliderApp (PID 1345, Running). CPU usage: 97%. Memory usage: 84 MB.
- Code Editor:** TDKMasterViewController.m. No Selection.

```
//
// TDKMasterViewController.m
// SliderApp
//
// Created by troykoelling on 6/3/13.
// Copyright (c) 2013 troykoelling. All rights reserved.
//

#import "TDKMasterViewController.h"
#import "TDKDetailViewController.h"

@interface TDKMasterViewController () {
    NSMutableArray *_objects;
}
@end

@implementation TDKMasterViewController

- (void)awakeFromNib
{
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) {
        self.clearsSelectionOnViewWillAppear = NO;
        self.contentSizeForViewInPopover = CGSizeMake(320.0, 600.0);
    }
    [super awakeFromNib];
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    self.navigationItem.leftBarButtonItem = self.editButtonItem;

    UIBarButtonItem *addButton = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemAdd target:self action:@selector(insertNewObject:)];
    self.navigationItem.rightBarButtonItem = addButton;
    self.detailViewController = (TDKDetailViewController *)[self.splitViewController.viewControllers lastObject] topViewController;
}

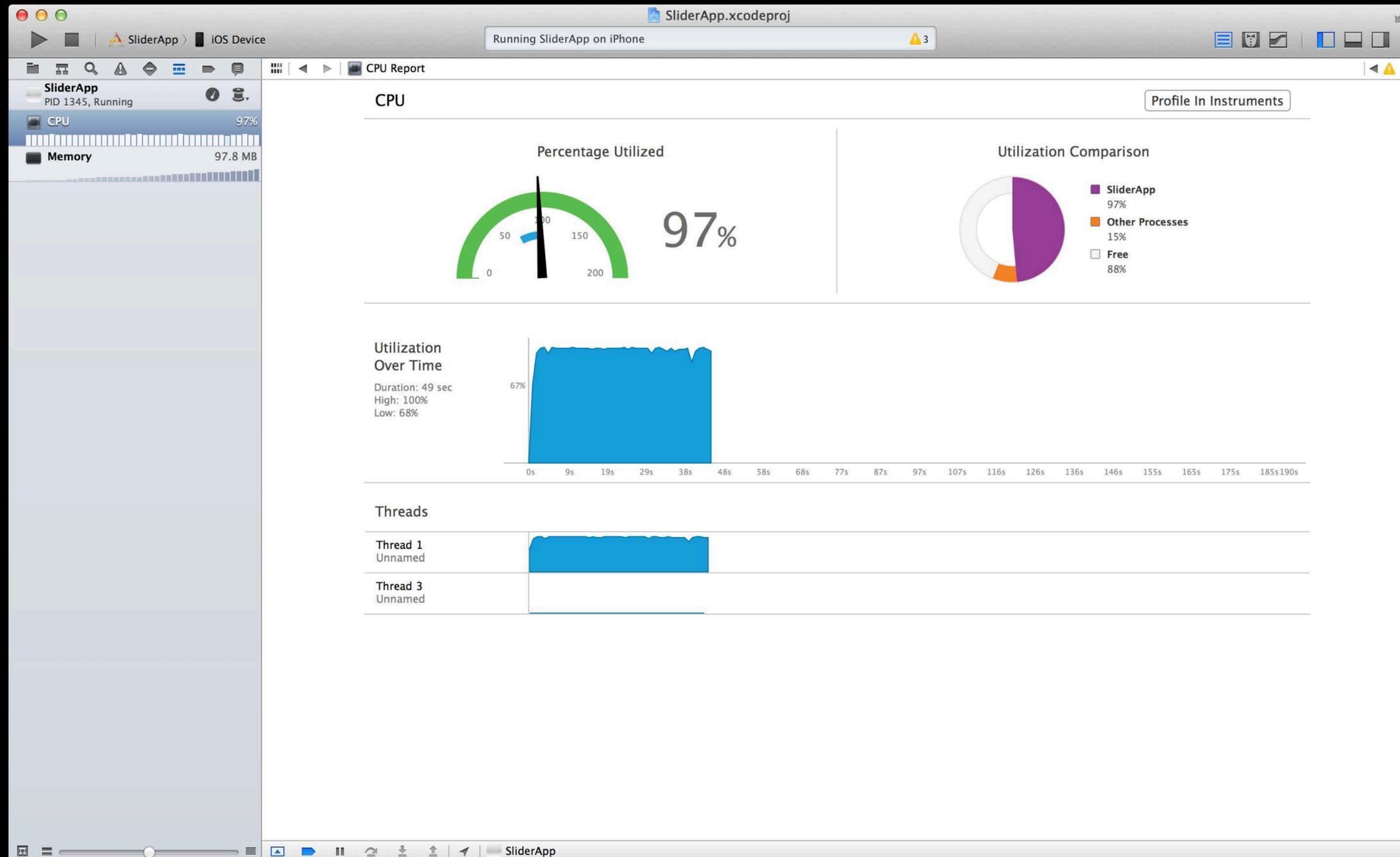
- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (void)insertNewObject:(id)sender
{
    if (!_objects) {
        _objects = [[NSMutableArray alloc] init];
    }
    [_objects addObject:@"Adjust Size" atIndex:0];
    NSIndexPath *indexPath = [NSIndexPath indexPathForRow:0 inSection:0];
    [self.tableView insertRowsAtIndexPaths:@[indexPath] withRowAnimation:UITableViewRowAnimationAutomatic];
}

#pragma mark - Table View

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}
```

# CPU Report



# CPU Report

SliderApp.xcodeproj

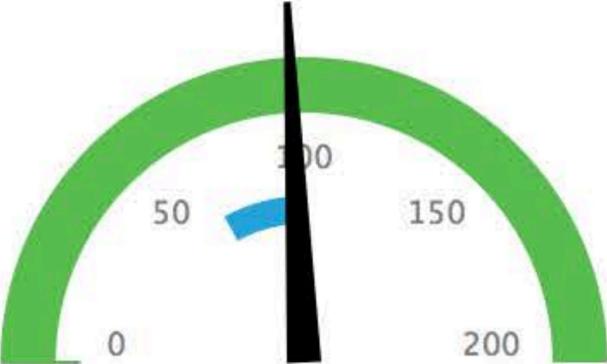
Running SliderApp on iPhone ⚠️ 3

erApp > iOS Device

CPU Report

### CPU

Percentage Utilized



97%

Utilization Over Time



Duration: 40 sec

97.8 MB

97%

Utilizat

67%

# CPU Report

SliderApp.xcodeproj

one ⚠️ 3

[Profile In Instruments](#)

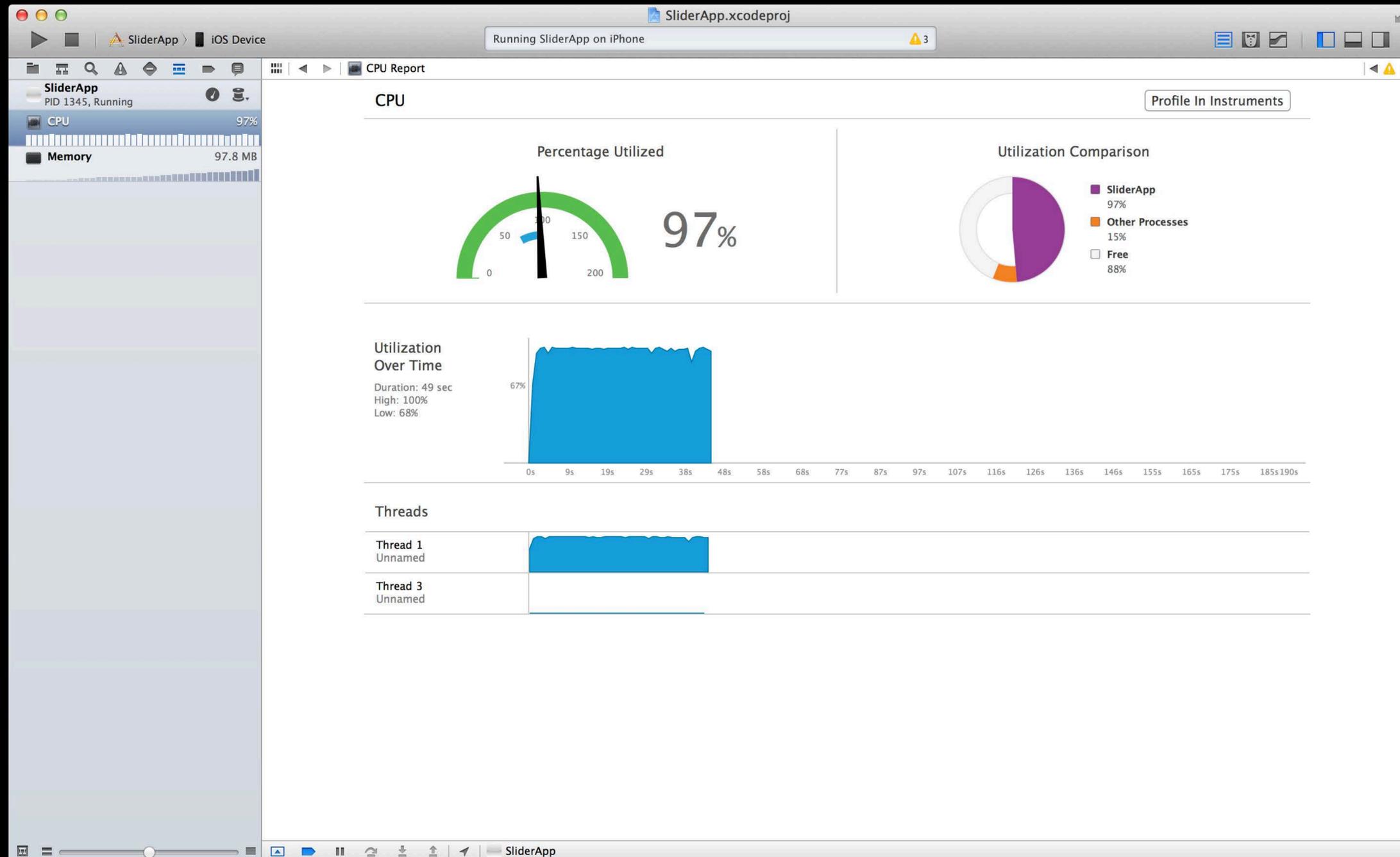
Utilized

97%

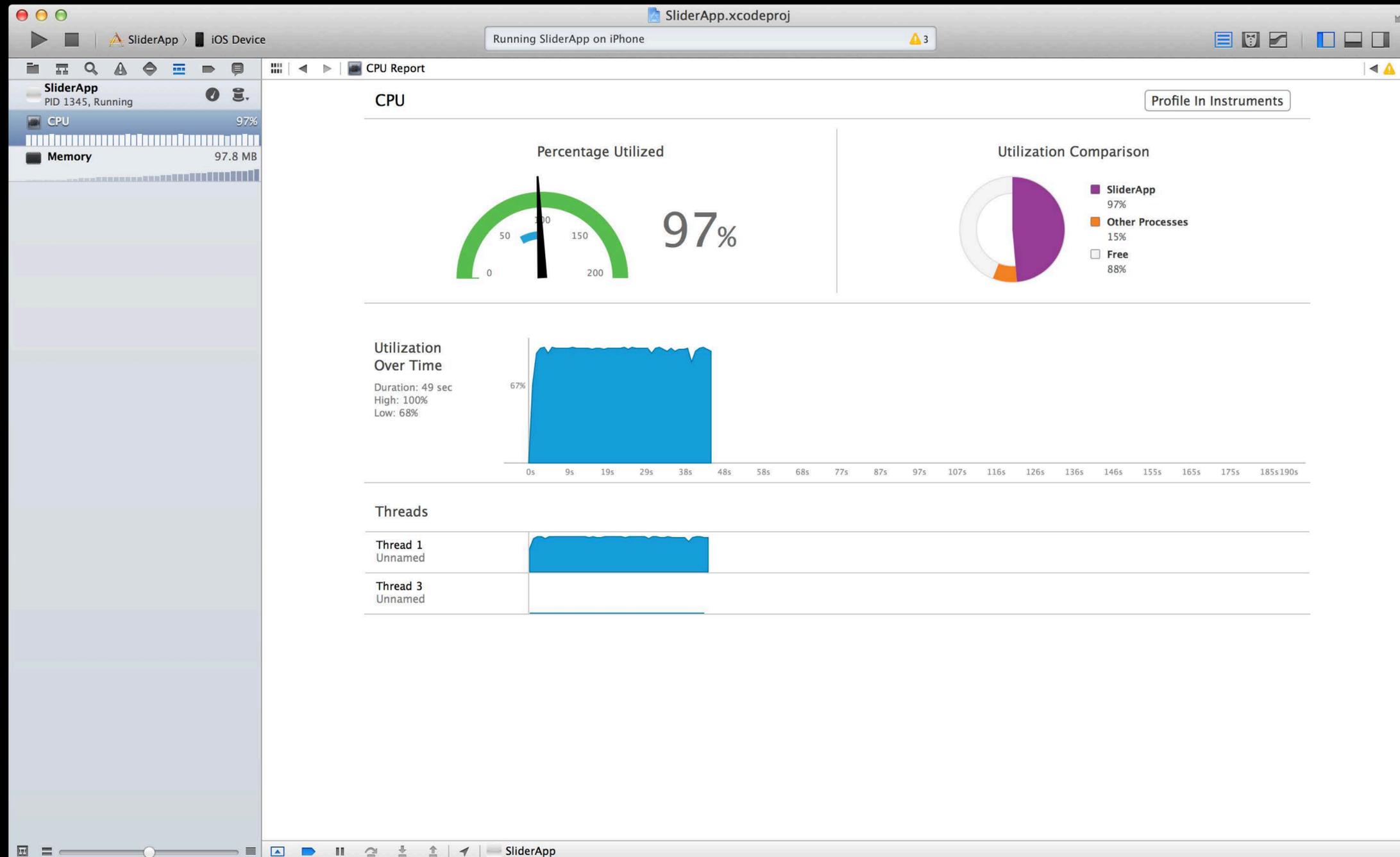
Utilization Comparison

Category	Percentage
SliderApp	97%
Other Processes	15%
Free	88%

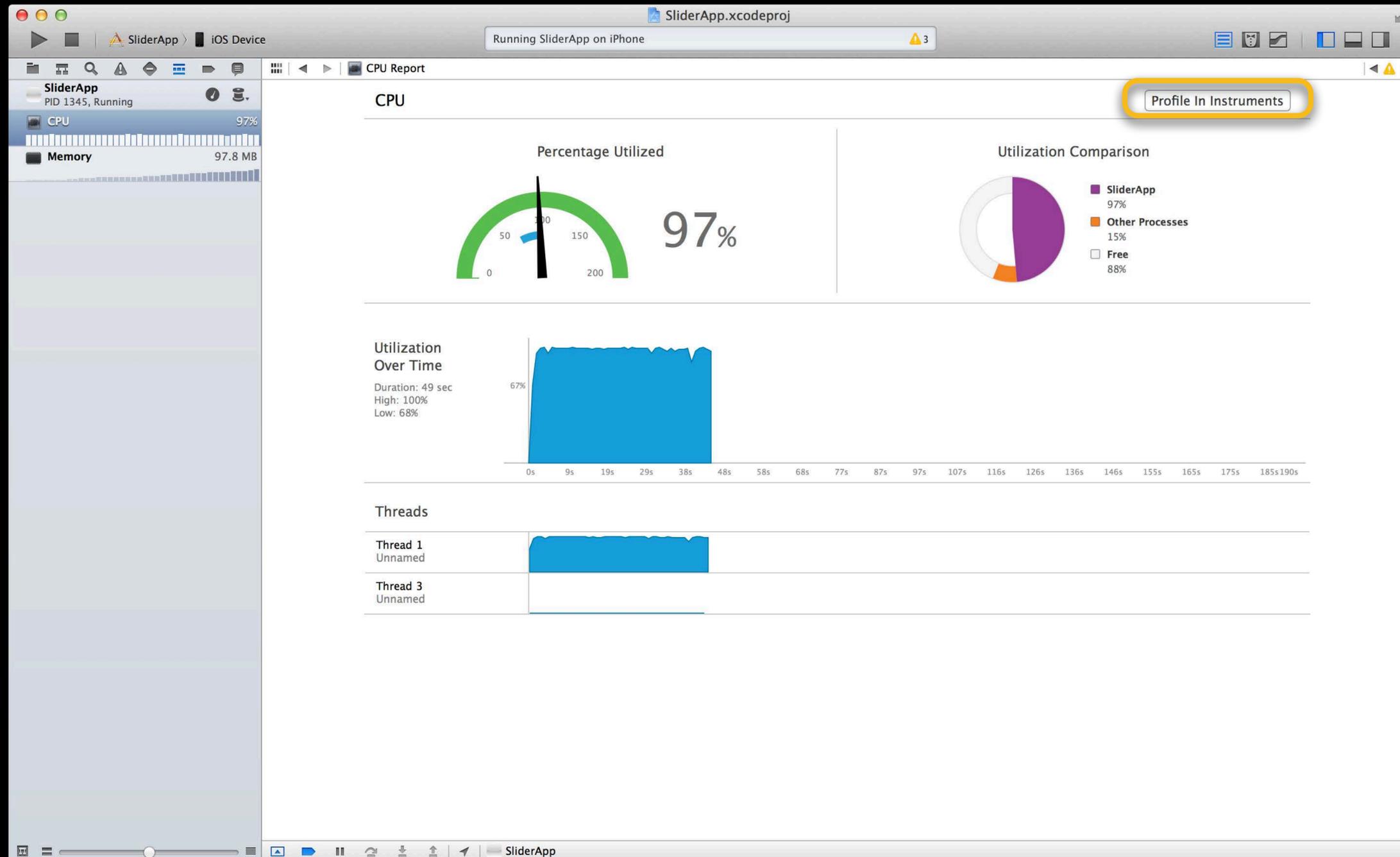
# CPU Report



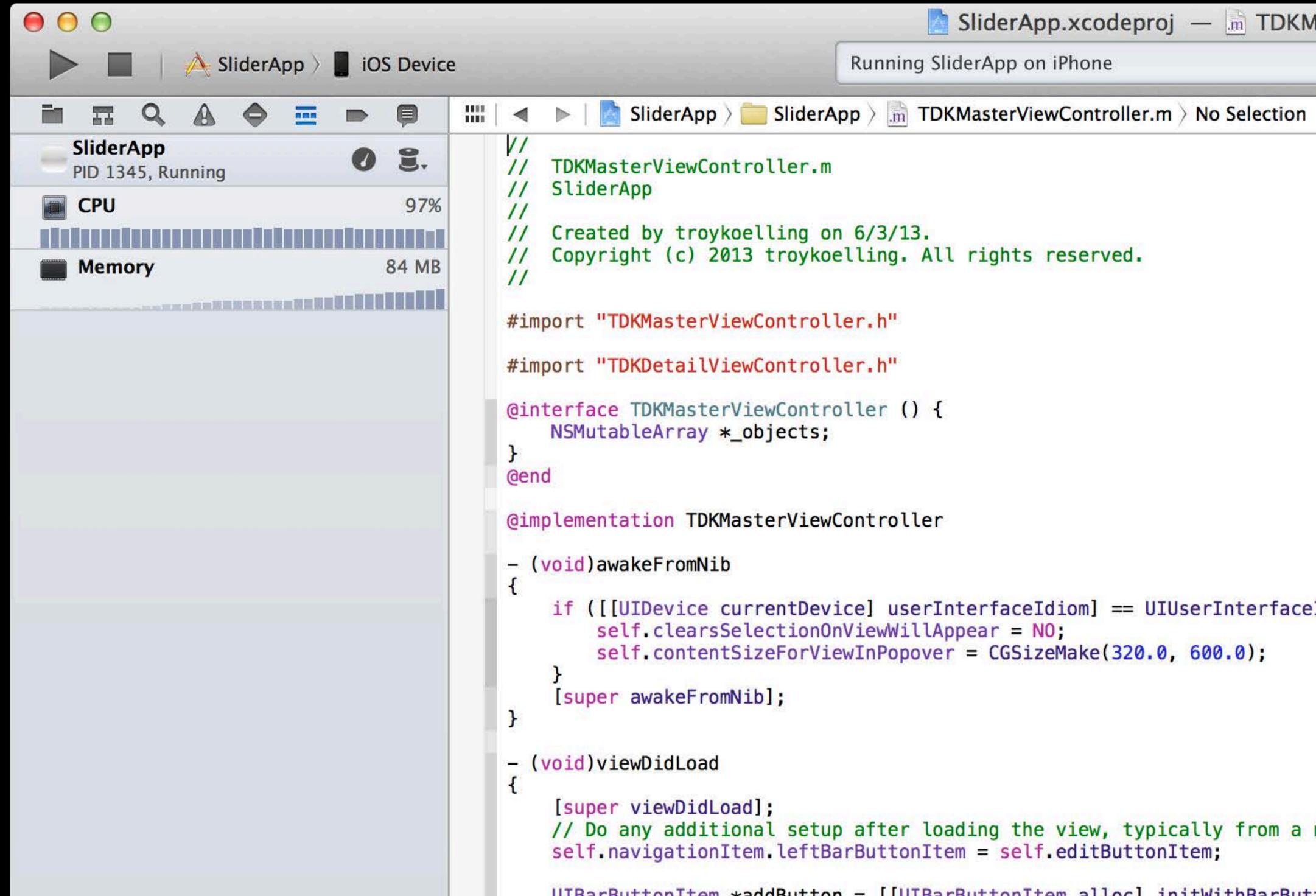
# CPU Report



# CPU Report



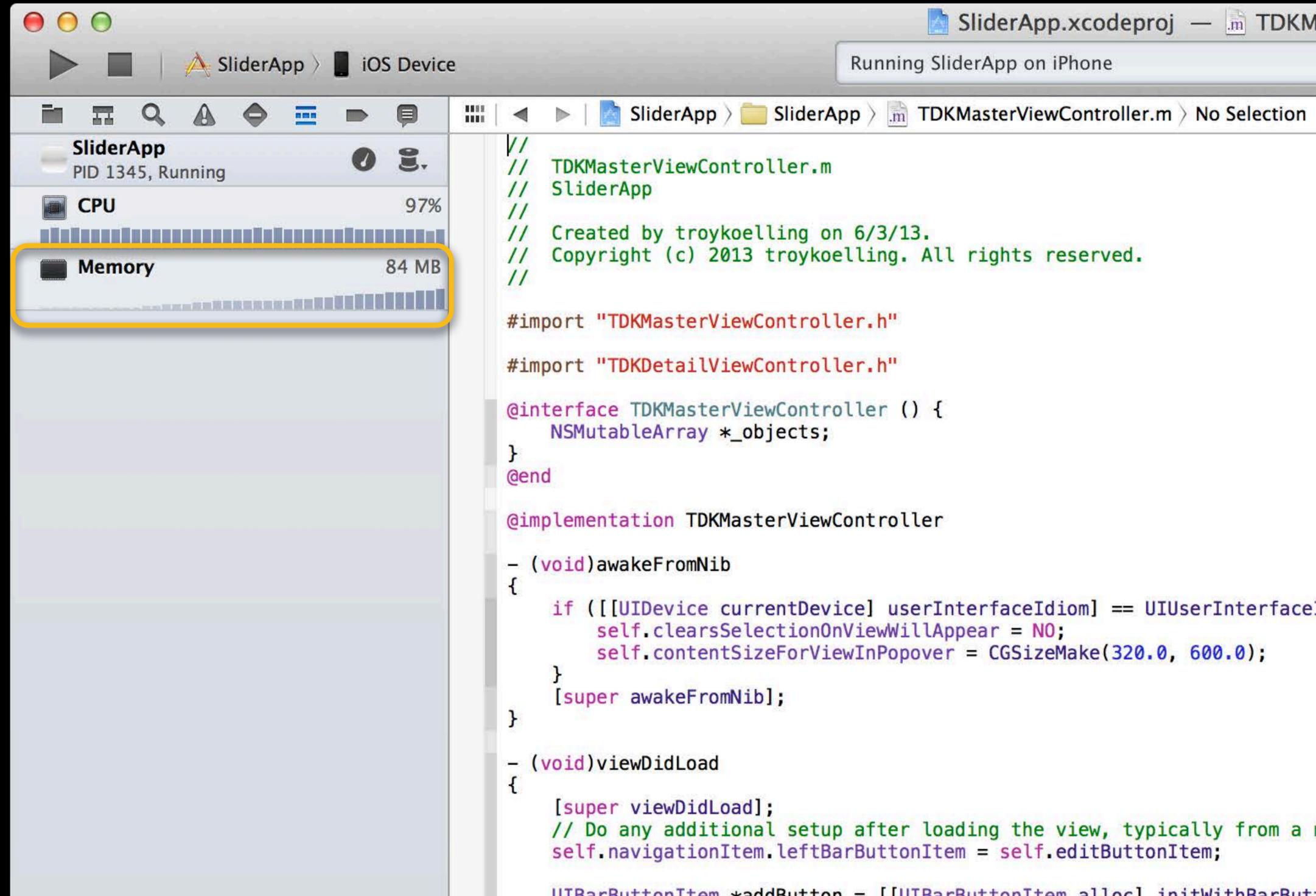
# Memory Report



The screenshot shows the Xcode interface with a memory report for the SliderApp on an iPhone. The report indicates that the app is running with PID 1345, CPU usage is at 97%, and memory usage is at 84 MB. The code editor displays the implementation of TDKMasterViewController.m, which includes imports for TDKMasterViewController.h and TDKDetailViewController.h, an interface definition for TDKMasterViewController, and implementation methods for awakeFromNib and viewDidLoad.

```
//  
// TDKMasterViewController.m  
// SliderApp  
//  
// Created by troykoelling on 6/3/13.  
// Copyright (c) 2013 troykoelling. All rights reserved.  
//  
  
#import "TDKMasterViewController.h"  
  
#import "TDKDetailViewController.h"  
  
@interface TDKMasterViewController () {  
    NSMutableArray *_objects;  
}  
@end  
  
@implementation TDKMasterViewController  
  
- (void)awakeFromNib  
{  
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPhone)  
        self.clearsSelectionOnViewWillAppear = NO;  
        self.contentSizeForViewInPopover = CGSizeMake(320.0, 600.0);  
    }  
    [super awakeFromNib];  
}  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib  
    self.navigationItem.leftBarButtonItem = self.editButtonItem;  
  
    UIBarButtonItem *addButton = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemAdd];  
    self.navigationItem.rightBarButtonItem = addButton;  
}
```

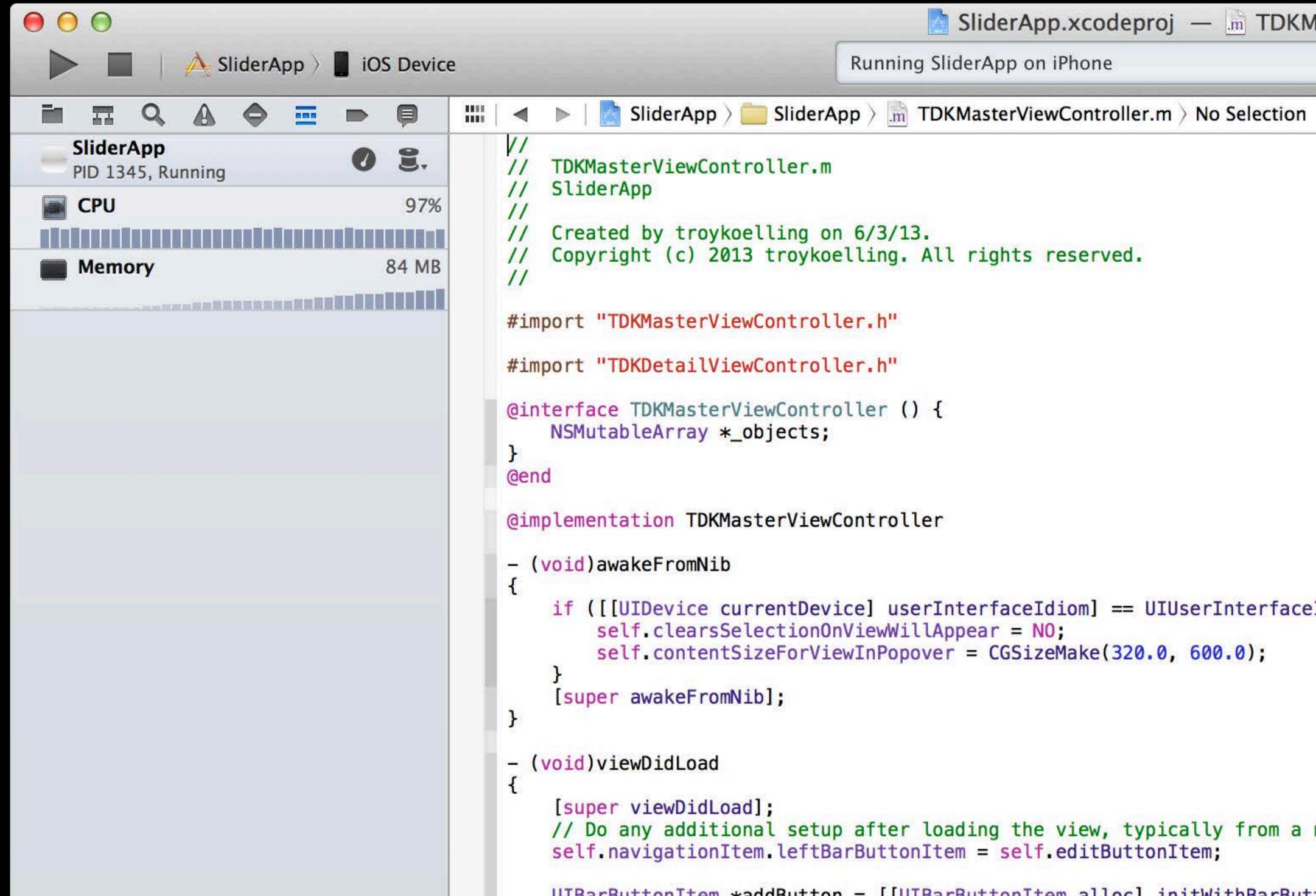
# Memory Report



The screenshot displays the Xcode interface for a project named "SliderApp" running on an iPhone. The top status bar indicates "Running SliderApp on iPhone". The left sidebar shows the project structure, with the "SliderApp" target selected. Below the target name, the PID is listed as "PID 1345, Running". The "CPU" usage is shown as 97%, and the "Memory" usage is highlighted with a yellow box, showing 84 MB. The right pane displays the source code for "TDKMasterViewController.m".

```
//  
// TDKMasterViewController.m  
// SliderApp  
//  
// Created by troykoelling on 6/3/13.  
// Copyright (c) 2013 troykoelling. All rights reserved.  
//  
  
#import "TDKMasterViewController.h"  
  
#import "TDKDetailViewController.h"  
  
@interface TDKMasterViewController () {  
    NSMutableArray *_objects;  
}  
@end  
  
@implementation TDKMasterViewController  
  
- (void)awakeFromNib  
{  
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPhone)  
        self.clearsSelectionOnViewWillAppear = NO;  
        self.contentSizeForViewInPopover = CGSizeMake(320.0, 600.0);  
    }  
    [super awakeFromNib];  
}  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib  
    self.navigationItem.leftBarButtonItem = self.editButtonItem;  
  
    UIBarButtonItem *addButton = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemAdd];  
    self.navigationItem.rightBarButtonItem = addButton;  
}
```

# Memory Report



The screenshot displays the Xcode interface for a project named "SliderApp.xcodeproj". The top status bar indicates "Running SliderApp on iPhone". The left sidebar shows the project structure and a performance monitor for the running application. The performance monitor shows:

- SliderApp (PID 1345, Running)
- CPU: 97%
- Memory: 84 MB

The main editor window shows the source code for "TDKMasterViewController.m". The code includes comments, imports, an interface definition, and implementation methods.

```
//
// TDKMasterViewController.m
// SliderApp
//
// Created by troykoelling on 6/3/13.
// Copyright (c) 2013 troykoelling. All rights reserved.
//

#import "TDKMasterViewController.h"
#import "TDKDetailViewController.h"

@interface TDKMasterViewController () {
    NSMutableArray *_objects;
}
@end

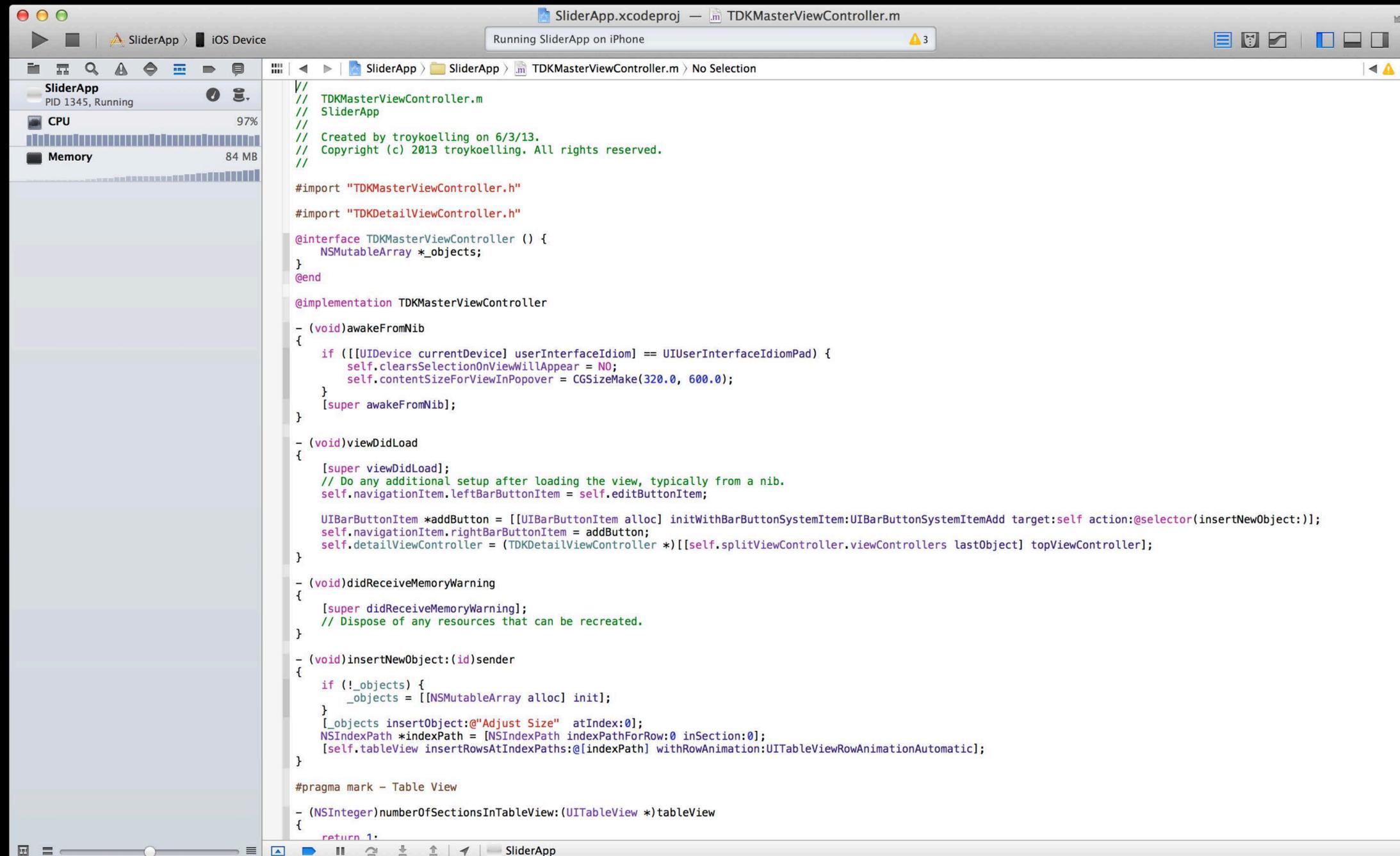
@implementation TDKMasterViewController

- (void)awakeFromNib
{
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) {
        self.clearsSelectionOnViewWillAppear = NO;
        self.contentSizeForViewInPopover = CGSizeMake(320.0, 600.0);
    }
    [super awakeFromNib];
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    self.navigationItem.leftBarButtonItem = self.editButtonItem;

    UIBarButtonItem *addButton = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemAdd];
}
```

# Memory Report



The screenshot shows the Xcode IDE with a project named "SliderApp" running on an iPhone. The interface is divided into three main sections:

- Left Sidebar (Performance Monitor):** Shows the application's performance metrics. CPU usage is at 97%, and Memory usage is at 84 MB. The application is identified as "SliderApp" with PID 1345, running on an iPhone device.
- Top Bar:** Displays the current file being edited: "SliderApp.xcodeproj" - "TDKMasterViewController.m". It also shows the status "Running SliderApp on iPhone" and a warning icon with the number 3.
- Main Editor (Code View):** Shows the source code for "TDKMasterViewController.m". The code includes comments, imports, an interface definition, and implementation methods for lifecycle events and data management.

```
//
//  TDKMasterViewController.m
//  SliderApp
//
//  Created by troykoelling on 6/3/13.
//  Copyright (c) 2013 troykoelling. All rights reserved.
//

#import "TDKMasterViewController.h"
#import "TDKDetailViewController.h"

@interface TDKMasterViewController () {
    NSMutableArray *_objects;
}
@end

@implementation TDKMasterViewController

- (void)awakeFromNib
{
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) {
        self.clearsSelectionOnViewWillAppear = NO;
        self.contentSizeForViewInPopover = CGSizeMake(320.0, 600.0);
    }
    [super awakeFromNib];
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    self.navigationItem.rightBarButtonItem = self.editButtonItem;

    UIBarButtonItem *addButton = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemAdd target:self action:@selector(insertNewObject:)];
    self.navigationItem.rightBarButtonItem = addButton;
    self.detailViewController = (TDKDetailViewController *)[self.splitViewController.viewControllers lastObject] topViewController;
}

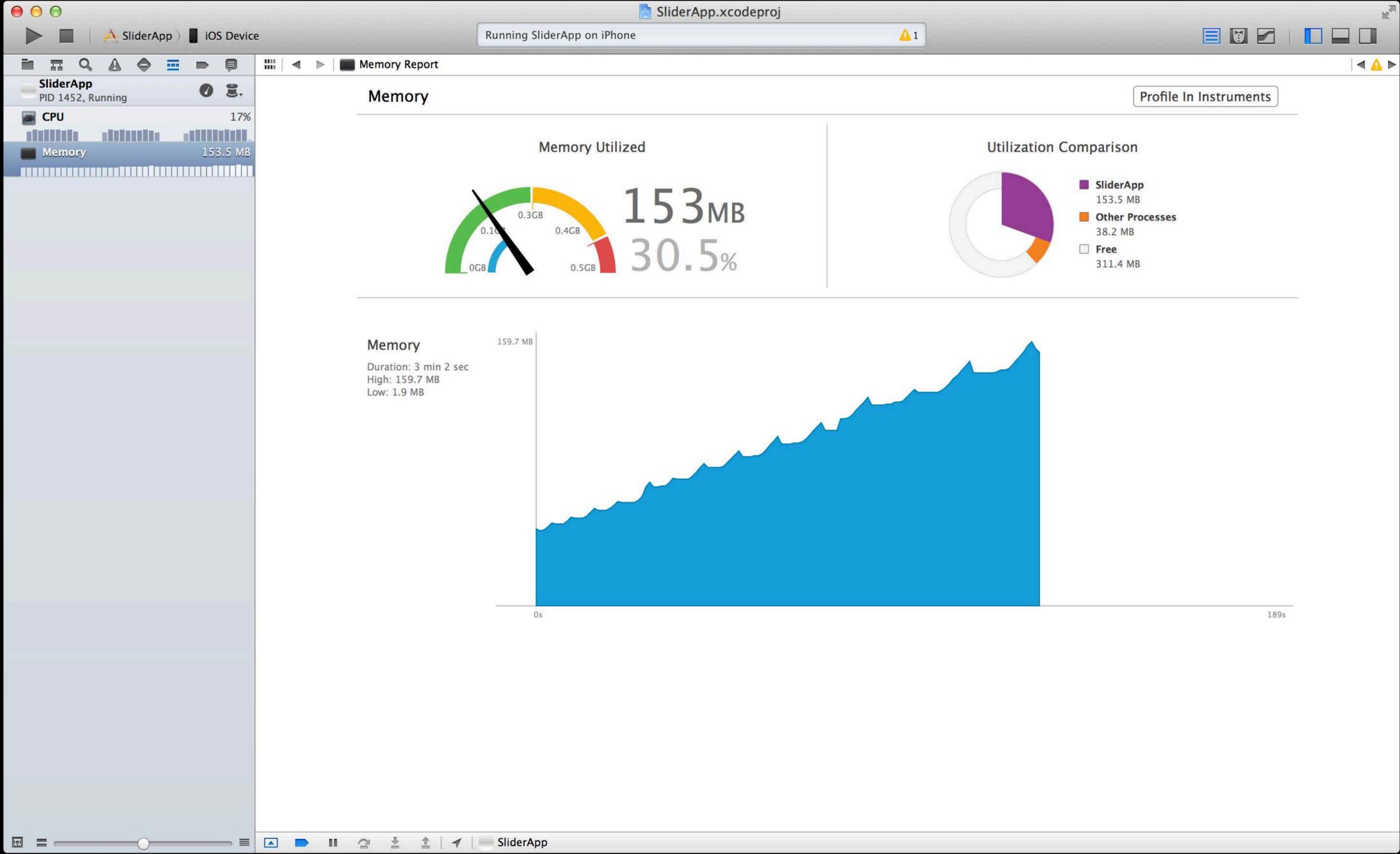
- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (void)insertNewObject:(id)sender
{
    if (!_objects) {
        _objects = [[NSMutableArray alloc] init];
    }
    [_objects insertObject:@"Adjust Size" atIndex:0];
    NSIndexPath *indexPath = [NSIndexPath indexPathForRow:0 inSection:0];
    [self.tableView insertRowsAtIndexPaths:@[indexPath] withRowAnimation:UITableViewRowAnimationAutomatic];
}

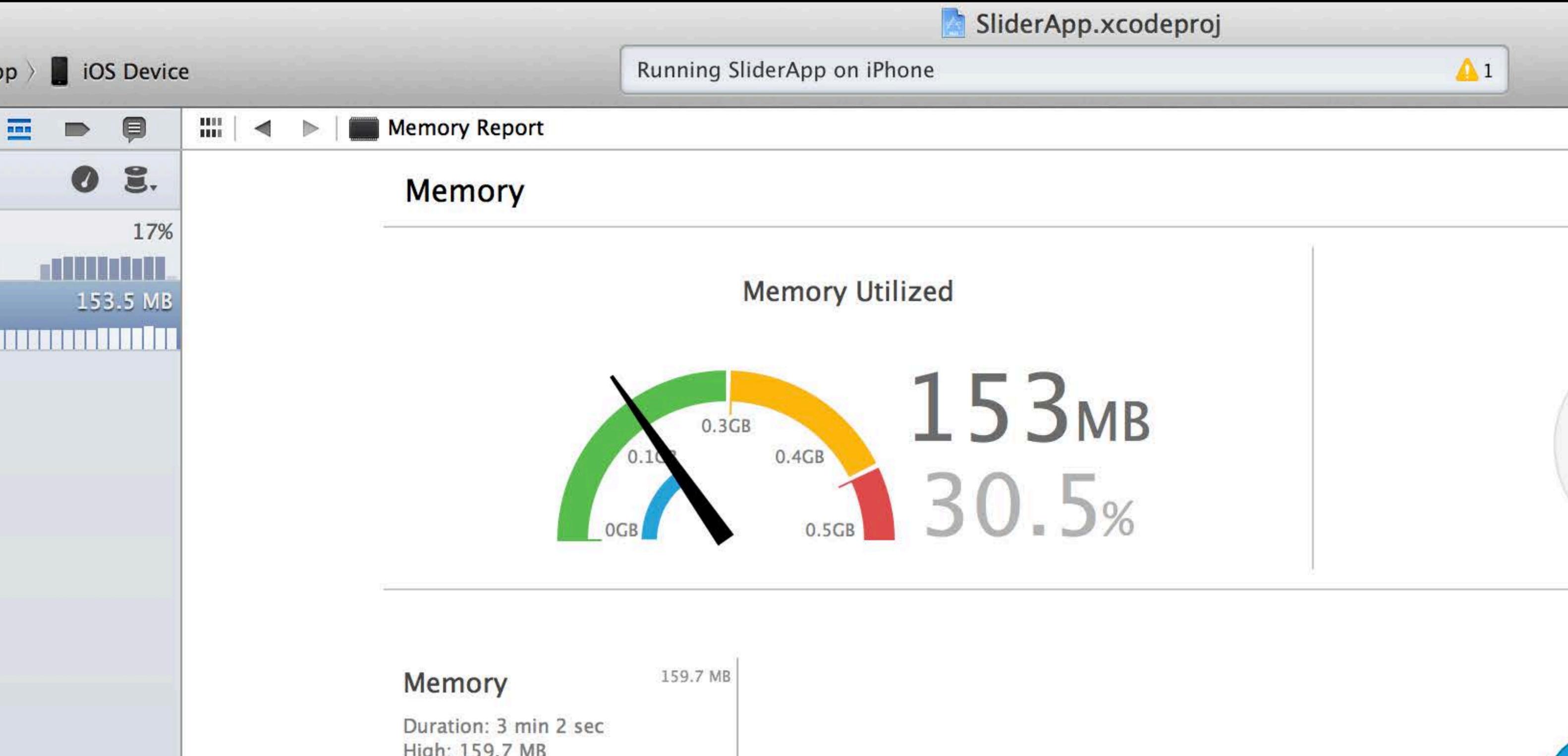
#pragma mark - Table View

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}
```

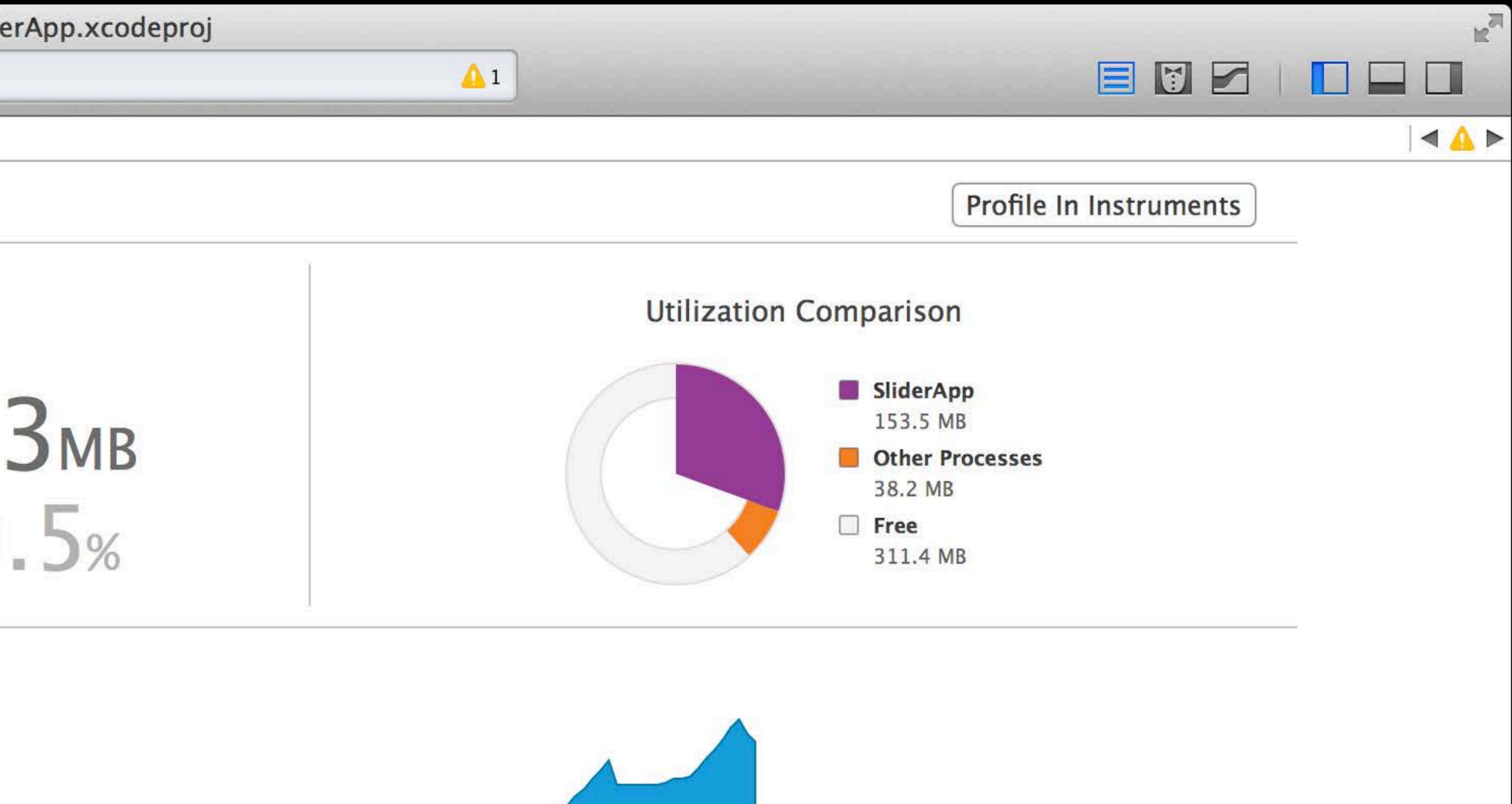
# Memory Report



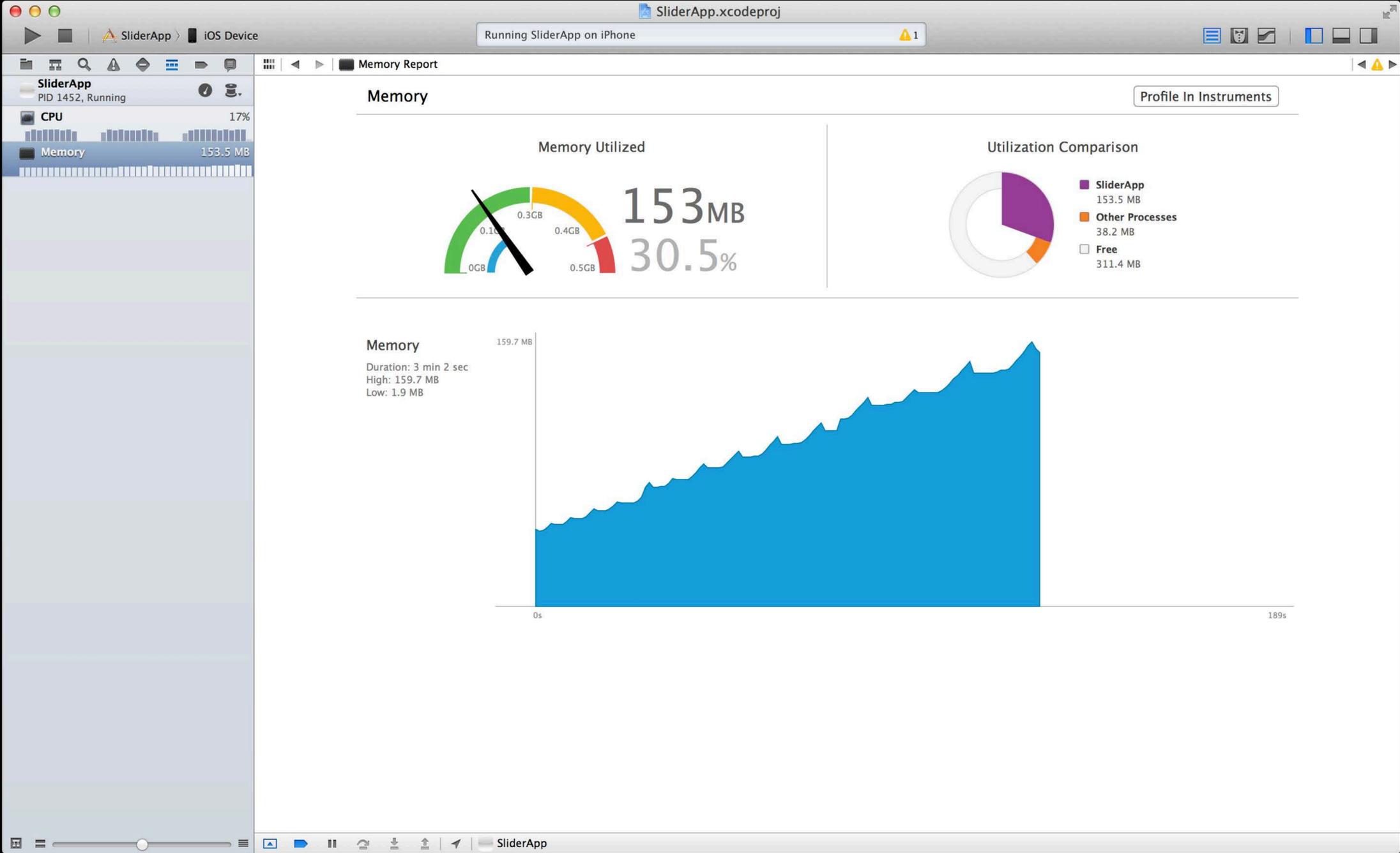
# Memory Report



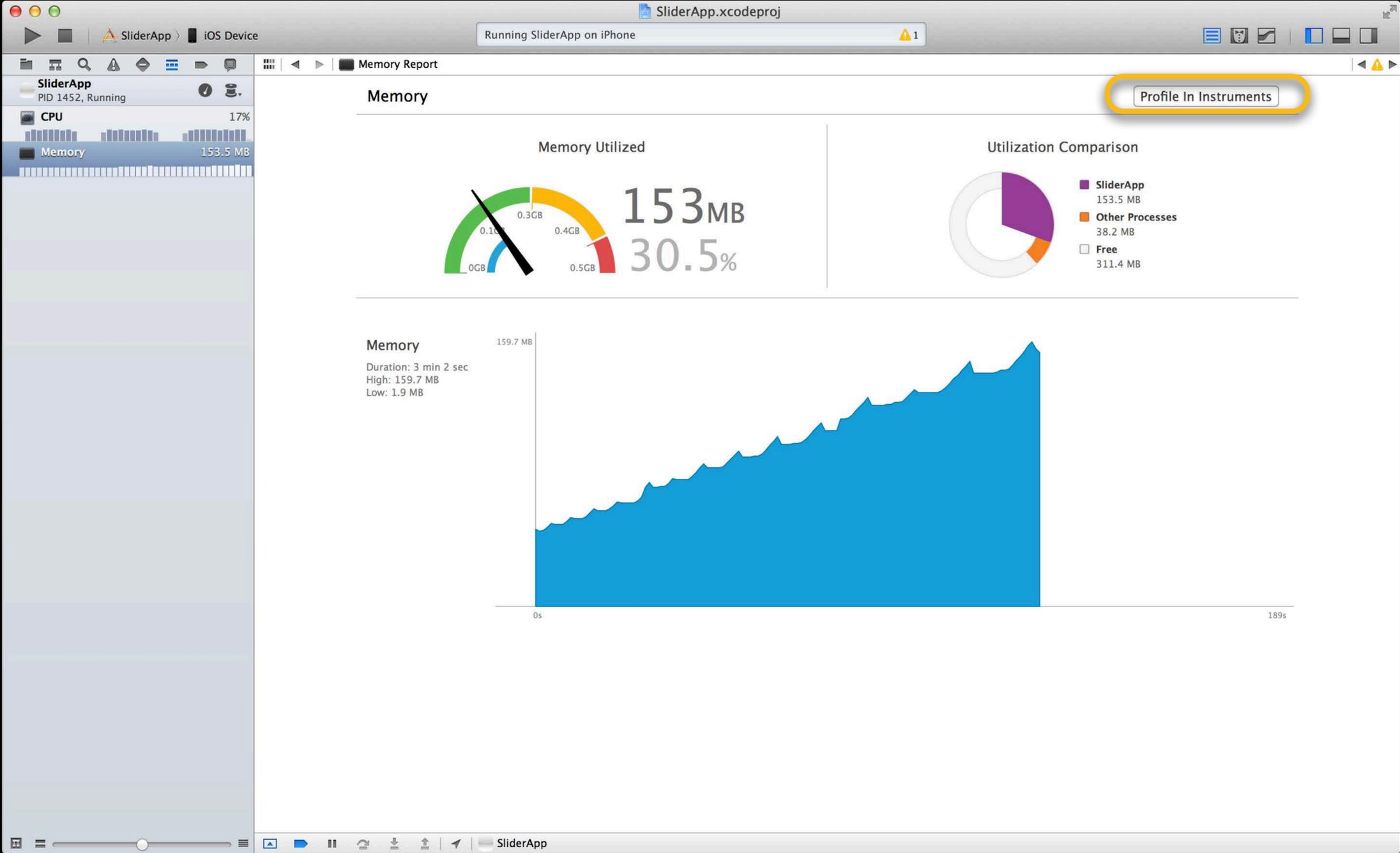
# Memory Report



# Memory Report



# Memory Report



*Demo*

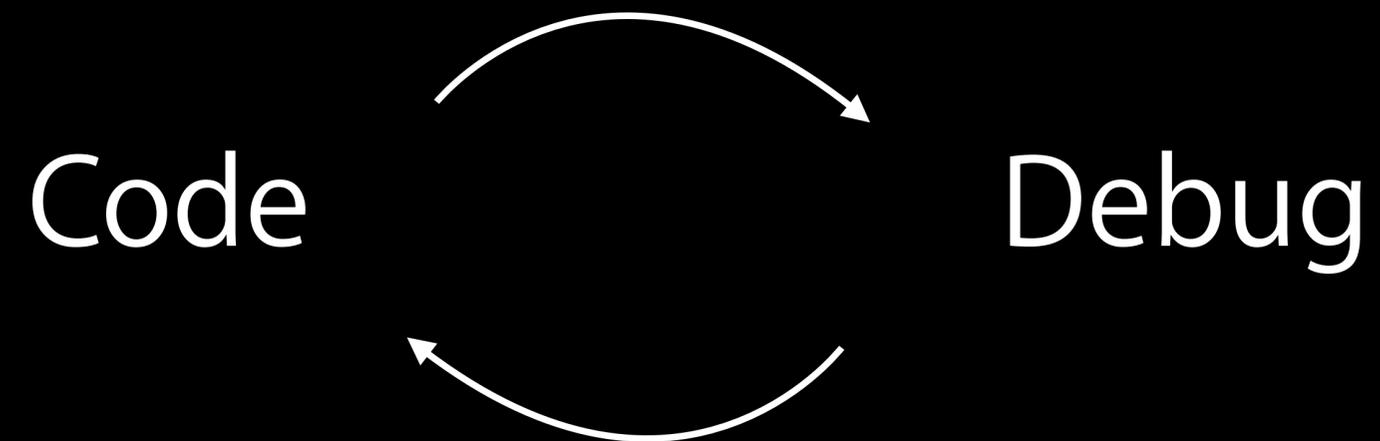
Xcode debugging reports

**Han Ming Ong**

Xcode Debugger UI Engineer

# Data Tips and Quick Look

# Workflows for Debugging



# Debugging Your Code

```
ImageFilterFun.xcworkspace — IFFCImageView.m
Finished running ImageFilterFun : ImageFilterFun No Issues
ImageFilterFun > ImageFilterFun > IFFCImageView.m > -_refilterImage

- (void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc] initWithServiceName:@"com.apple.dt.Xcode.FilterIt"];
    xpcObserverConnection.remoteObjectInterface = [NSXPCInterface interfaceWithProtocol:@protocol(IFFFilterService)];
    [xpcObserverConnection resume];

    NSURL *imageUrl = [NSURL URLWithString:_imageUrlString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, NSSize unfilteredImageSize) {
        dispatch_async(dispatch_get_main_queue(), ^{
            _filteredImage = [[UIImage alloc] initWithData:data];

            // Cache so that we don't have to recalculate.
            //[_imageCache setObject:@(_filter) forKey:_filteredImage];

            _unfilteredImageSize = unfilteredImageSize;

            [self removeConstraint:_aspectRatioConstraint];

            // maintain the aspect ratio of the image.
            if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
                CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
                _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutRelationEqual toItem:self
                    attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

                [self addConstraint:_aspectRatioConstraint];
            }

            _filterChanged = YES;
            [self setNeedsDisplay:YES];
        });
    }];
}

- (void)setRotation:(CGFloat)rotation
{
    if ((int)_rotation != (int)rotation) {
        _rotation = rotation;

        _rotationChanged = YES;
        [self _refilterImage];

        [self setNeedsDisplay:YES];
    }
}

#pragma mark -
#pragma mark NSObject Overrides

- (void)drawRect:(NSRect)dirtyRect
{
    BOOL shouldDraw = _filterChanged;
    if (_rotationChanged) {
        [self setBoundsRotation:self.rotation * -1];
        shouldDraw = YES;
    }

    if (shouldDraw) {
        NSRect fromRect = NSMakeRect(0.0, 0.0, _unfilteredImageSize.width, _unfilteredImageSize.height);
    }
}
```

# Debugging Your Code

The screenshot shows the Xcode IDE interface for a project named "ImageFilterFun". The main window displays the source code of the file "IFFCIImageView.m". A breakpoint is set at line 1.1, which is highlighted in green. The code in the main window is as follows:

```
-(void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc] initWithServiceName:@"com.apple.dt.Xcode.FilterIt"];
    xpcObserverConnection.remoteObjectInterface = [NSXPCInterface interfaceWithProtocol:@protocol(IFFFilterService)];
    [xpcObserverConnection resume];

    NSURL *imageURL = [NSURL URLWithString:_imageURLString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, NSSize unfilteredImageSize) {
        dispatch_async(dispatch_get_main_queue(), ^{
            _filteredImage = [[UIImage alloc] initWithData:data];

            // Cache so that we don't have to recalculate.
            //[_imageCache setObject:@(_filter) forKey:_filteredImage];

            _unfilteredImageSize = unfilteredImageSize;

            [self removeConstraint:_aspectRatioConstraint];

            // maintain the aspect ratio of the image.
            if (_unfilteredImageSize.width > 0 && _unfilteredImageSize.height > 0) {
                CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
                _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeHeight relatedBy:NSLayoutRelationEqual toItem:self
                    attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0.0f];

                [self addConstraint:_aspectRatioConstraint];
            }

            _filterChanged = YES;
            [self setNeedsDisplay:YES];
        });
    }];
};
```

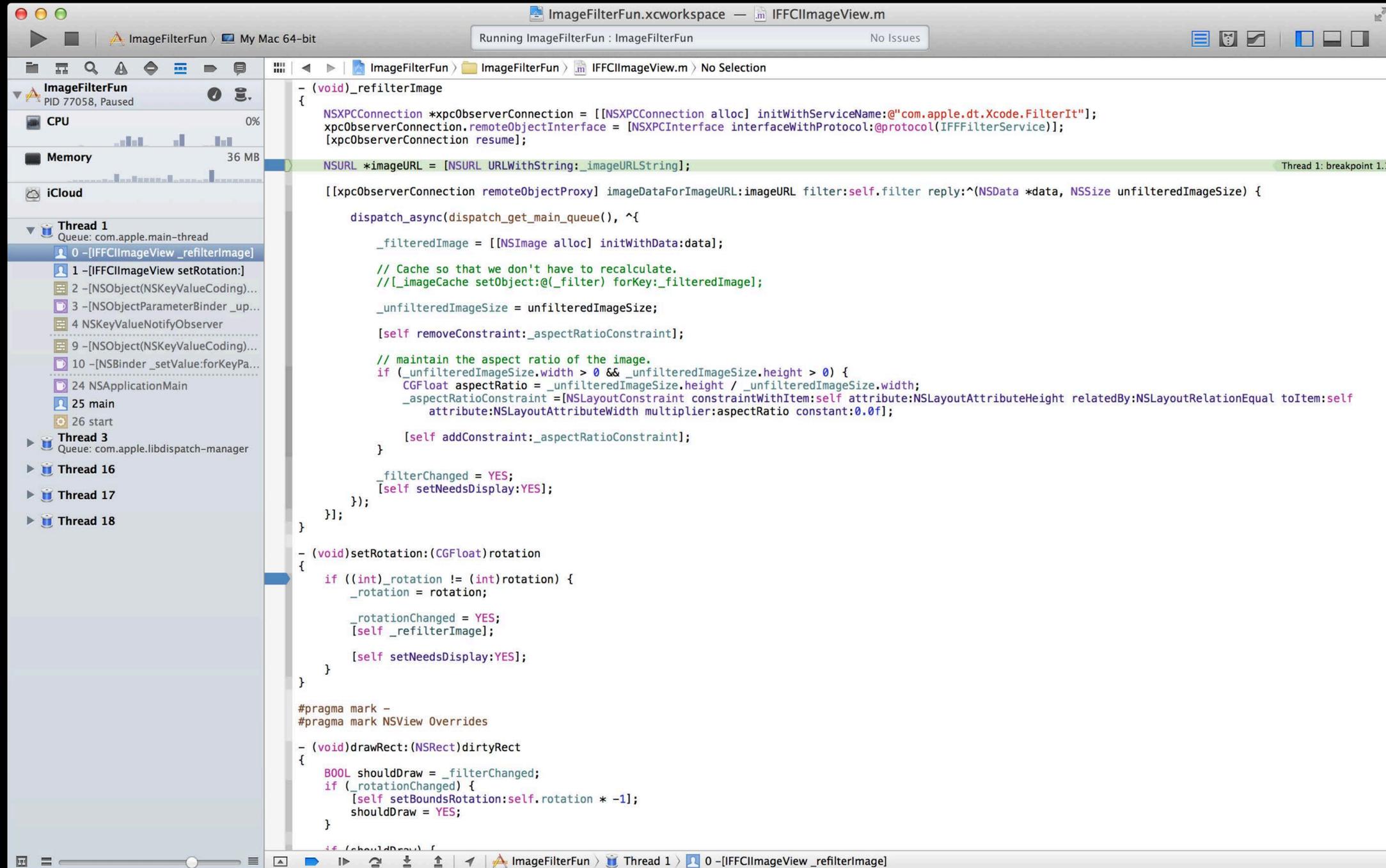
The left sidebar shows the project structure and a list of threads. The selected thread is "Thread 1" with the queue "com.apple.main-thread". The bottom pane shows the variable inspector with the following variables:

- self** = (IFFCIImageView \*) 0x10052b380
- imageURL** = (NSURL \*) 0x7fff5fbfe6c0
- xpcObserverConnection** = (NSXPCConnection \*) 0x100540410
- imageURLString** = (\_\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

The right pane shows thelldb console output for the selected variable self:

```
(lldb) po self
<IFFCIImageView: 0x10052b380>
(lldb) p _imageURLString
(NSString *) $8 = 0x0000000100519f10 @"http://images.apple.com/ipad/overview/
images/hero.png"
(lldb) |
```

# Debugging Your Code



# Data Tips

```
_unfilteredImageSize = size;
```

▶ (width=1216, height=811)  

# Data Tips

```
_unfilteredImageSize = size;
```

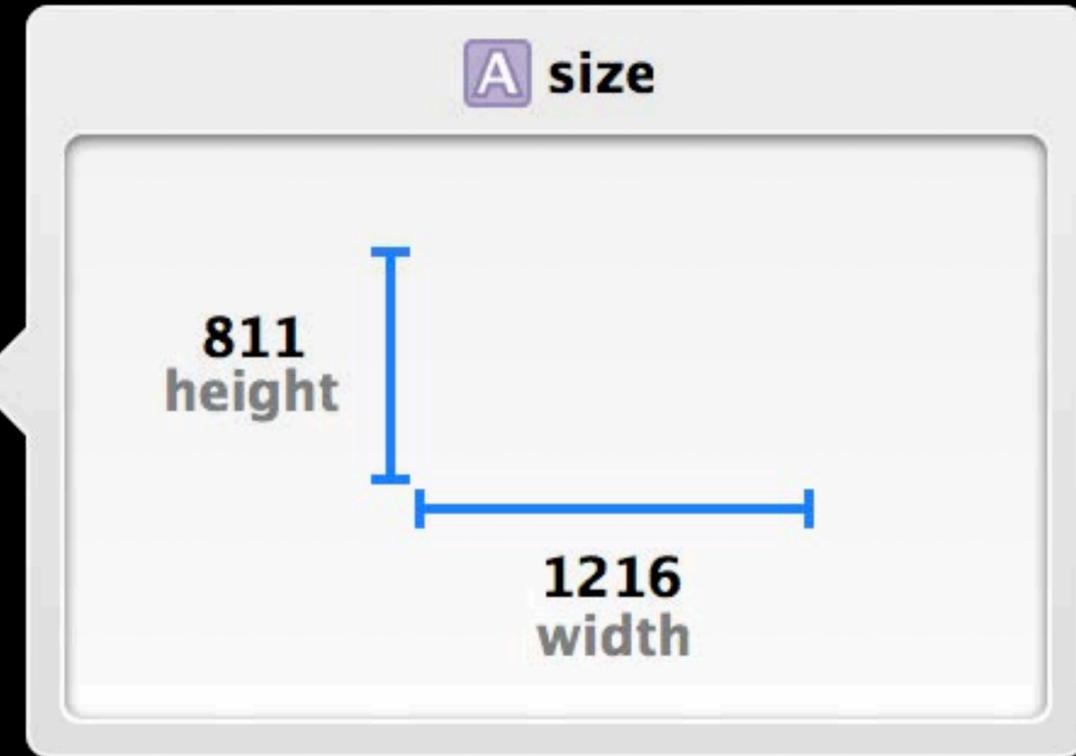
▶ (width=1216, height=811)  

(NSSize) size = (width=1216, height=811)

# Quick Look of Variables

```
_unfilteredImageSize = size;
```

▶ (width=1216, height=811)  



*Demo*

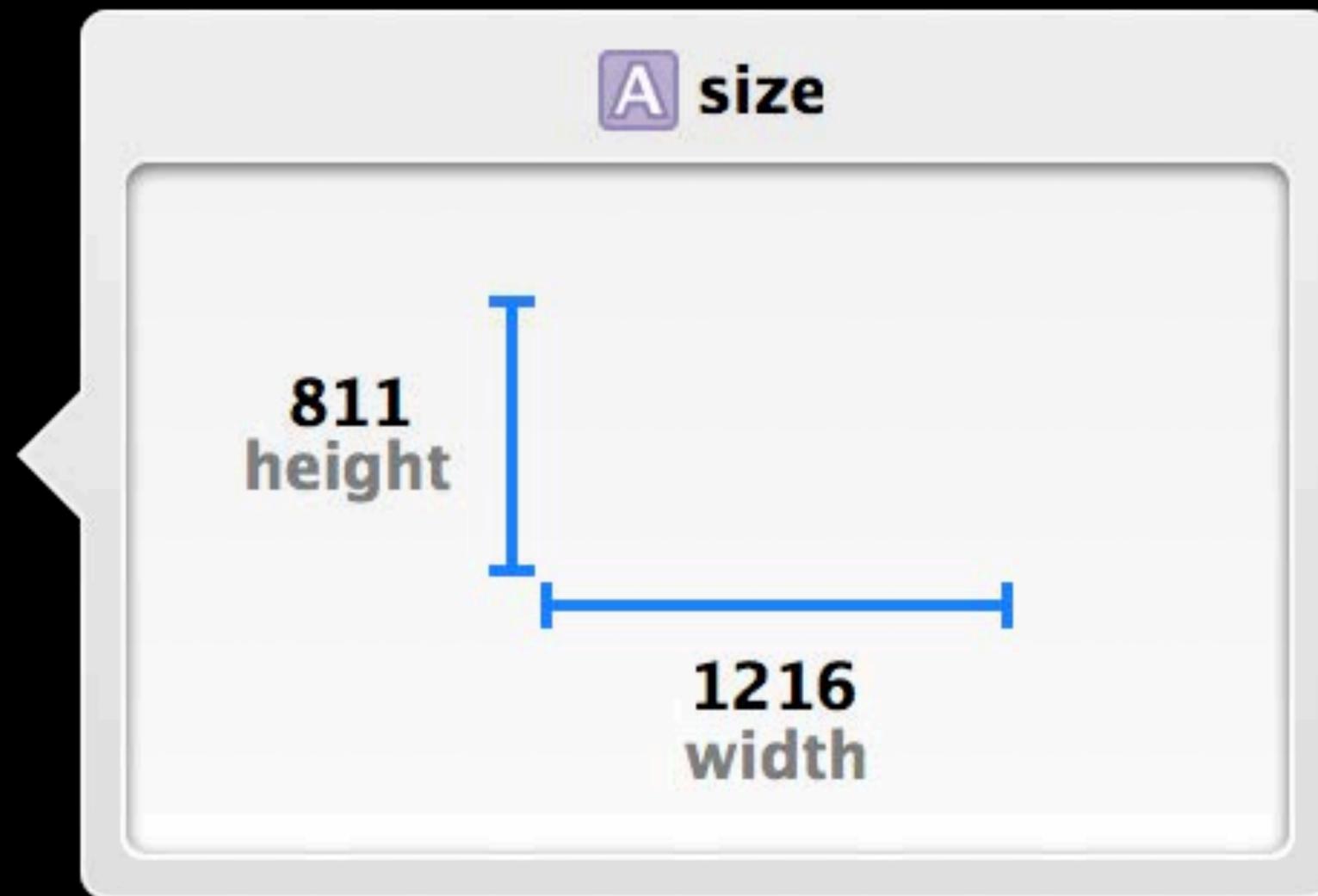
Data Tips and Quick Look

**Christopher Friesen**

Xcode Debugger UI Engineer

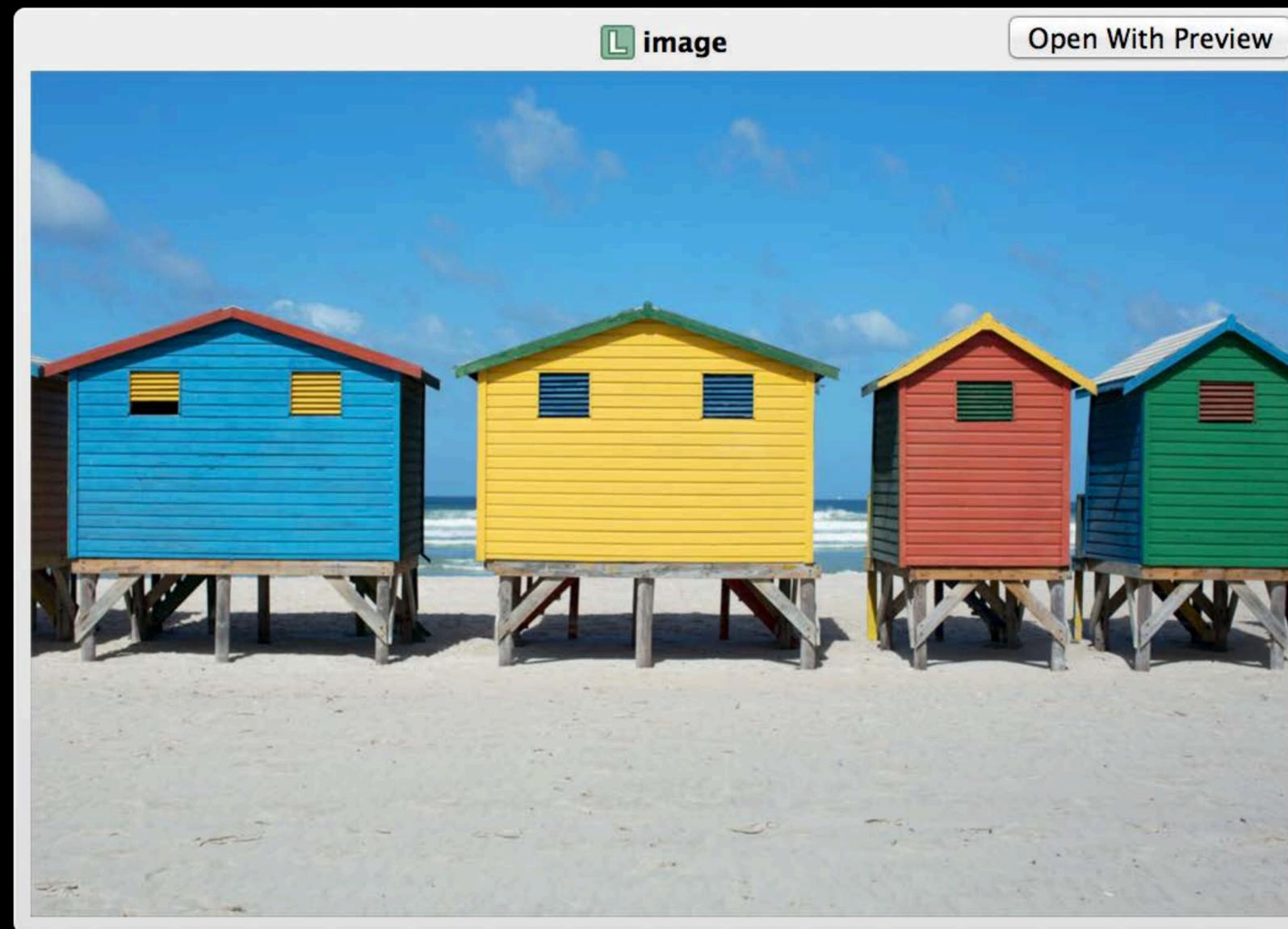
# Quick Look in Xcode

NSSize and CGSize



# Quick Look in Xcode

UIImage, NSImage, CGImageRef, and CIImage



# Quick Look in Xcode

NSString and NSAttributedString, including long text

```
[textField setValue:@"loremIpsum"];
```

▶ @"Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut l...  

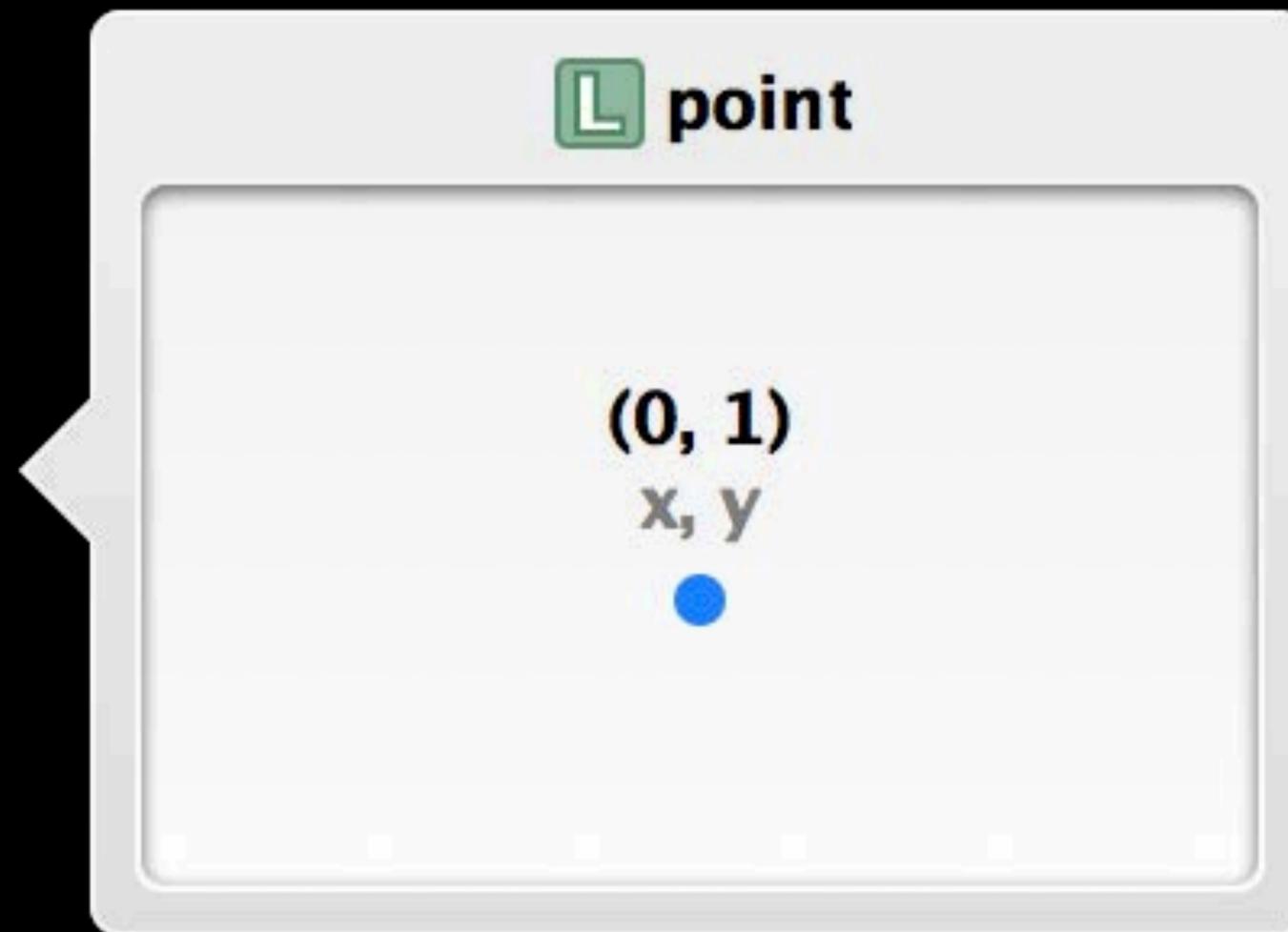
 **loremIpsum**

Open With TextEdit

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

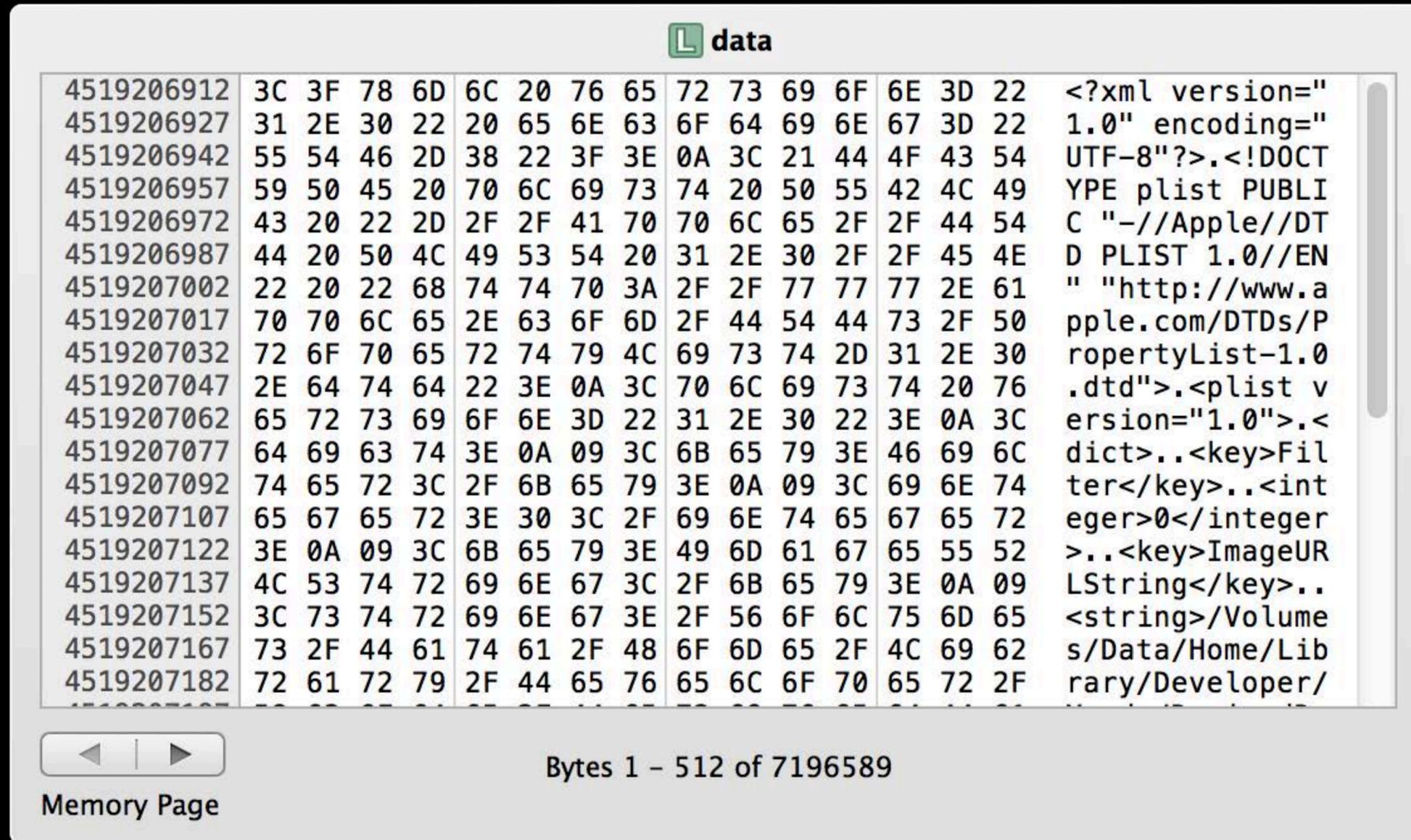
# Quick Look in Xcode

NSPoint and CGPoint



# Quick Look in Xcode

## NSData



The screenshot shows the Quick Look window for an NSData object. The window title is "data". The content is displayed in a table with three columns: memory address, hexadecimal bytes, and the corresponding ASCII text. The text is an XML declaration for a plist file.

Memory Address	Hex Bytes	ASCII Text
4519206912	3C 3F 78 6D 6C 20 76 65 72 73 69 6F 6E 3D 22	<?xml version="
4519206927	31 2E 30 22 20 65 6E 63 6F 64 69 6E 67 3D 22	1.0" encoding="
4519206942	55 54 46 2D 38 22 3F 3E 0A 3C 21 44 4F 43 54	UTF-8"?>.<!DOCT
4519206957	59 50 45 20 70 6C 69 73 74 20 50 55 42 4C 49	YPE plist PUBLI
4519206972	43 20 22 2D 2F 2F 41 70 70 6C 65 2F 2F 44 54	C "-//Apple//DT
4519206987	44 20 50 4C 49 53 54 20 31 2E 30 2F 2F 45 4E	D PLIST 1.0//EN
4519207002	22 20 22 68 74 74 70 3A 2F 2F 77 77 77 2E 61	" "http://www.a
4519207017	70 70 6C 65 2E 63 6F 6D 2F 44 54 44 73 2F 50	pple.com/DTDs/P
4519207032	72 6F 70 65 72 74 79 4C 69 73 74 2D 31 2E 30	ropertyList-1.0
4519207047	2E 64 74 64 22 3E 0A 3C 70 6C 69 73 74 20 76	.dtd">.<plist v
4519207062	65 72 73 69 6F 6E 3D 22 31 2E 30 22 3E 0A 3C	ersion="1.0">.<
4519207077	64 69 63 74 3E 0A 09 3C 6B 65 79 3E 46 69 6C	dict>..<key>Fil
4519207092	74 65 72 3C 2F 6B 65 79 3E 0A 09 3C 69 6E 74	ter</key>..<int
4519207107	65 67 65 72 3E 30 3C 2F 69 6E 74 65 67 65 72	eger>0</integer
4519207122	3E 0A 09 3C 6B 65 79 3E 49 6D 61 67 65 55 52	
4519207137	4C 53 74 72 69 6E 67 3C 2F 6B 65 79 3E 0A 09	LString</key>..
4519207152	3C 73 74 72 69 6E 67 3E 2F 56 6F 6C 75 6D 65	<string>/Volume
4519207167	73 2F 44 61 74 61 2F 48 6F 6D 65 2F 4C 69 62	s/Data/Home/Lib
4519207182	72 61 72 79 2F 44 65 76 65 6C 6F 70 65 72 2F	rary/Developer/

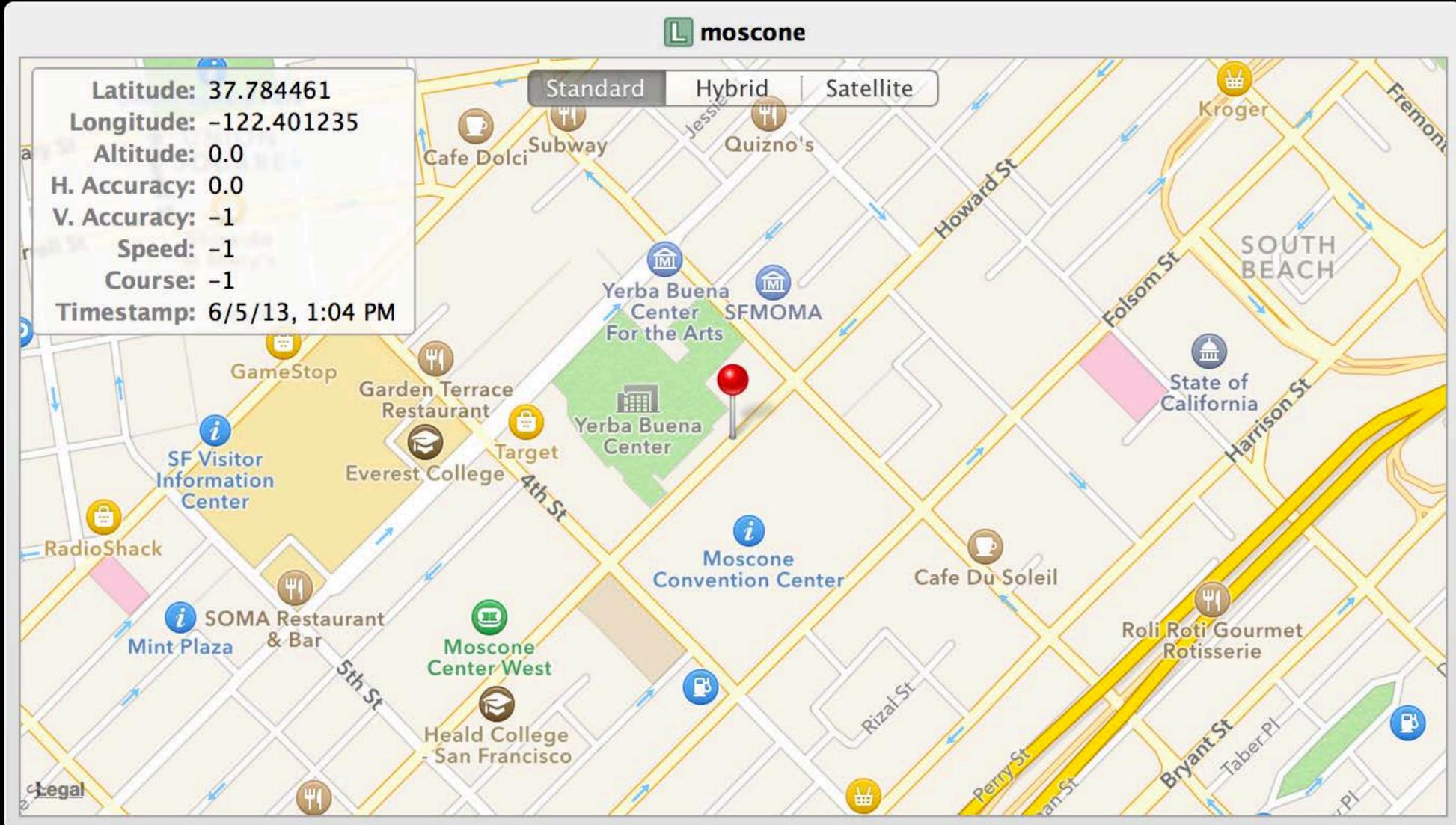
Navigation: ◀ | ▶

Bytes 1 - 512 of 7196589

Memory Page

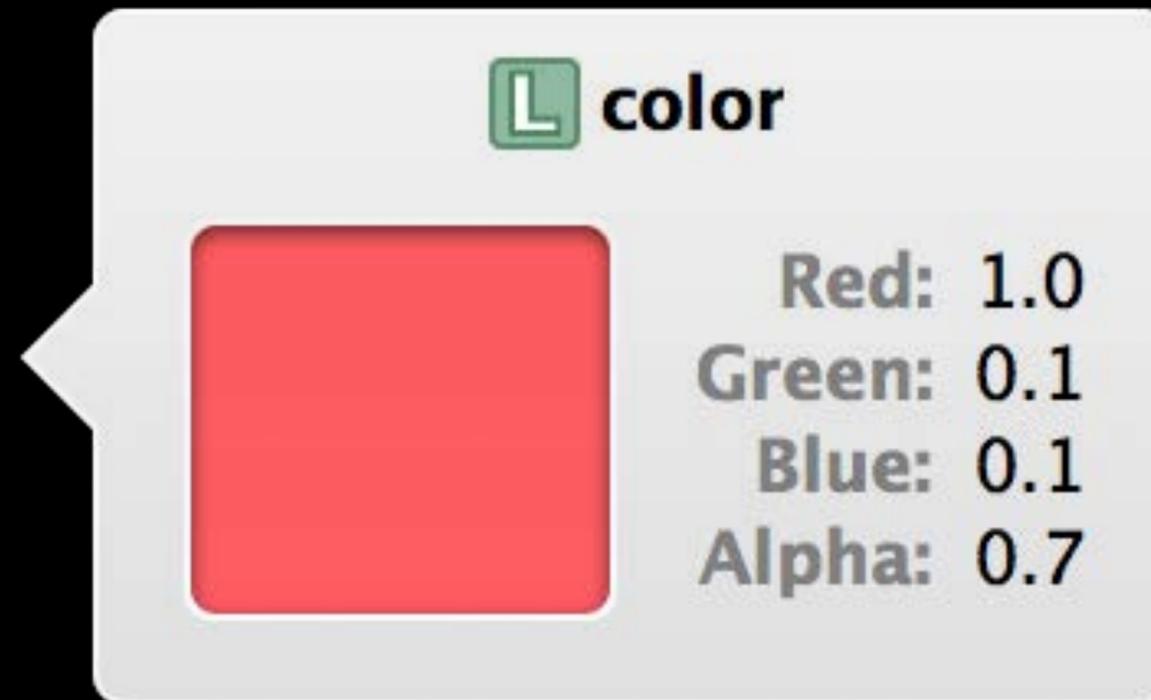
# Quick Look in Xcode

## CLLocation, CLLocationCoordinate2D



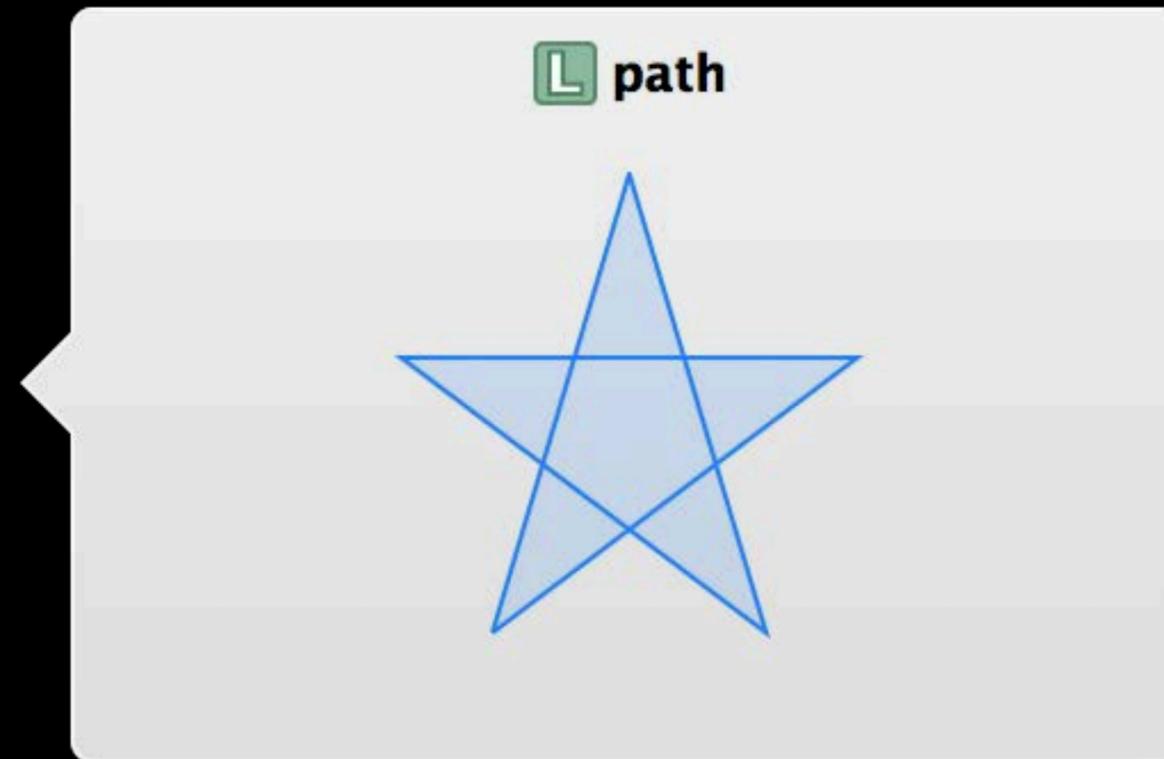
# Quick Look in Xcode

UIColor, NSColor, and CGColor



# Quick Look in Xcode

UIBezierPath and NSBezierPath

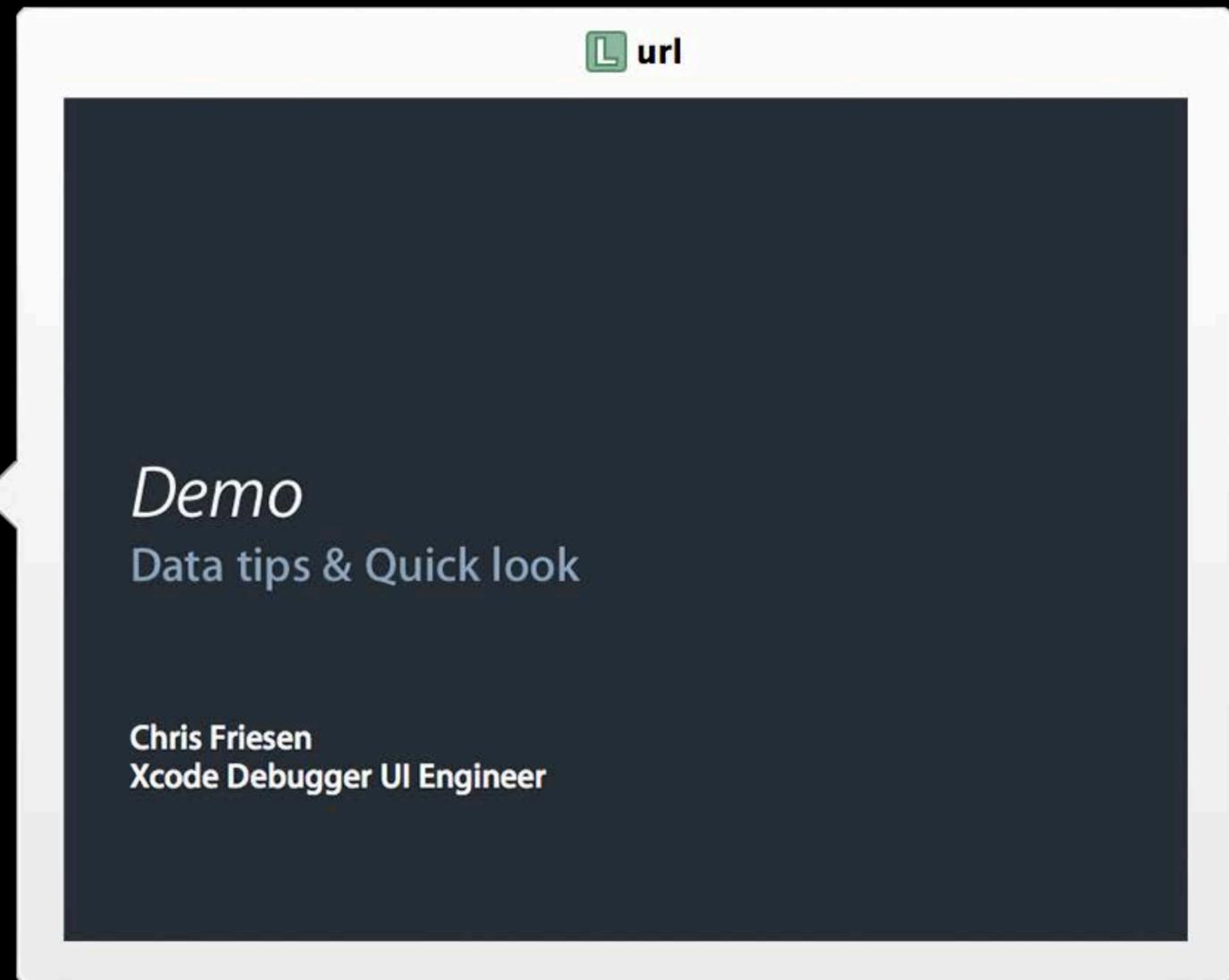


# Quick Look in Xcode

## NSURL

```
[pathControl setURL:url];
```

▶ (NSURL \*) 0x101838bf0

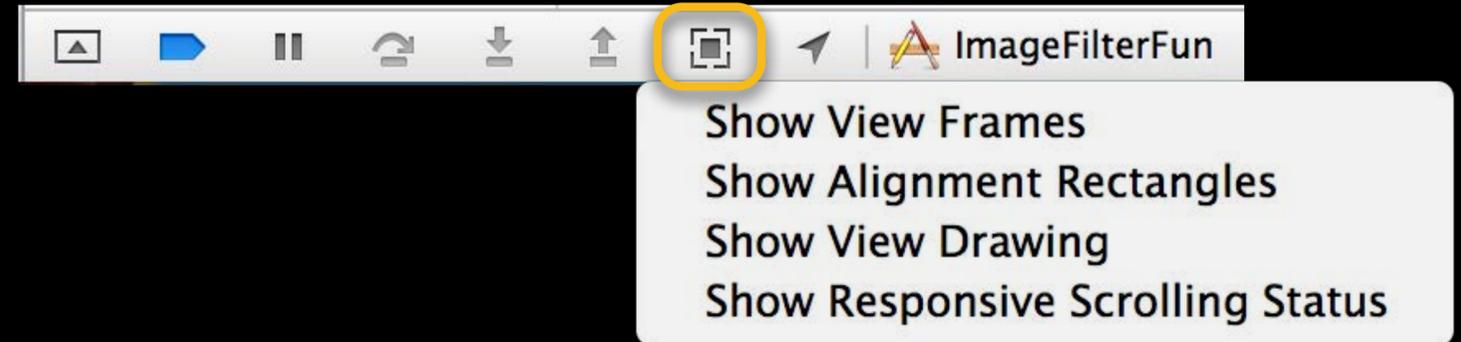




# Drawing Diagnostics



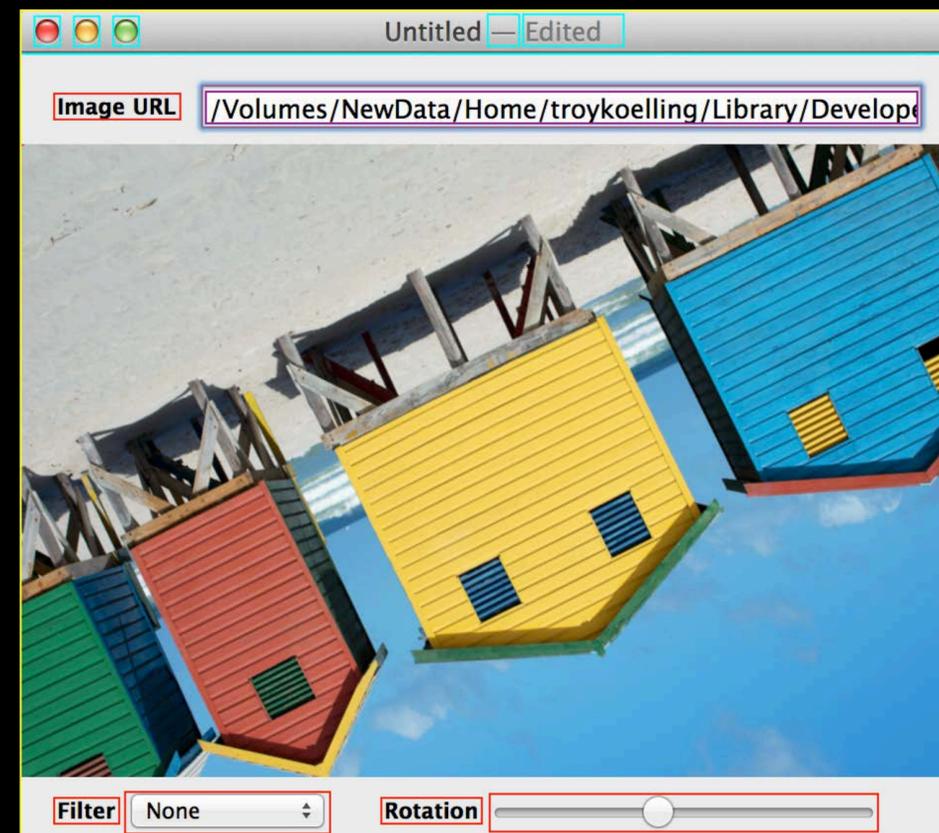
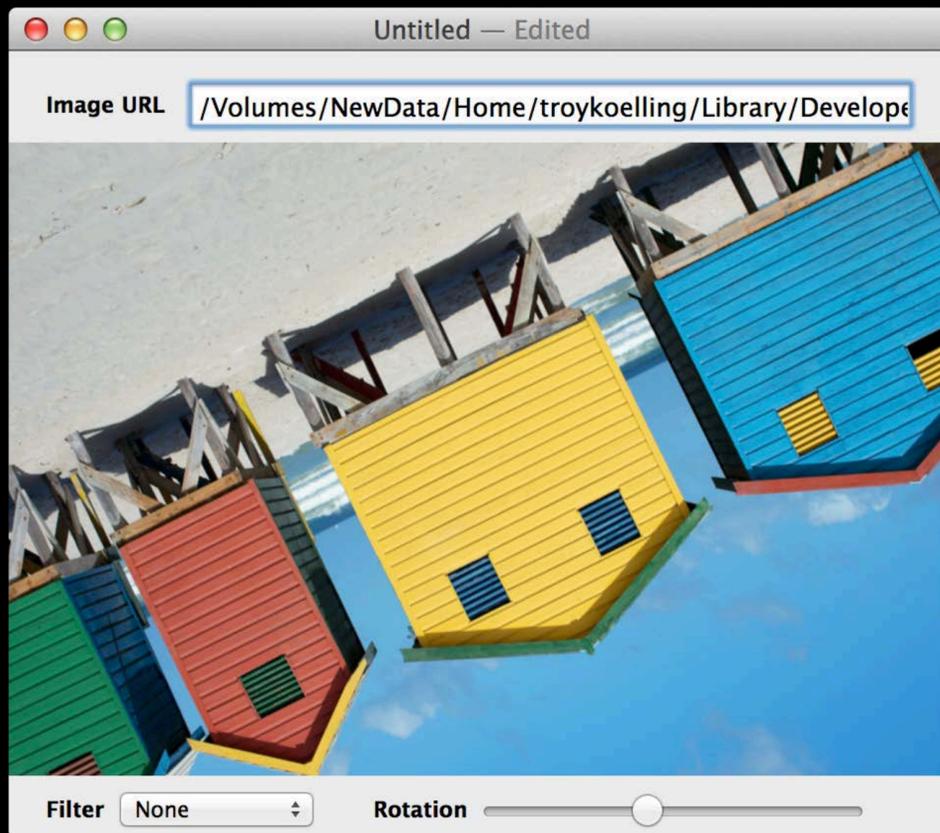
- Show View Frames
- Show Alignment Rectangles
- Show View Drawing
- Show Responsive Scrolling Status



# Drawing Diagnostics (AppKit)



- Show View Frames
- Show Alignment Rectangles
- Show View Drawing
- Show Responsive Scrolling Status



# Printing a UIView's Hierarchy



(lldb)

# Printing a UIView's Hierarchy



```
(lldb) po [self.view recursiveDescription]
```

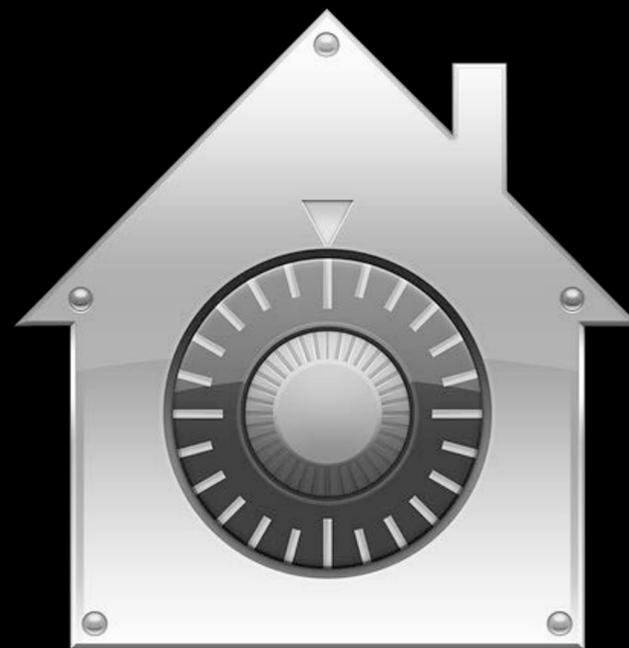
```
(id) $1 = 0x07b55860 <UIView: 0x6e60e90; frame = (20 20; 280 224); auto  
  | <UIRoundedRectButton: 0x6e5f830; frame = (97 90; 86 44); opaque = 1  
  |   | <UIGroupTableViewCellBackground: 0x6e60150; frame = (0 0; 86 44)  
  |   | <UIImageView: 0x6e60b40; frame = (1 1; 84 43); opaque = NO; us  
  |   | <UIButtonLabel: 0x6e60ba0; frame = (12 12; 62 19); text = 'Push'
```

```
(lldb)
```

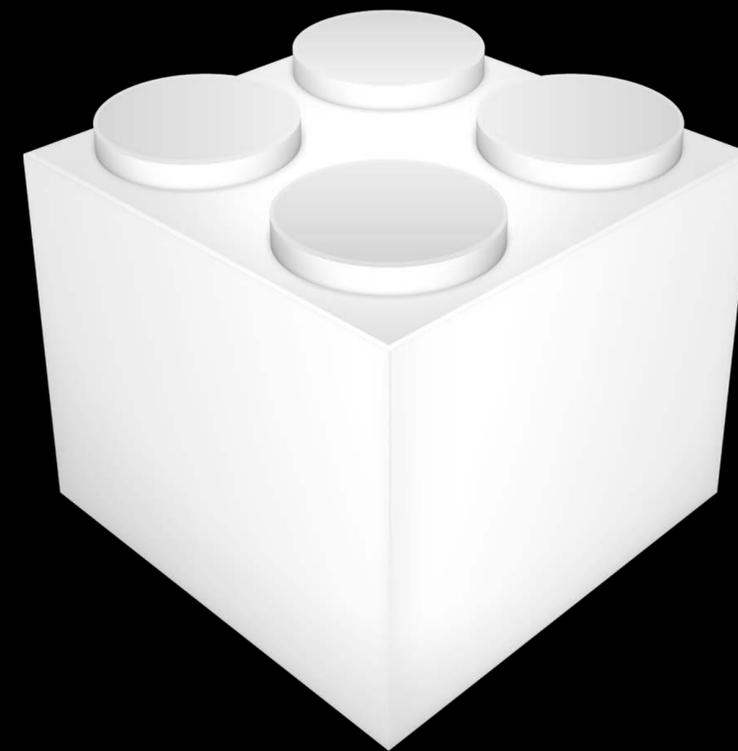
# XPC

## Debugging with Xcode 5

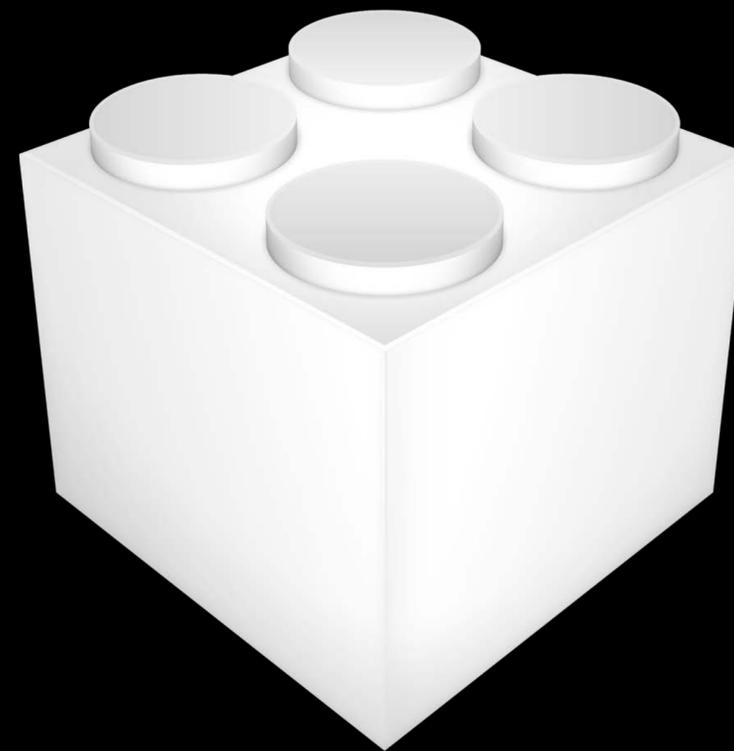
# XPC



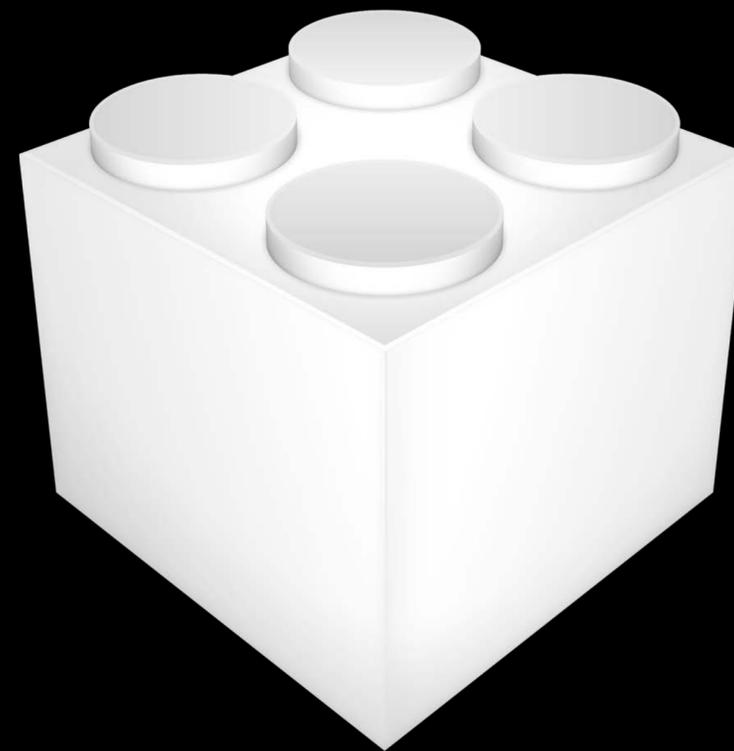
# XPC



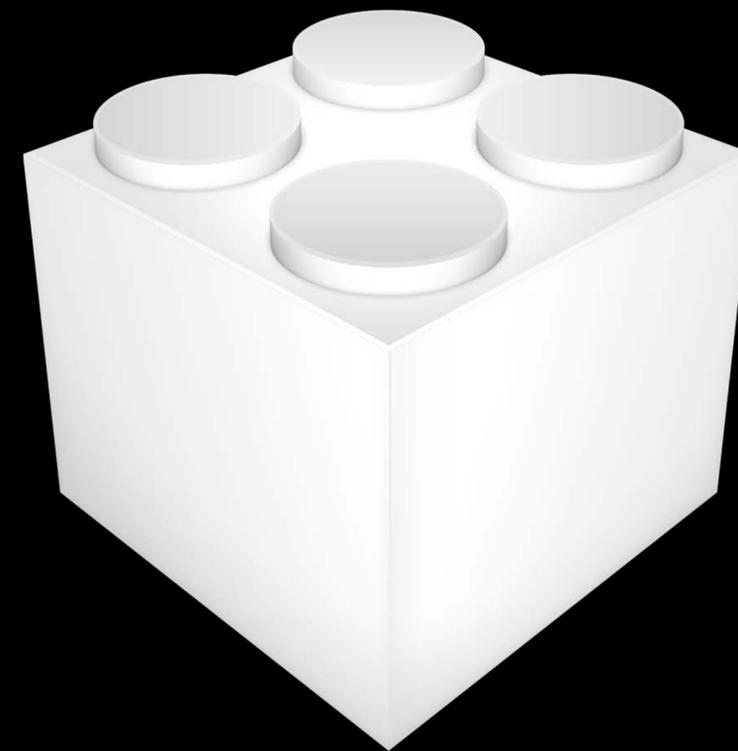
# XPC



# XPC



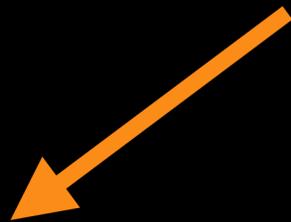
# XPC



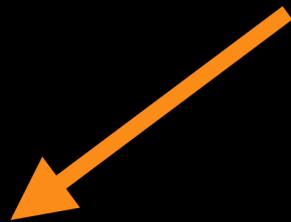
# XPC



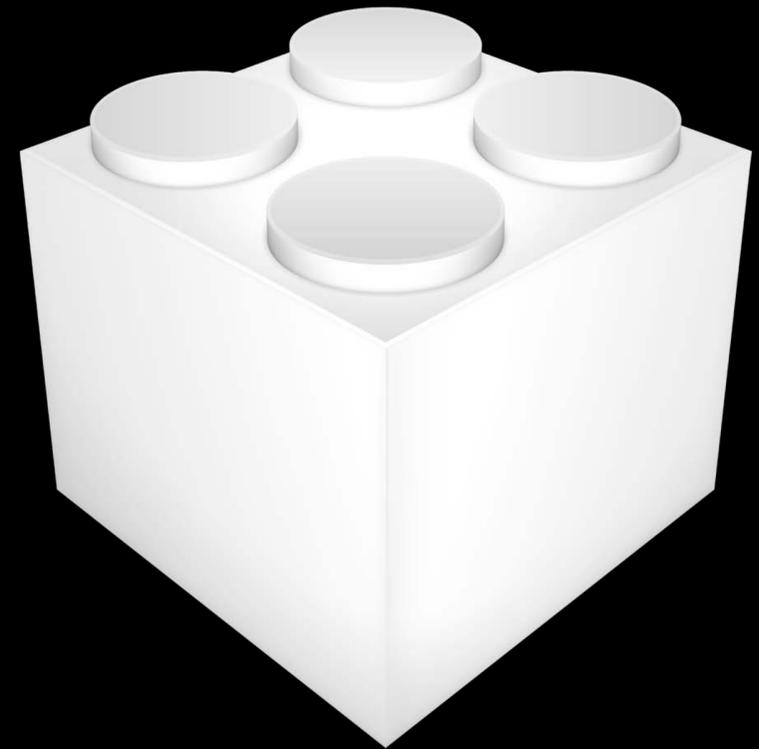
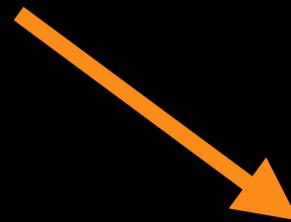
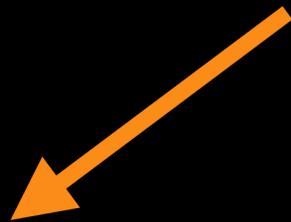
# XPC



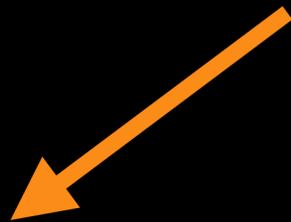
# XPC



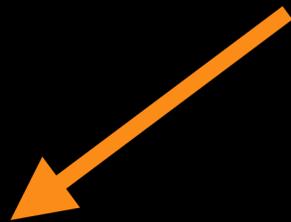
# XPC



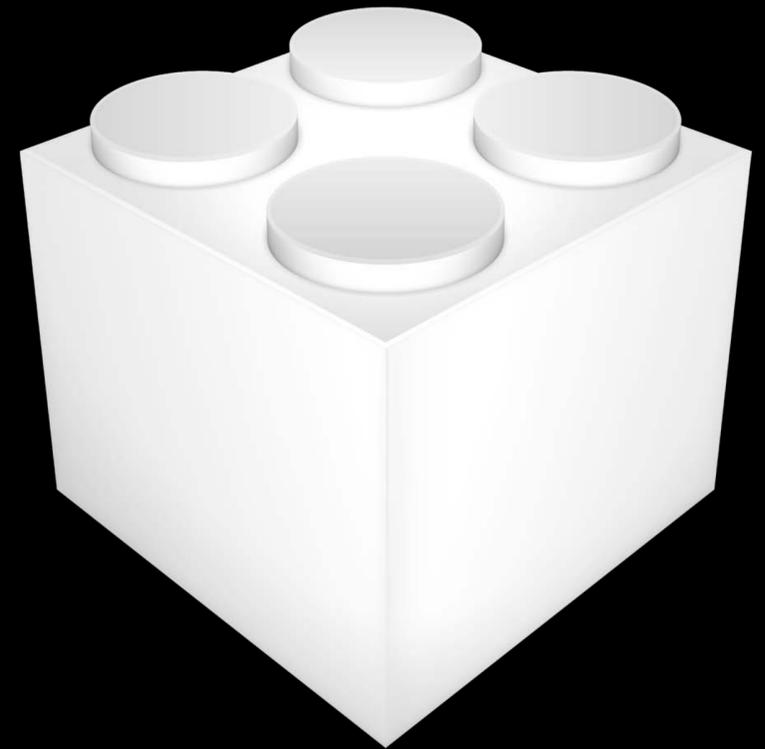
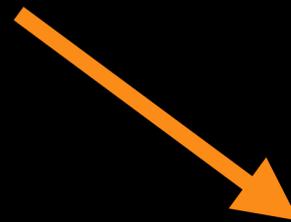
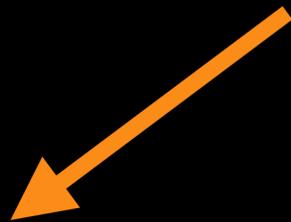
# XPC



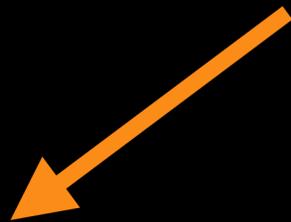
# XPC



# XPC



# XPC



*Demo*

XPC debugging

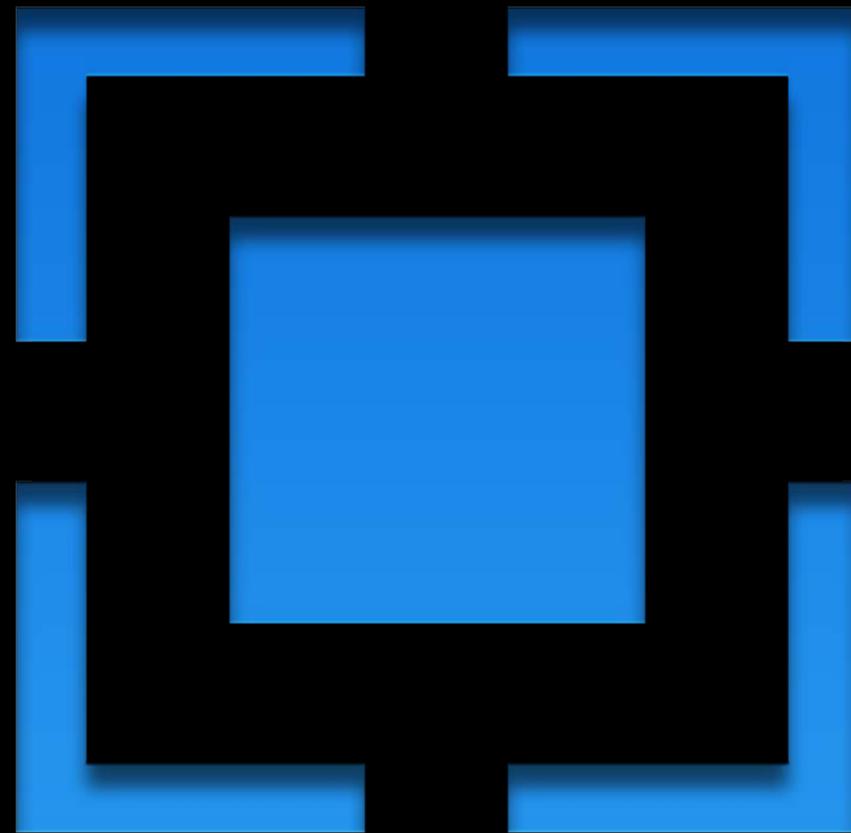
**Christopher Friesen**

Xcode Debugger UI Engineer

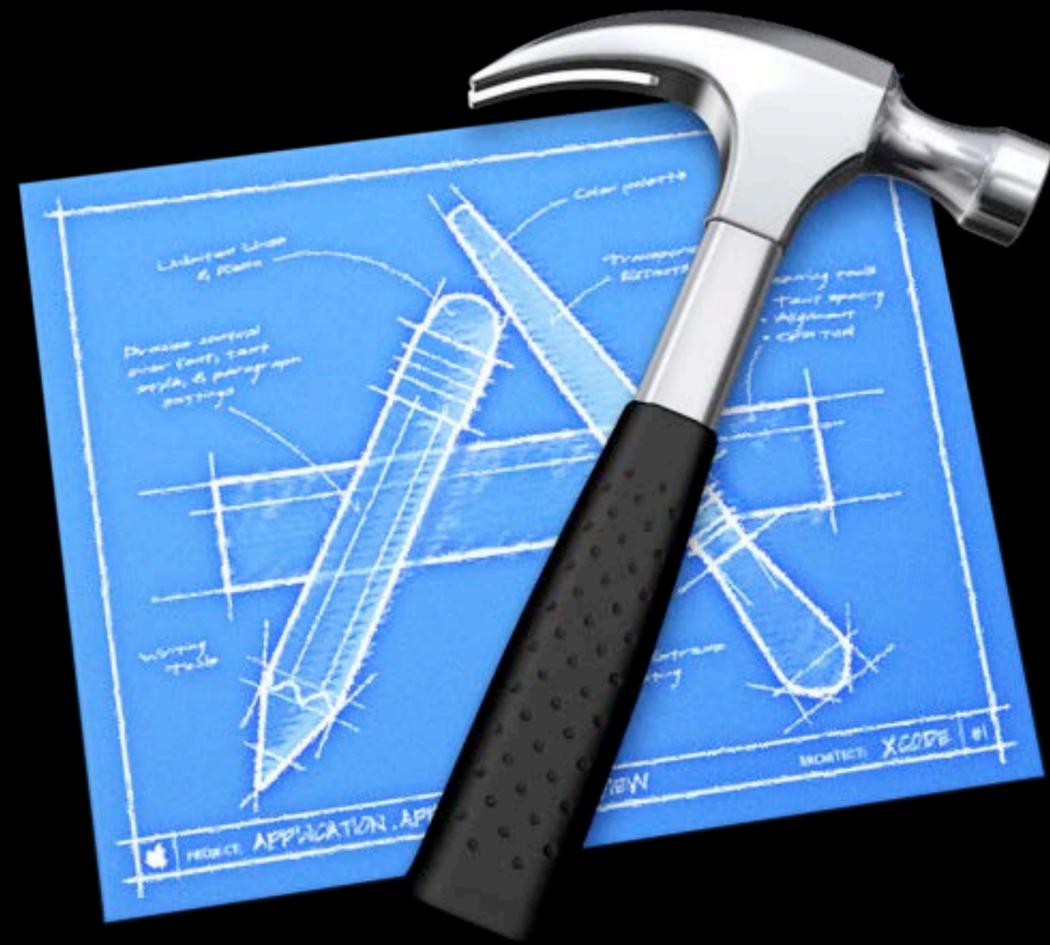












# More Information

## Dave DeLong

App Frameworks and Developer Tools Evangelist  
[delong@apple.com](mailto:delong@apple.com)

## Documentation

Mac OS X Human Interface Guidelines  
<http://developer.apple.com/ue>

## Apple Developer Forums

<http://devforums.apple.com>

# Related Sessions

Efficient Design with XPC

Russian Hill  
Tuesday 2:00PM

Advanced Debugging with LLDB

Pacific Heights  
Friday 9:00AM

# Labs

Xcode Lab	Tools Lab A Thursday 2:00PM	
Power and Performance for OS X Apps Lab	Tools Lab A Thursday 4:30PM	
Xcode Lab	Tools Lab A Friday 9:00AM	
LLDB and Instruments Lab	Tools Lab C Friday 10:15AM	

 WWDC2013