

# Debugging in Xcode

Session 407

Troy Koelling  
Xcode Debugger UI Engineer

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

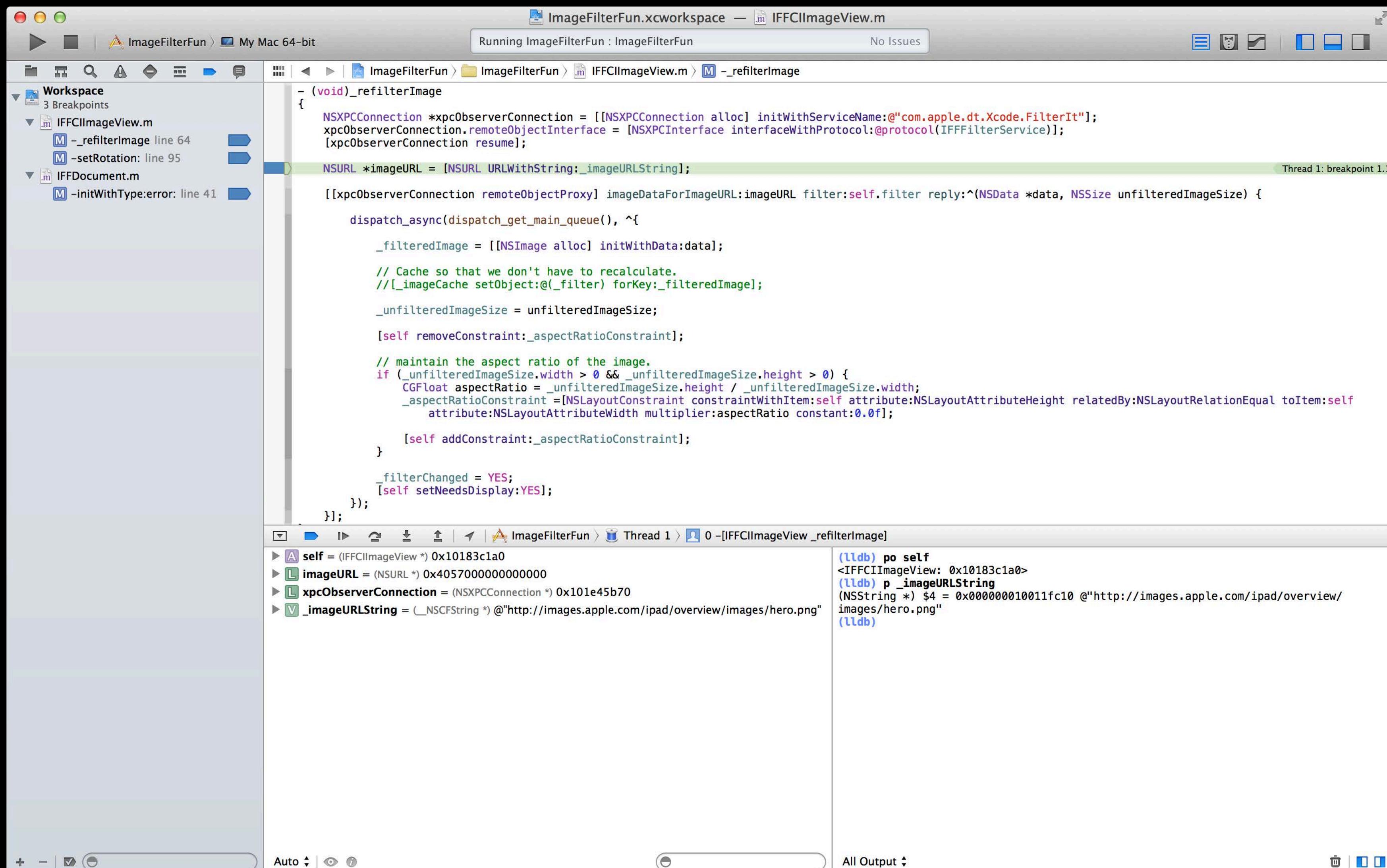


“Everyone knows that **debugging** is **twice as hard** as writing a program in the first place. So if you’re as clever as you can be when you write it, how will you ever debug it?”

Brian Kernighan

# Debugging in Xcode 5

# Debugging in Xcode 5



ImageFilterFun.xcworkspace —

Running ImageFilterFun : ImageFilterFun

ImageFilterFun > My Mac 64-bit

Workspace  
3 Breakpoints

IFFCIIImageView.m  
M -\_refilterImage line 64  
M -setRotation: line 95

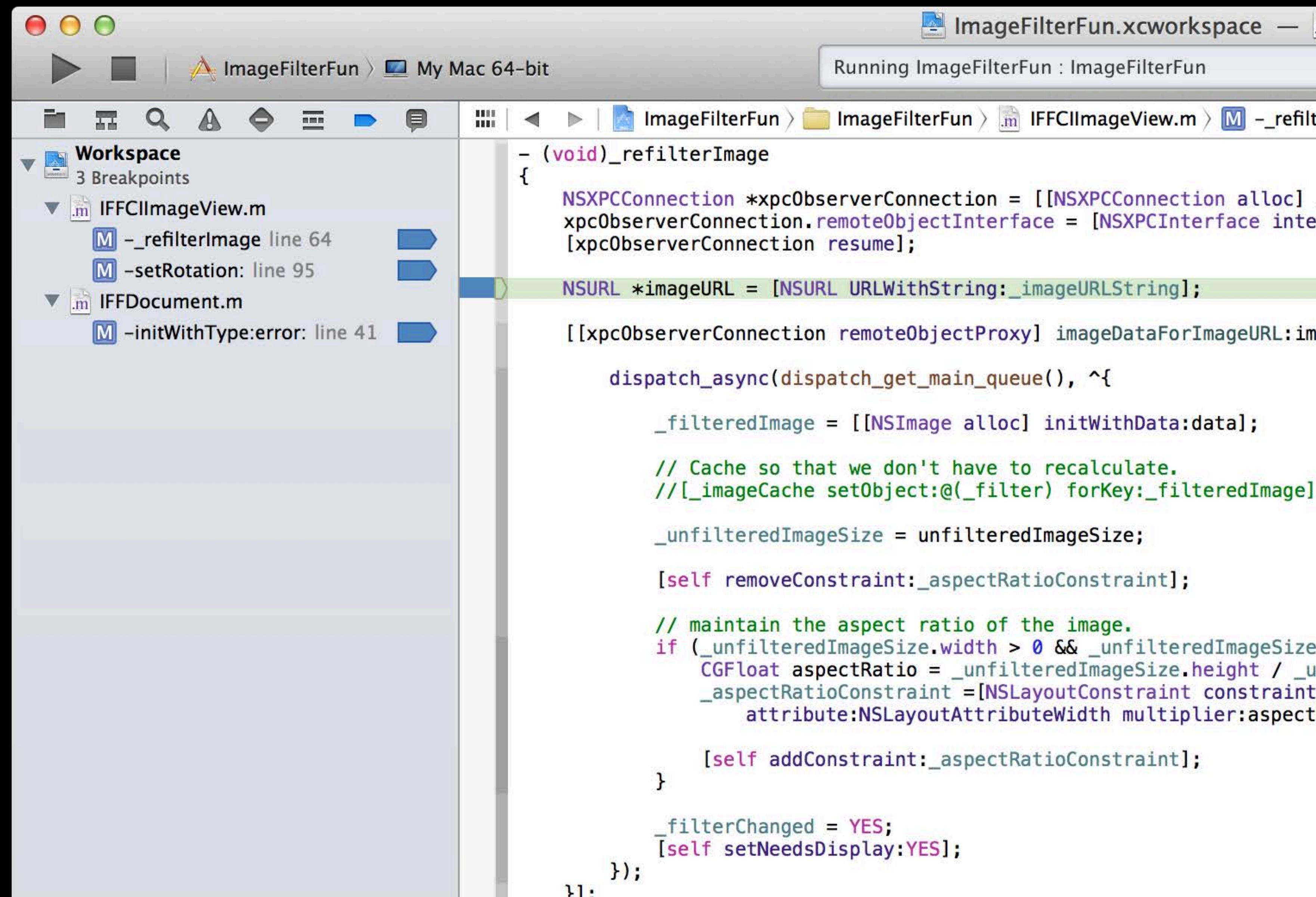
IFFDocument.m  
M -initWithType:error: line 41

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

    NSURL *imageURL = [NSURL URLWithString:_imageURLString];

    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL
dispatch_async(dispatch_get_main_queue(), ^{
        _filteredImage = [[NSImage alloc] initWithData:[imageURL 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:_filteredImage
attribute:NSLayoutAttributeWidth multiplier:aspectRatio
constant:(unfilteredImageSize.width - _unfilteredImageSize.width * aspectRatio) / 2];
            [self addConstraint:_aspectRatioConstraint];
        }
        _filterChanged = YES;
        [self setNeedsDisplay:YES];
    });
}
```

# Breakpoints Navigator



M -initWithType:error: line 41

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

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

(lldb) po self  
<IFFCIIImageView: 0x10183c1a0>  
(lldb) p \_imageURLString  
(NSString \*) \$4 = 0x00000010011fc10 @"http://  
images/hero.png"  
(lldb)

Auto |

All Output

M -initWithType:error: line 41

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

File ▶ Run ▶ Stop ▶ □ |  ImageFilterFun ▶ Thread 1 ▶  0 -[IFFCIIImageView \_refilterImage]

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

(lldb) po self  
<IFFCIIImageView: 0x10183c1a0>  
(lldb) p \_imageURLString  
(NSString \*) \$4 = 0x00000010011fc10 @"http://  
images/hero.png"  
(lldb)

Auto ▾ |  

All Output ▾

M -initWithType:error: line 41

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

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

(lldb) po self  
<IFFCIIImageView: 0x10183c1a0>  
(lldb) p \_imageURLString  
(NSString \*) \$4 = 0x00000010011fc10 @"http://  
images/hero.png"  
(lldb)

Auto |

All Output

M -initWithType:error: line 41

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

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

(lldb) po self  
<IFFCIIImageView: 0x10183c1a0>  
(lldb) p \_imageURLString  
(NSString \*) \$4 = 0x00000010011fc10 @"http://  
images/hero.png"  
(lldb)

Auto | ⚡

All Output

M -initWithType:error: line 41

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

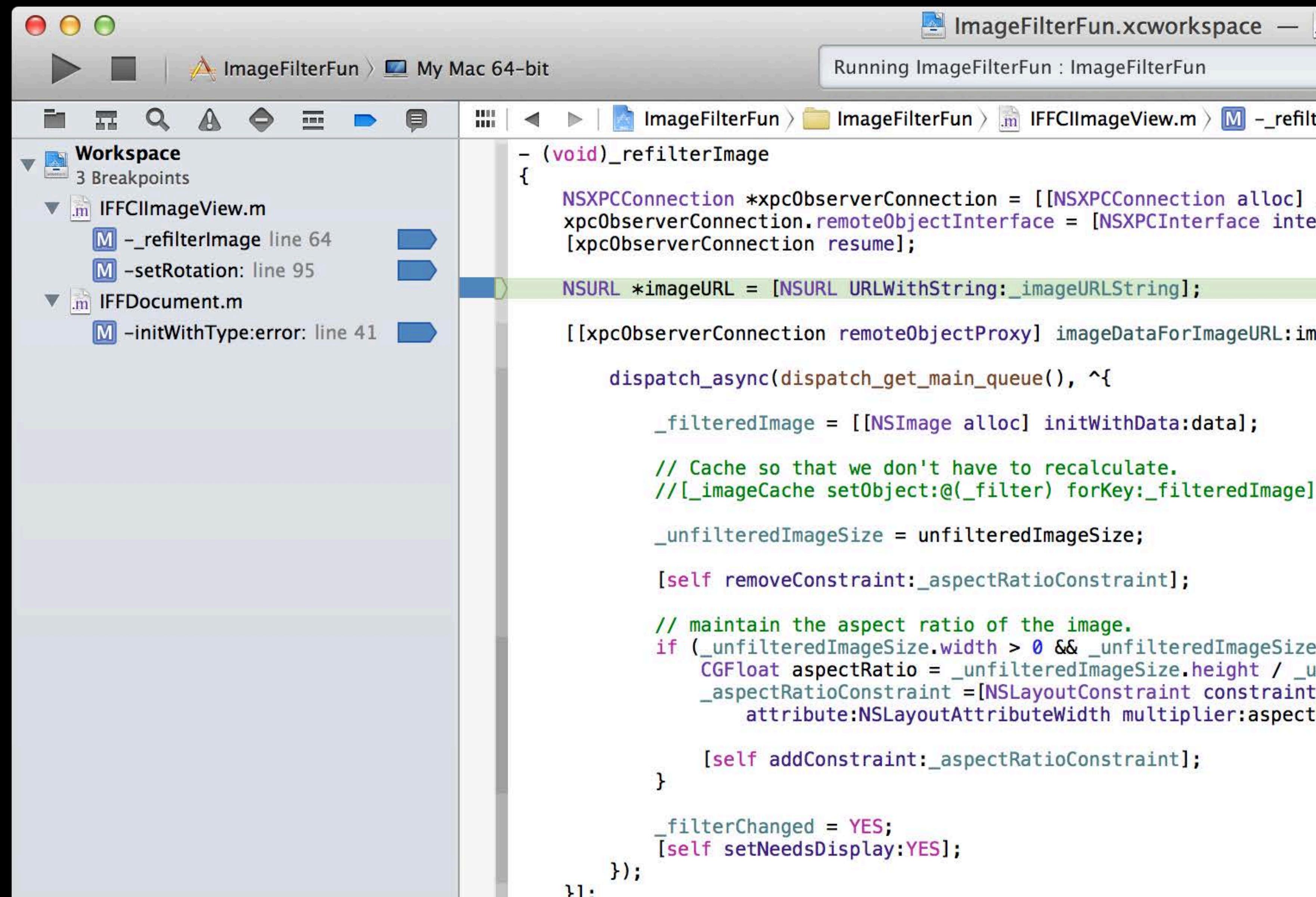
▶ A self = (IFFCIIImageView \*) 0x10183c1a0  
▶ L imageURL = (NSURL \*) 0x4057000000000000  
▶ L xpcObserverConnection = (NSXPCCConnection \*) 0x101e45b70  
▶ V \_imageURLString = (\_NSCFString \*) @"http://images.apple.com/ipad/overview/images/hero.png"

(lldb) po self  
<IFFCIIImageView: 0x10183c1a0>  
(lldb) p \_imageURLString  
(NSString \*) \$4 = 0x00000010011fc10 @"http://  
images/hero.png"  
(lldb)

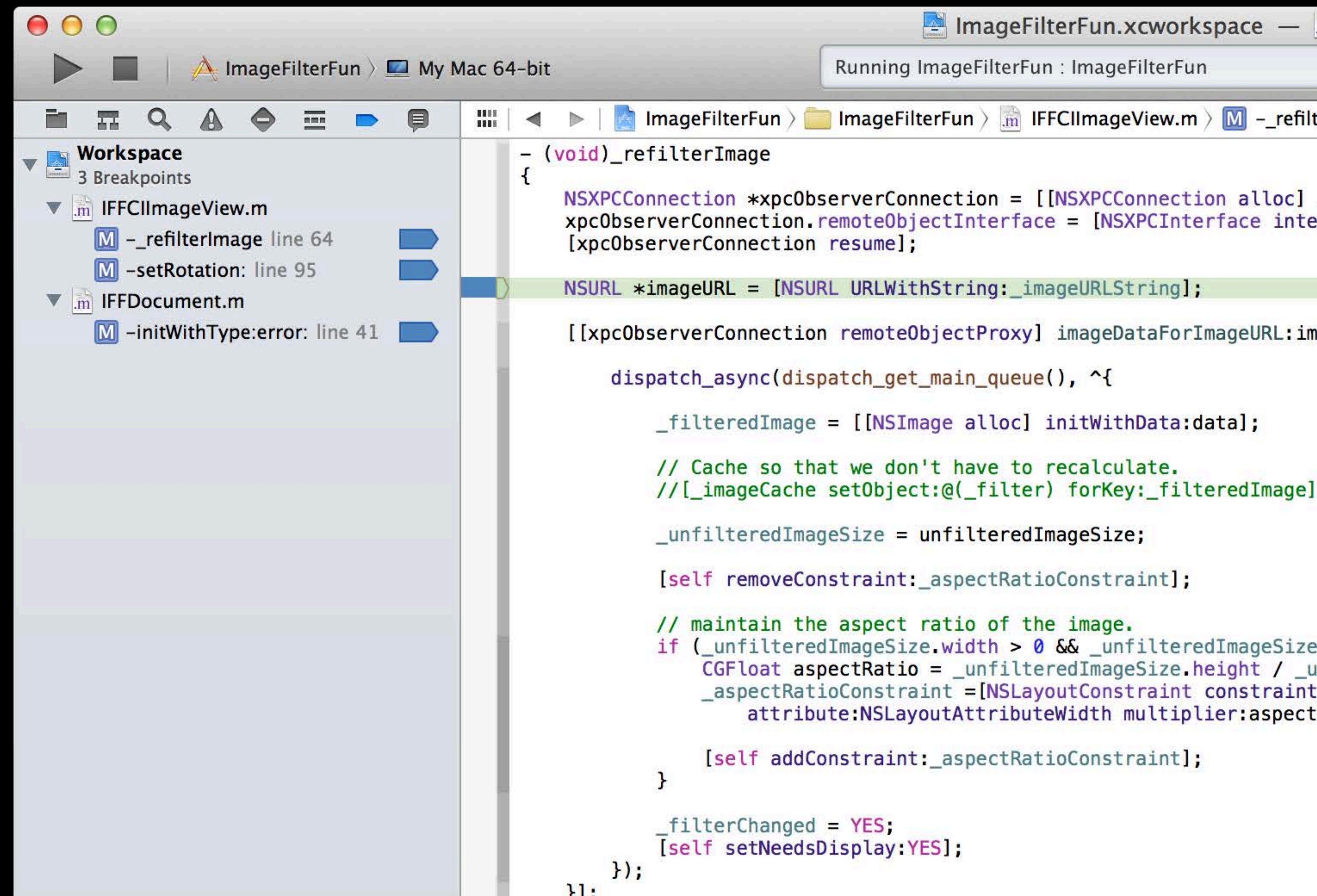
Auto |

All Output

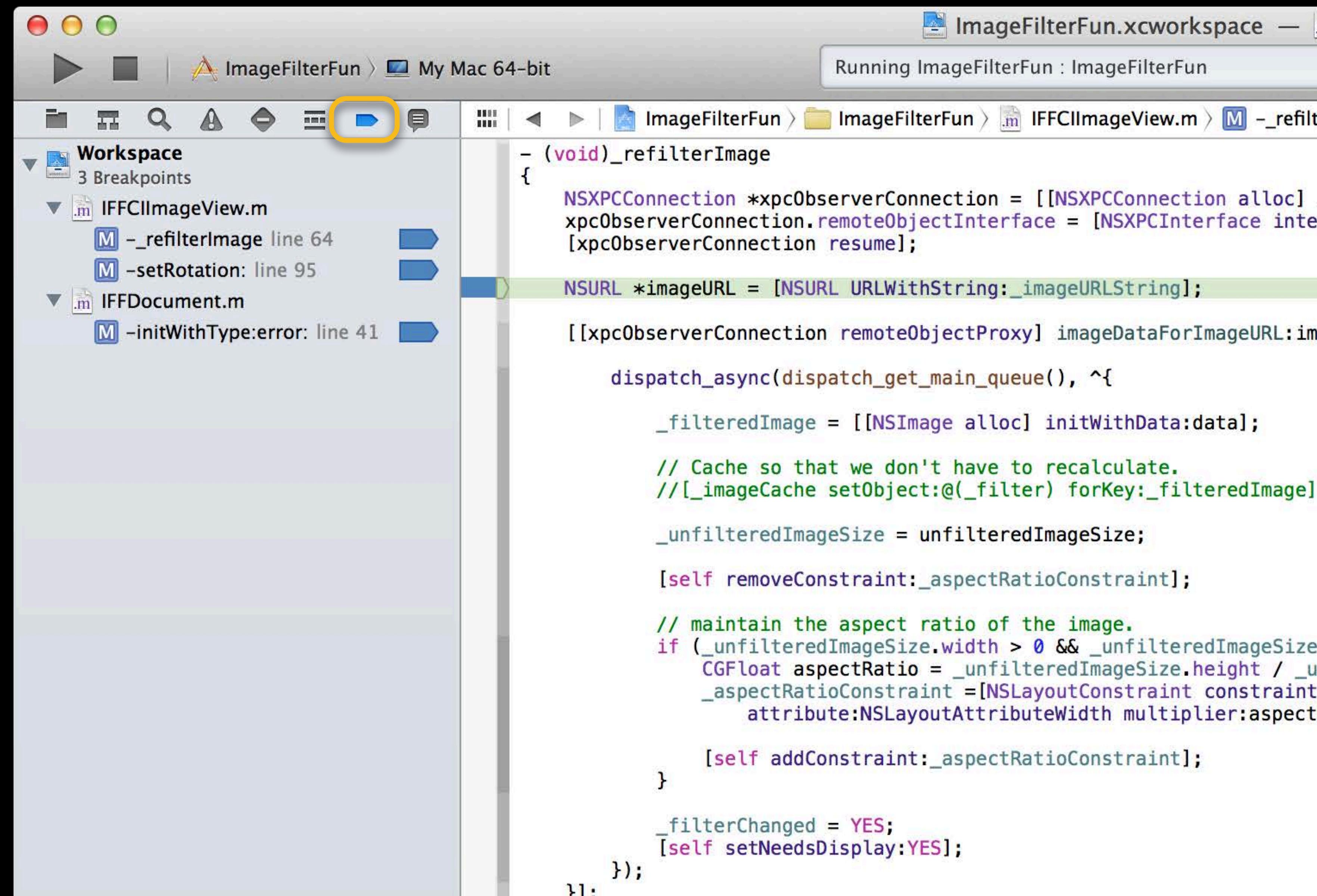
# Breakpoints Navigator



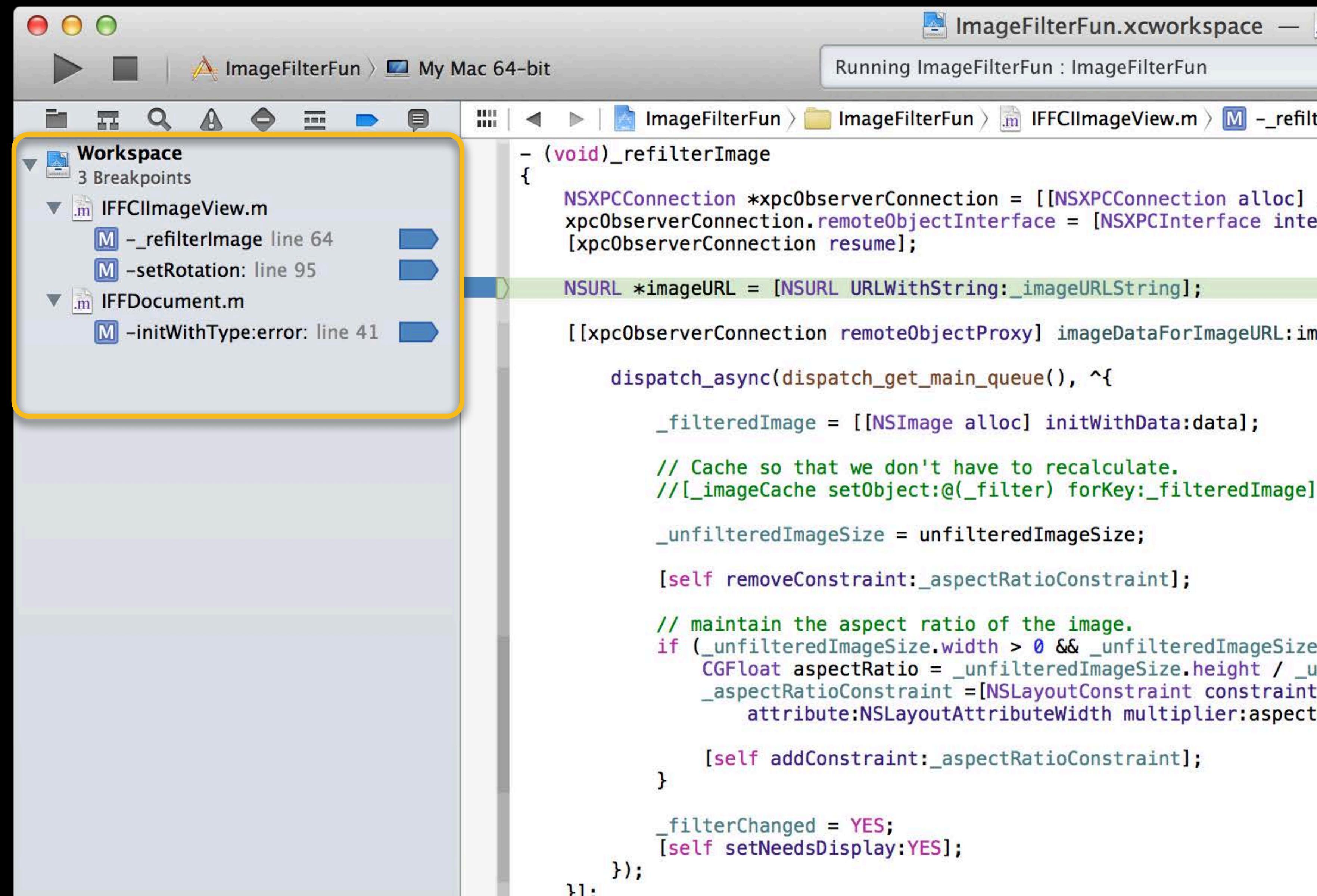
# Breakpoints Navigator



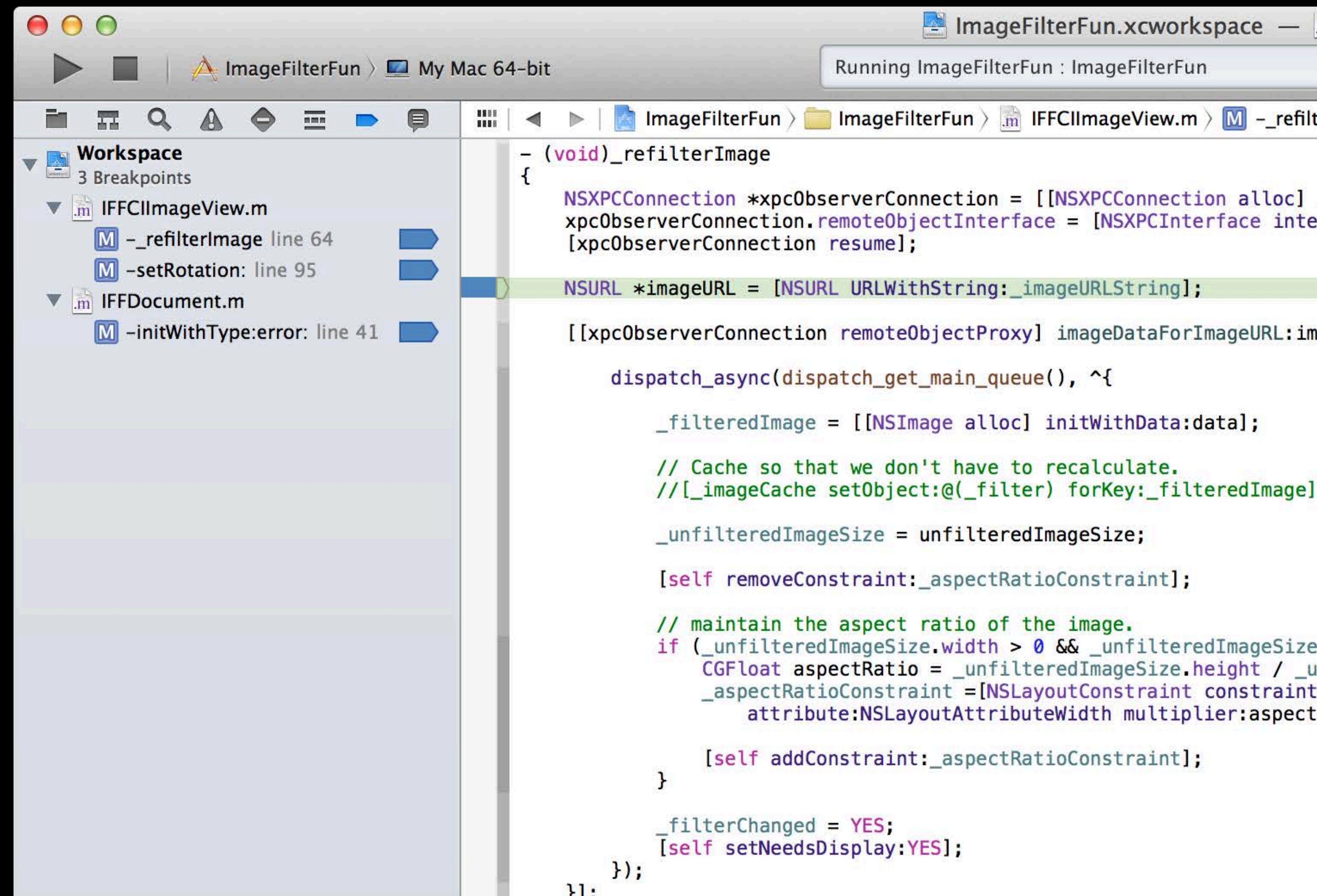
# Breakpoints Navigator



# Breakpoints Navigator



# Breakpoints Navigator



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

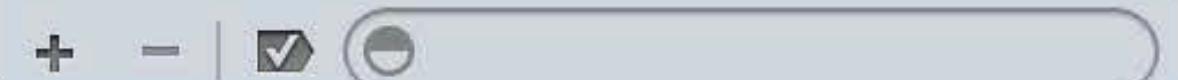
[ ] ▶ ▶ ▶ ▶ ▶ ▶ | ↻ | ↺ | ImageFilterFun > Thread 1 > 0 -[IFFCII

▶ A self = (IFFCIIimageView \*) 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];
});
}];
```

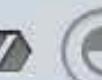
[ ] ▶ ▶ ▶ | ↴ | ↵ | ↲ | ↳ |  ImageFilterFun >  Thread 1 >  0 -[IFFCII

▶ A self = (IFFCIIimageView \*) 0x10183c1a0

▶ L imageURL = (NSURL \*) 0x4057000000000000

▶ L xpcObserverConnection = (NSXPCCConnection \*) 0x101e45b70

▶ V \_imageURLString = (\_NSCFString \*) @"http://images.apple.com/ipad/overview/image



Auto |  



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

[ □ ▶ ▶ ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ | ↗ | A ImageFilterFun > ⚒ Thread 1 > 0 -[IFFCII

► A self = (IFFCIIimageView \*) 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];
});
}];
```

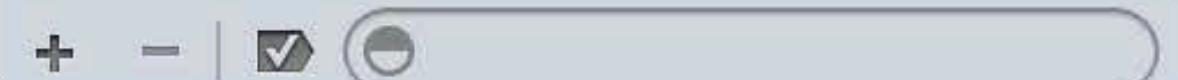
[ ] ImageFilterFun > Thread 1 > 0 -[IFFCII

► A self = (IFFCIIimageView \*) 0x10183c1a0

► L imageURL = (NSURL \*) 0x4057000000000000

► L xpcObserverConnection = (NSXPCConnection \*) 0x101e45b70

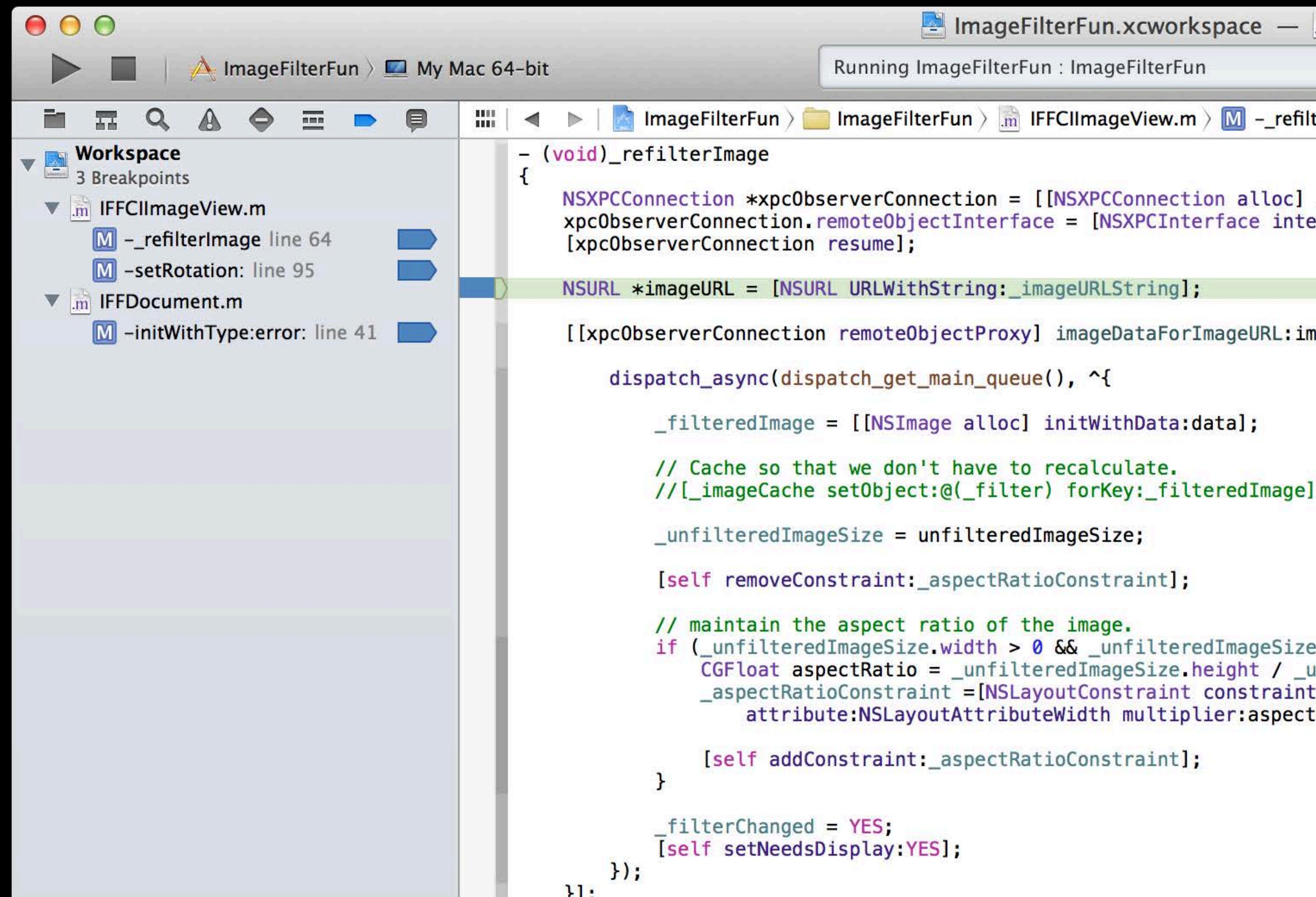
► V \_imageURLString = (\_NSCFString \*) @"http://images.apple.com/ipad/overview/image



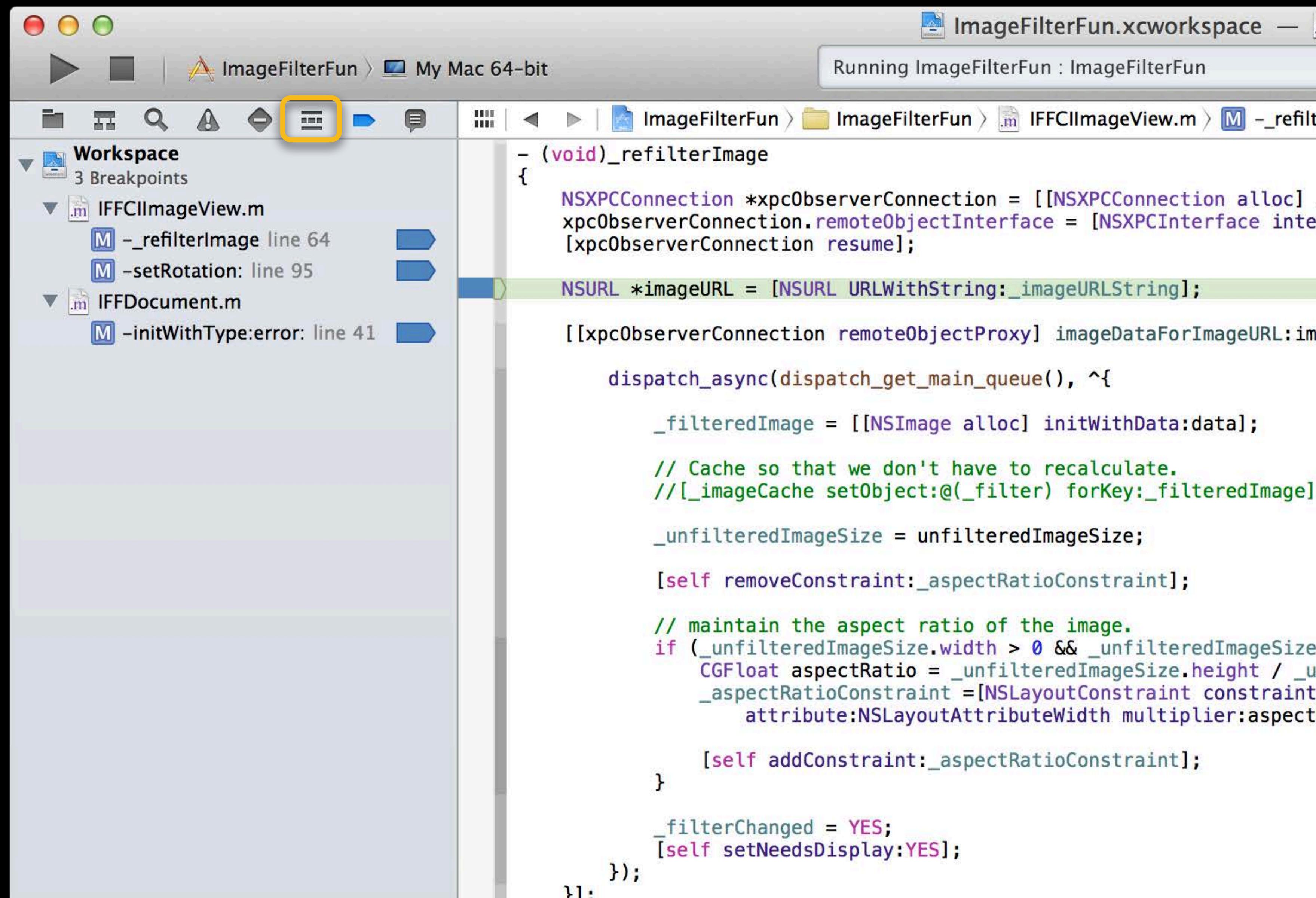
Auto |



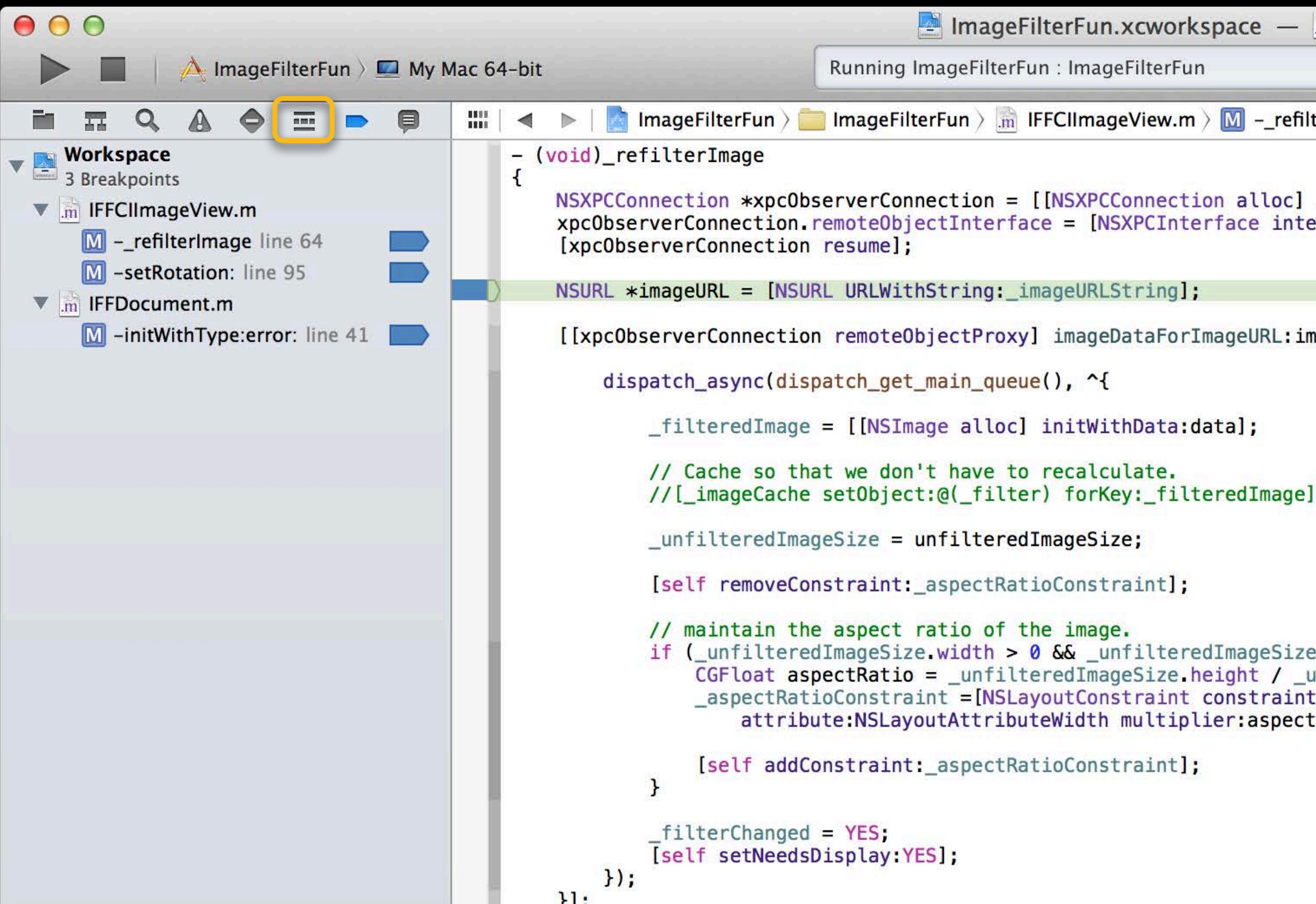
# Breakpoints Navigator



# Breakpoints Navigator



# Debug Navigator



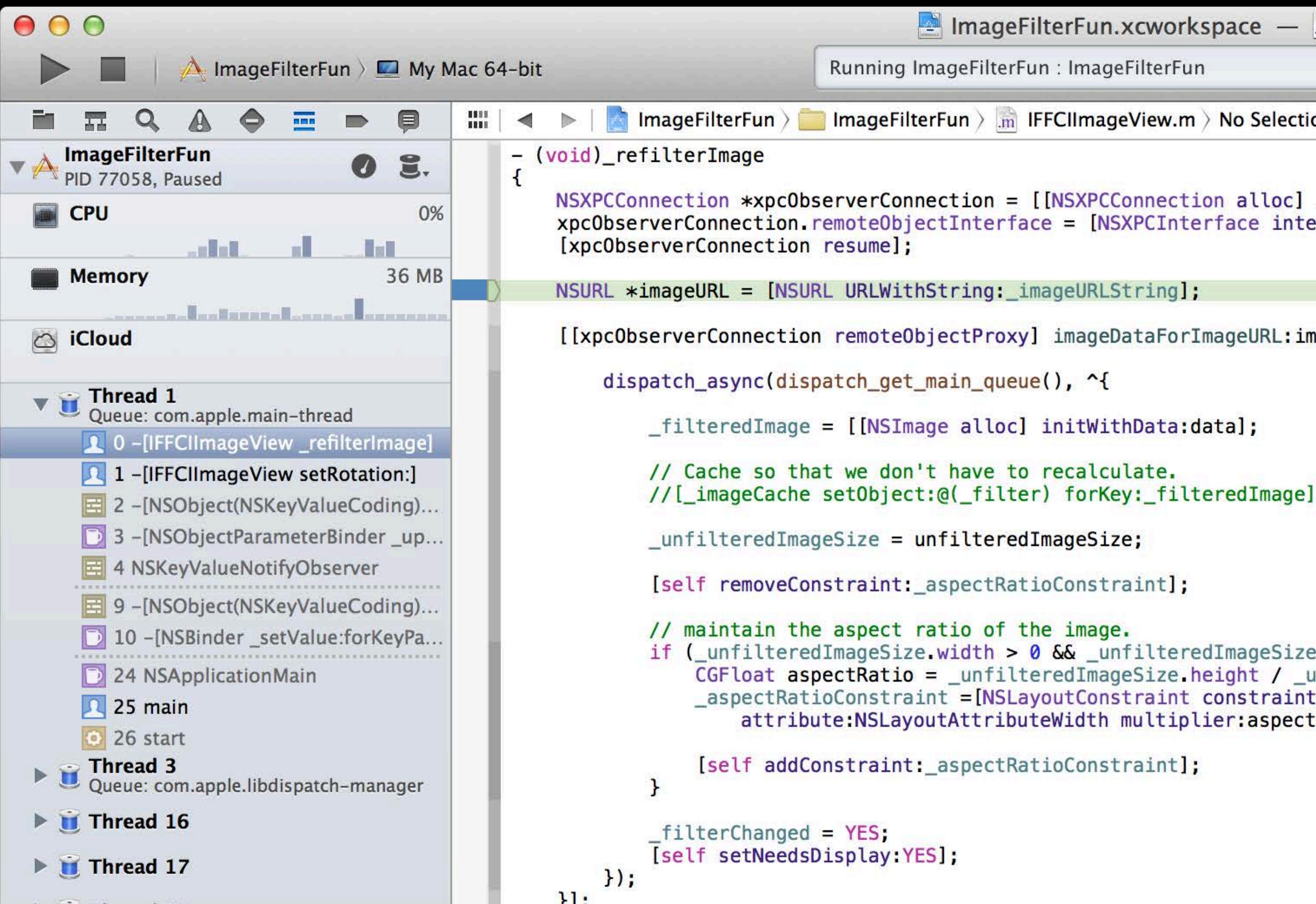
# Debug Navigator

The screenshot shows the Xcode interface with the title bar "ImageFilterFun.xcworkspace — ImageFilterFun" and "Running ImageFilterFun : ImageFilterFun". The Debug Navigator is open, displaying a stack trace. The current frame is in `IFFCIIImageView.m` at line 64, with the method `-(_refilterImage)`. The stack trace continues through `IFFDocument.m` and `ImageFilterFun`, ending at the entry point `My Mac 64-bit`.

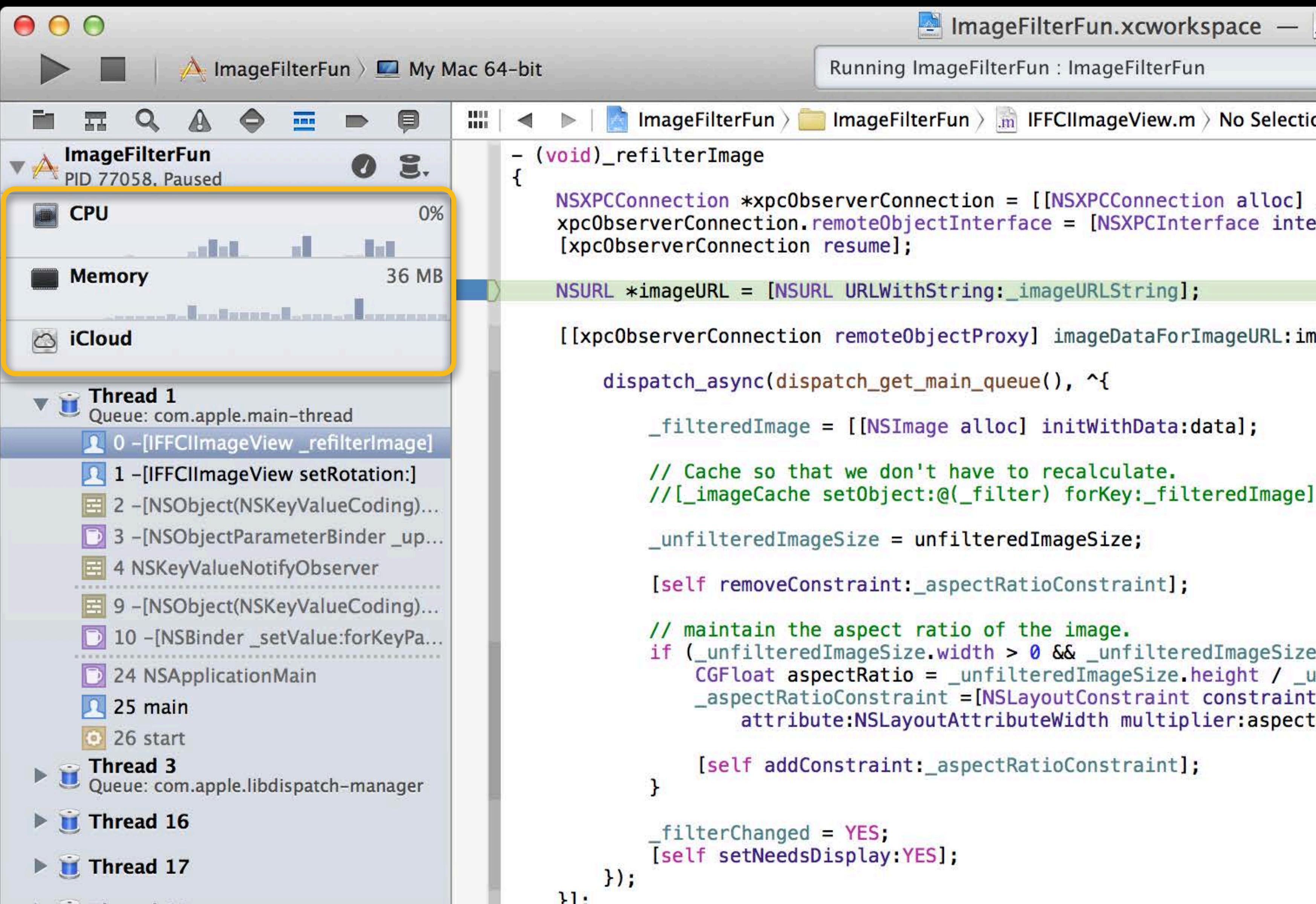
```
- (void)_refilterImage
{
    NSXPCConnection *xpcObserverConnection = [[NSXPCConnection alloc]
xpcObserverConnection.remoteObjectInterface = [NSXPCInterface interfaceWithProtocol:@[xpcObserverConnection resume]];

    NSURL *imageURL = [NSURL URLWithString:_imageURLString];
    [[xpcObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL
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.height > 0)
            CGFloat aspectRatio = _unfilteredImageSize.height / _unfilteredImageSize.width;
            _aspectRatioConstraint = [NSLayoutConstraint constraintWithItem:self attribute:NSLayoutAttributeWidth multiplier:aspectRatio constant:0];
            [self addConstraint:_aspectRatioConstraint];
    }];
    _filterChanged = YES;
    [self setNeedsDisplay:YES];
});}
```

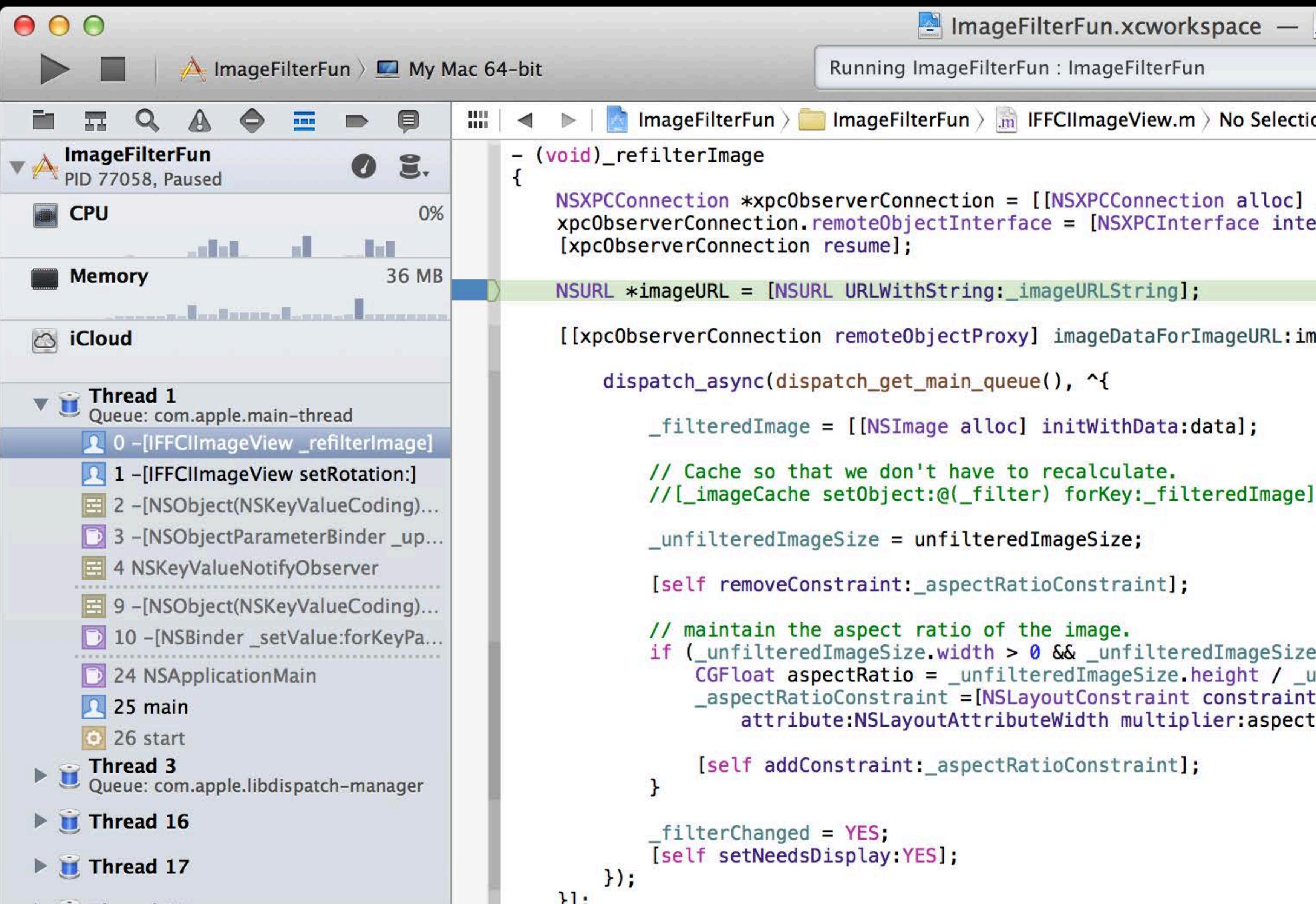
# Debug Navigator



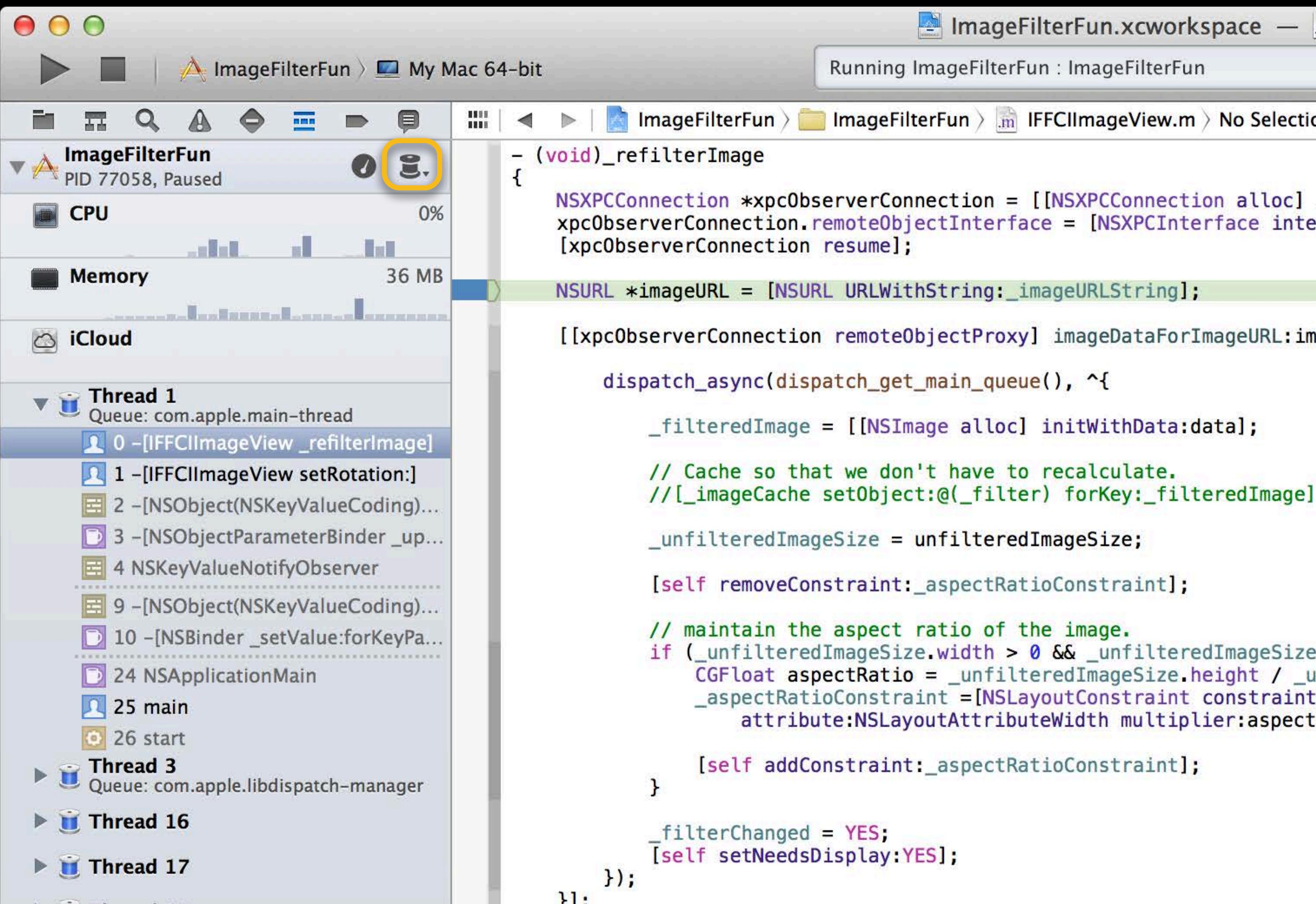
# Debug Navigator



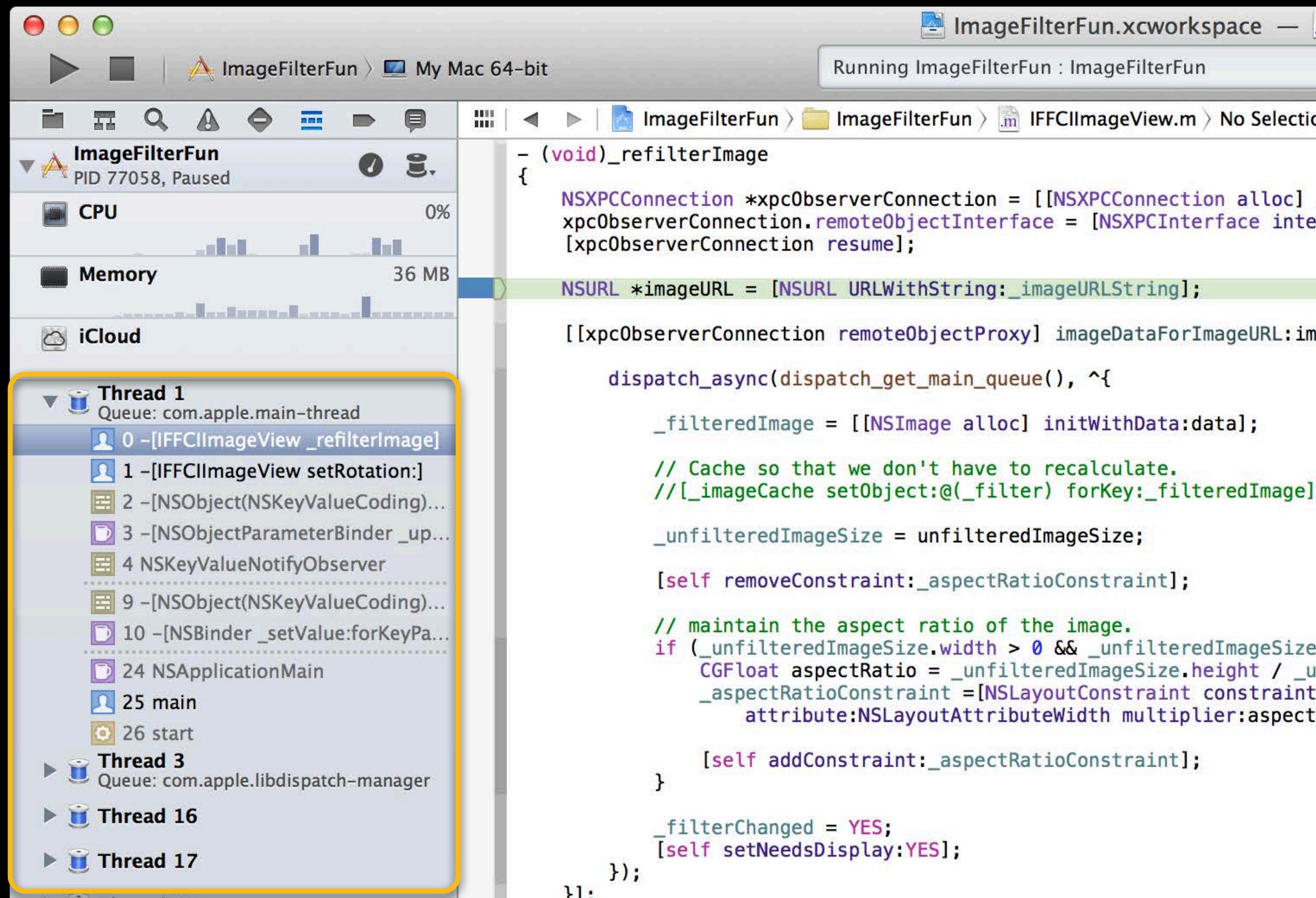
# Debug Navigator



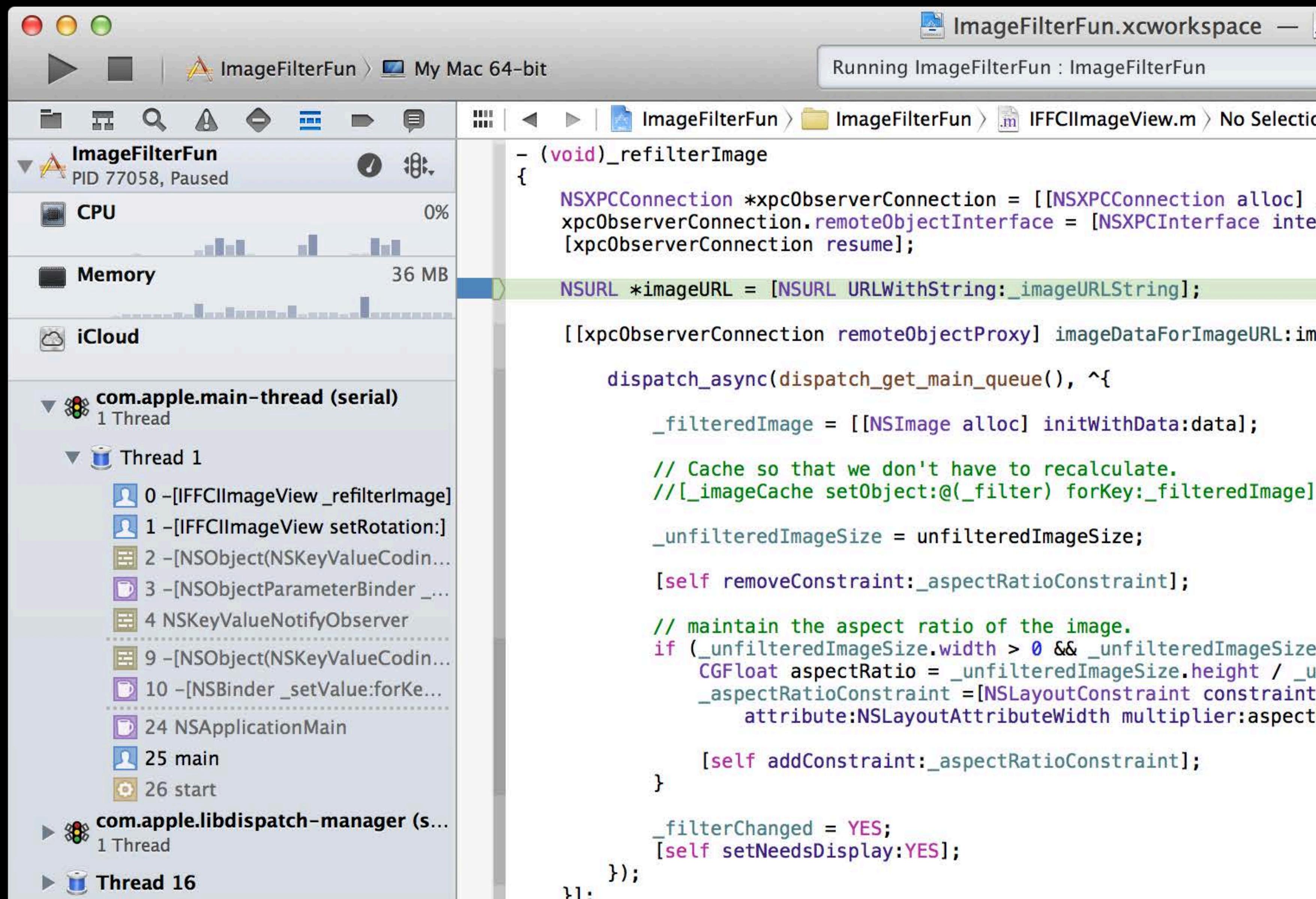
# Debug Navigator



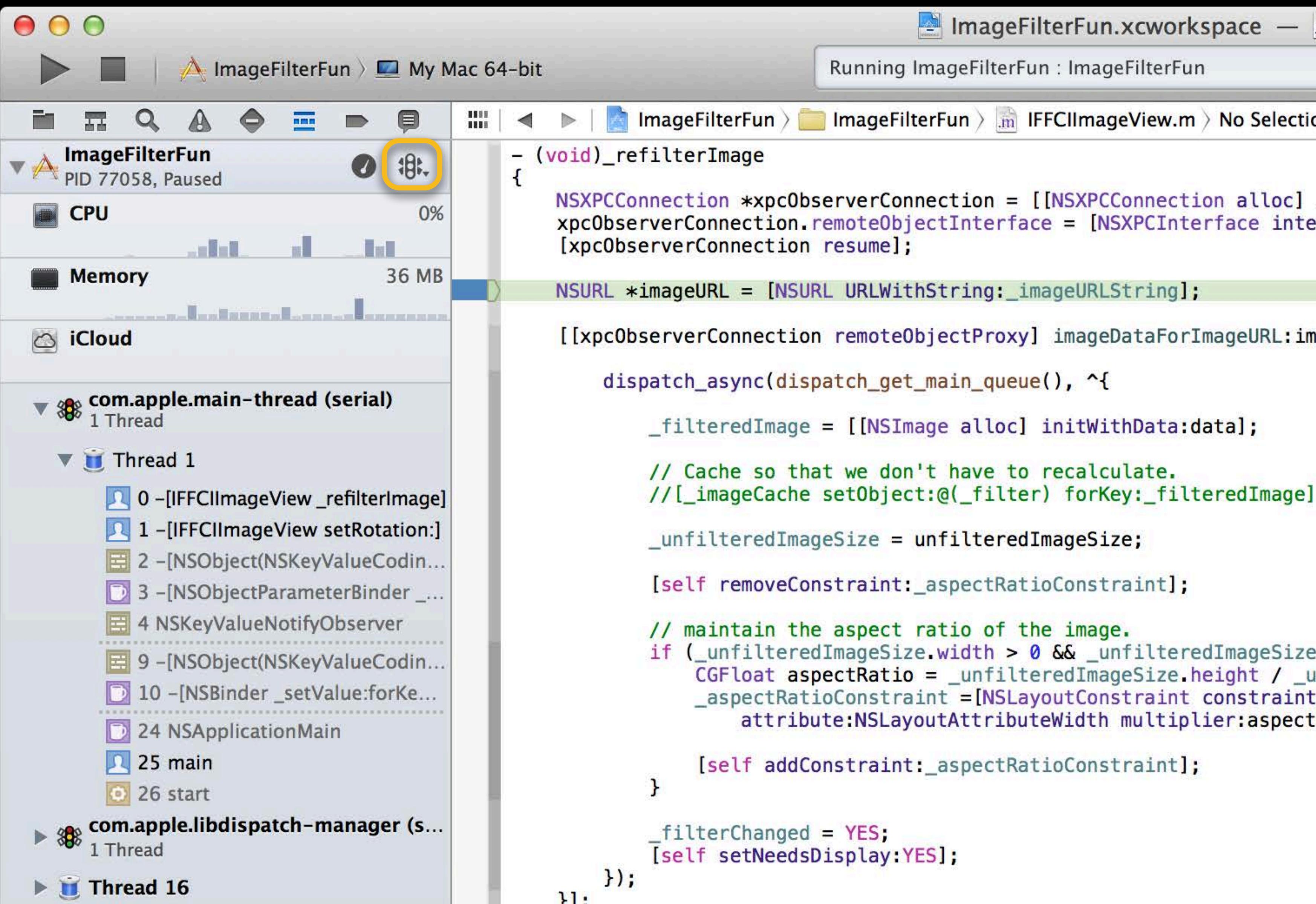
# Debug Navigator



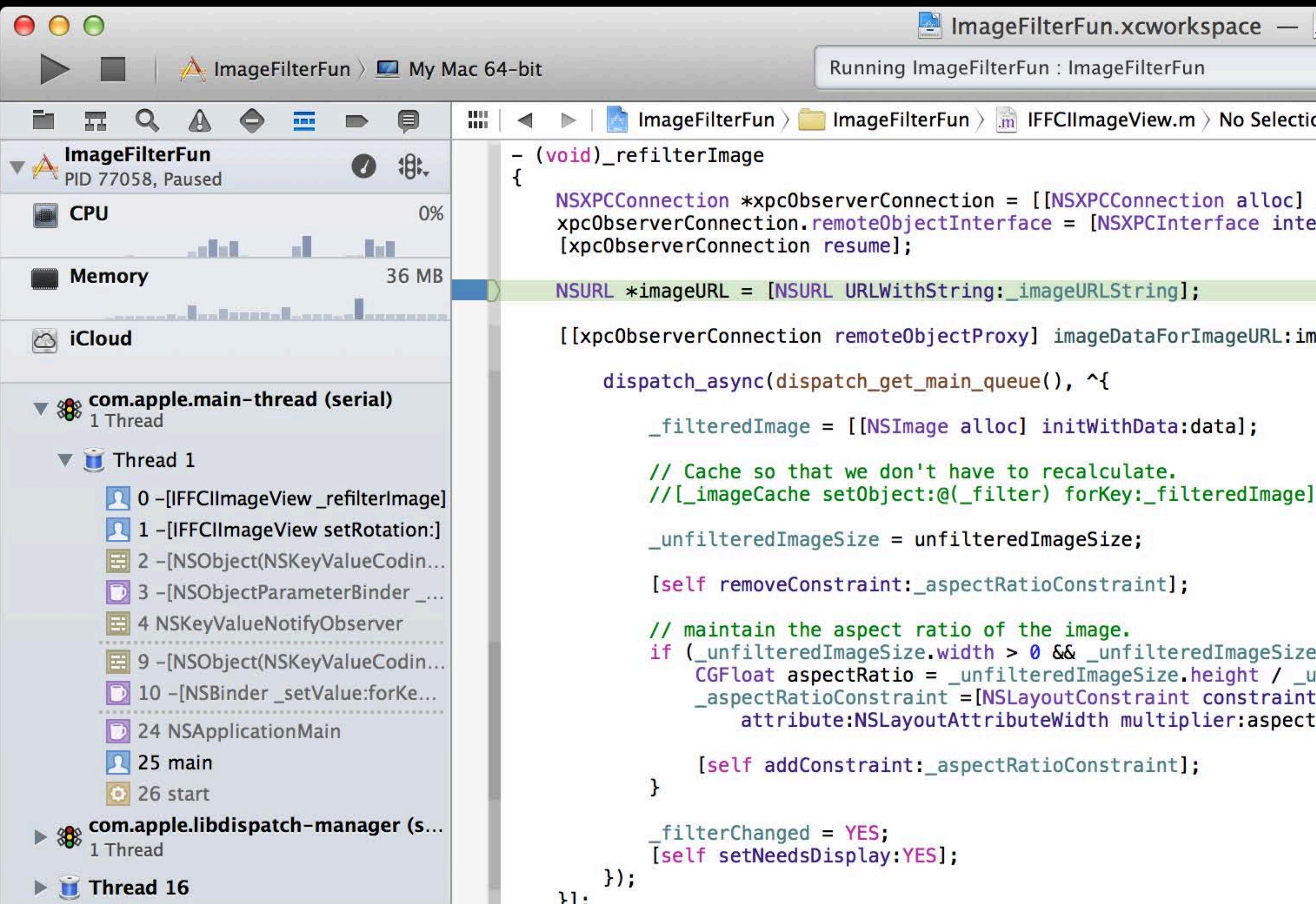
# Debug Navigator



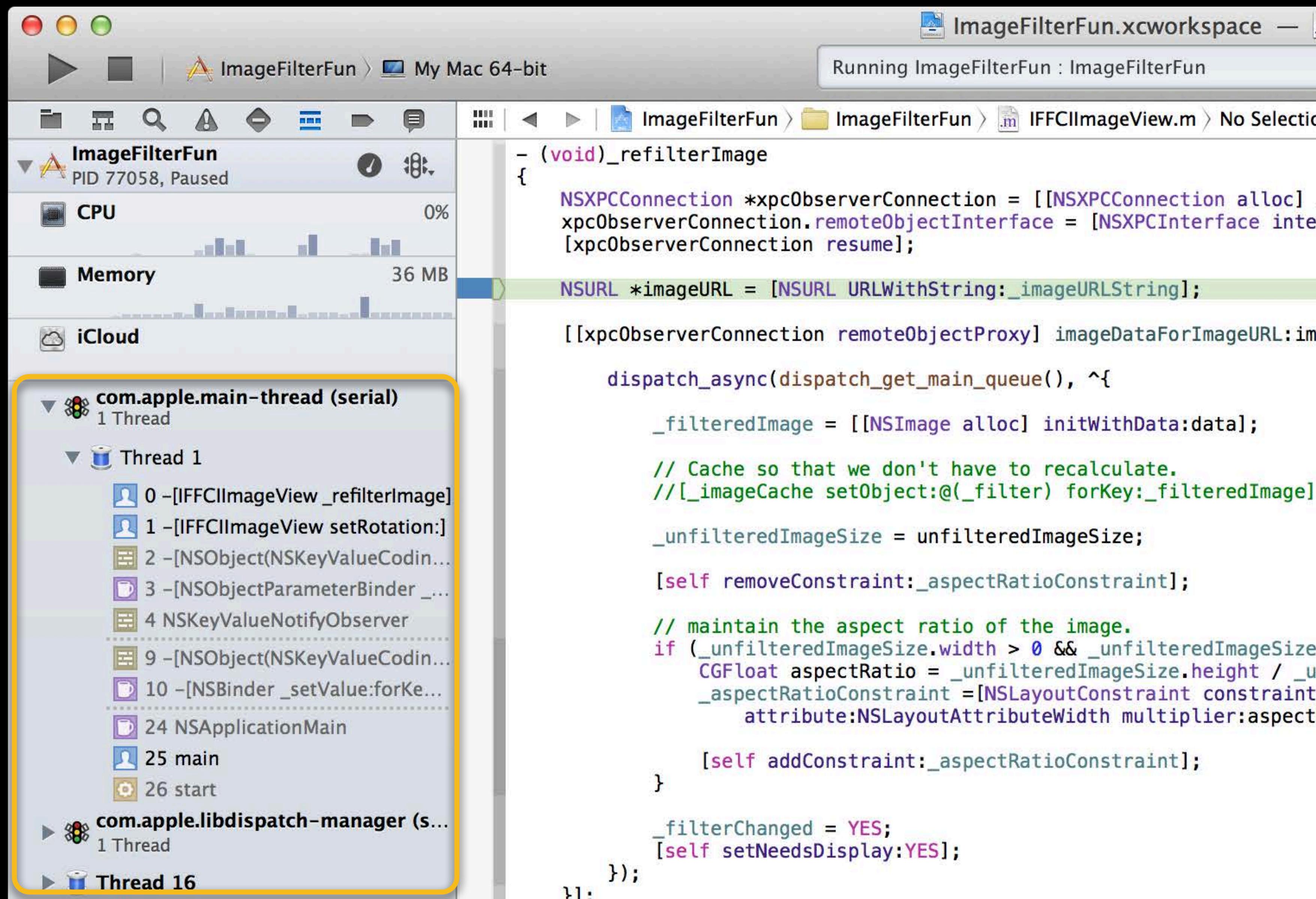
# Debug Navigator



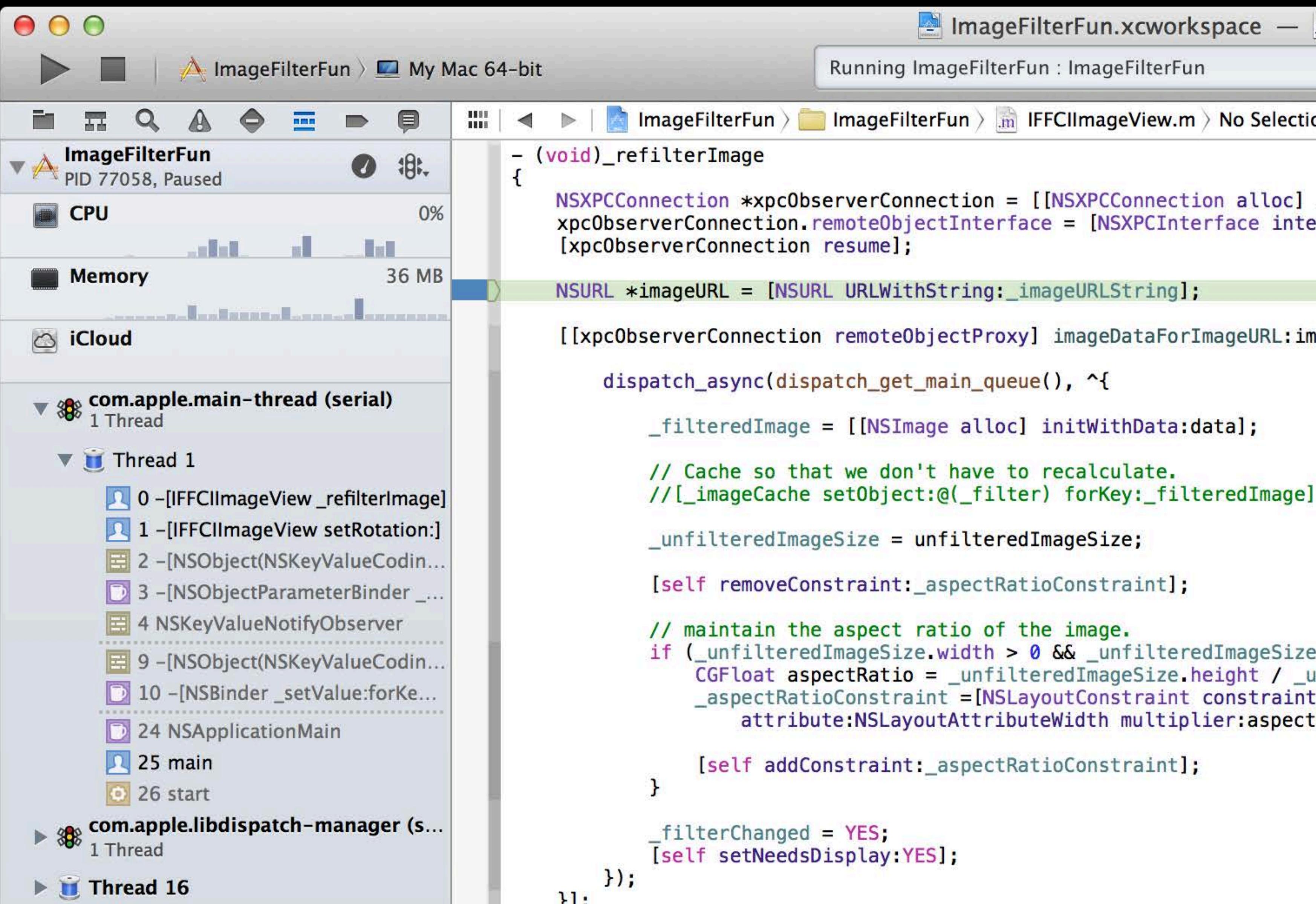
# Debug Navigator



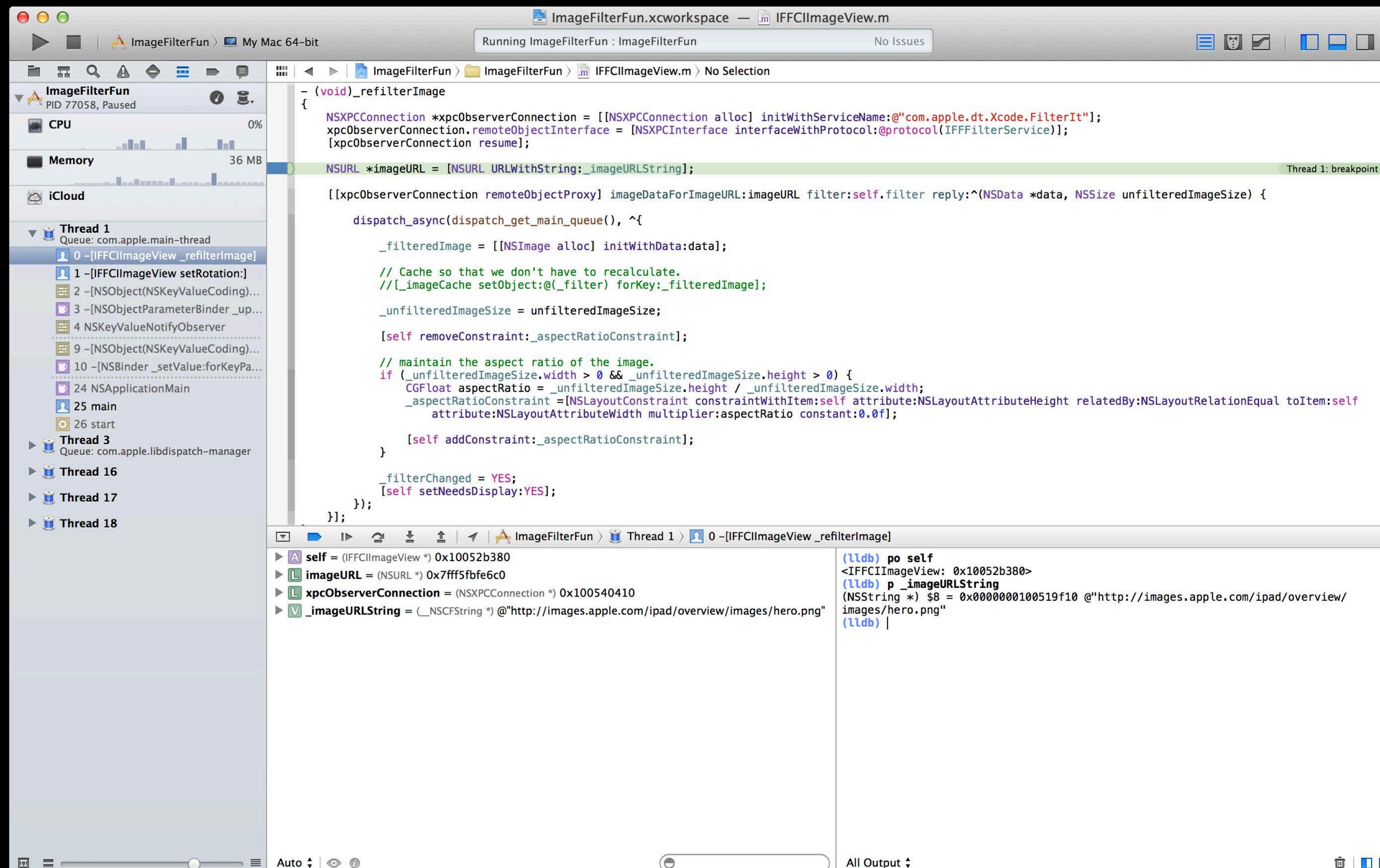
# Debug Navigator



# Debug Navigator



# Variables View



```
1 -[IFFCIIImageView setRotation:] // Cache so that we don't have to recalculate.  
2 -[NSObject(NSKeyValueCoding) ... //[_imageCache setObject:@(_filter) forKey:_filteredImage];  
3 -[NSObjectParameterBinder _up...  
4 NSKeyValueNotifyObserver  
-----  
9 -[NSObject(NSKeyValueCoding) ...  
10 -[NSBinder _setValueForKeyPa...  
24 NSApplicationMain  
25 main  
26 start  
Thread 3 Queue: com.apple.libdispatch-manager  
Thread 16  
Thread 17  
Thread 18
```

1 -[IFFCIIImageView setRotation:] // Cache so that we don't have to recalculate.  
2 -[NSObject(NSKeyValueCoding) ... //[\_imageCache setObject:@(\_filter) forKey:\_filteredImage];  
3 -[NSObjectParameterBinder \_up...  
4 NSKeyValueNotifyObserver  
-----  
9 -[NSObject(NSKeyValueCoding) ...  
10 -[NSBinder \_setValueForKeyPa...  
24 NSApplicationMain  
25 main  
26 start

Thread 3 Queue: com.apple.libdispatch-manager  
Thread 16  
Thread 17  
Thread 18

0 -[IFFCIIImageView \_refilterImage]

```
▶ A self = (IFFCIIImageView *) 0x10052b380  
▶ L imageURL = (NSURL *) 0x7fff5fbfe6c0  
▶ L xpcObserverConnection = (NSXPCCConnection *) 0x100540410  
▶ V _imageURLString = (_NSCFString *) @"http://images.apple.com/ipad/overview/images/hero.png"  
(lldb) po self  
<IFFCIIImageView: 0x10052b380>  
(lldb) p _imageURLString  
(NSString *) $8 = 0x000000100519f10 @"http://images.apple.com/ipad/overview/images/hero.png"  
(lldb)
```

```
1 -[IFFCIIImageView setRotation:] // Cache so that we don't have to recalculate.  
2 -[NSObject(NSKeyValueCoding) ... //[_imageCache setObject:@(_filter) forKey:_filteredImage];  
3 -[NSObjectParameterBinder _up...  
4 NSKeyValueNotifyObserver  
-----  
9 -[NSObject(NSKeyValueCoding) ...  
10 -[NSBinder _setValueForKeyPa...  
24 NSApplicationMain  
25 main  
26 start  
Thread 3 Queue: com.apple.libdispatch-manager  
Thread 16  
Thread 17  
Thread 18
```

Thread 1 > 0 -[IFFCIIImageView \_refilterImage]

```
(lldb) po self  
<IFFCIIImageView: 0x10052b380>  
(lldb) p _imageURLString  
(NSString *) $8 = 0x000000100519f10 @"http://images.apple.com/ipad/overview/images/hero.png"  
(lldb)
```

Auto | All Output

```
1 -[IFFCIIImageView setRotation:] // Cache so that we don't have to recalculate.  
2 -[NSObject(NSKeyValueCoding) ... //[_imageCache setObject:@(_filter) forKey:_filteredImage];  
3 -[NSObjectParameterBinder _up...  
4 NSKeyValueNotifyObserver  
-----  
9 -[NSObject(NSKeyValueCoding) ...  
10 -[NSBinder _setValueForKeyPa...  
24 NSApplicationMain  
25 main  
26 start  
Thread 3 Queue: com.apple.libdispatch-manager  
Thread 16  
Thread 17  
Thread 18
```

1 -[IFFCIIImageView setRotation:] // Cache so that we don't have to recalculate.  
2 -[NSObject(NSKeyValueCoding) ... //[\_imageCache setObject:@(\_filter) forKey:\_filteredImage];  
3 -[NSObjectParameterBinder \_up...  
4 NSKeyValueNotifyObserver  
-----  
9 -[NSObject(NSKeyValueCoding) ...  
10 -[NSBinder \_setValueForKeyPa...  
24 NSApplicationMain  
25 main  
26 start

Thread 3 Queue: com.apple.libdispatch-manager  
Thread 16  
Thread 17  
Thread 18

0 -[IFFCIIImageView \_refilterImage]

```
(lldb) po self  
<IFFCIIImageView: 0x10052b380>  
(lldb) p _imageURLString  
(NSString *) $8 = 0x000000100519f10 @"http://images.apple.com/ipad/overview/images/hero.png"  
(lldb)
```

```
1 -[IFFCIIImageView setRotation:] // Cache so that we don't have to recalculate.  
2 -[NSObject(NSKeyValueCoding) ... //[_imageCache setObject:@(_filter) forKey:_filteredImage];  
3 -[NSObjectParameterBinder _up...  
4 NSKeyValueNotifyObserver  
-----  
9 -[NSObject(NSKeyValueCoding) ...  
10 -[NSBinder _setValueForKeyPa...  
24 NSApplicationMain  
25 main  
26 start  
Thread 3 Queue: com.apple.libdispatch-manager  
Thread 16  
Thread 17  
Thread 18
```

1 -[IFFCIIImageView setRotation:] // Cache so that we don't have to recalculate.  
2 -[NSObject(NSKeyValueCoding) ... //[\_imageCache setObject:@(\_filter) forKey:\_filteredImage];  
3 -[NSObjectParameterBinder \_up...  
4 NSKeyValueNotifyObserver  
-----  
9 -[NSObject(NSKeyValueCoding) ...  
10 -[NSBinder \_setValueForKeyPa...  
24 NSApplicationMain  
25 main  
26 start

Thread 3 Queue: com.apple.libdispatch-manager  
Thread 16  
Thread 17  
Thread 18

0 -[IFFCIIImageView \_refilterImage]

```
▶ A self = (IFFCIIImageView *) 0x10052b380  
▶ L imageURL = (NSURL *) 0x7fff5fbfe6c0  
▶ L xpcObserverConnection = (NSXPCCConnection *) 0x100540410  
▶ V _imageURLString = (_NSCFString *) @"http://images.apple.com/ipad/overview/images/hero.png"  
(lldb) po self  
<IFFCIIImageView: 0x10052b380>  
(lldb) p _imageURLString  
(NSString *) $8 = 0x000000100519f10 @"http://images.apple.com/ipad/overview/images/hero.png"  
(lldb)
```

Auto ▾

All Output ▾

```
1 -[IFFCIIImageView setRotation:] // Cache so that we don't have to recalculate.  
2 -[NSObject(NSKeyValueCoding)... //[_imageCache setObject:@(_filter) forKey:_filteredImage];  
3 -[NSObjectParameterBinder _up...  
4 NSKeyValueNotifyObserver  
-----  
9 -[NSObject(NSKeyValueCoding)...  
10 -[NSBinder _setValueForKeyPa...  
24 NSApplicationMain  
25 main  
26 start  
Thread 3 Queue: com.apple.libdispatch-manager  
Thread 16  
Thread 17  
Thread 18  
| [ ] ImageFilterFun > Thread 1 > 0 -[IFFCIIImageView _refilterImage]  
| ► A self = (IFFCIIImageView *) 0x10052b380  
| ► A _cmd = (SEL) "_refilterImage"  
| ► L xpcObserverConnection = (NSXPCCConnection *) 0x100540410  
| ► L imageURL = (NSURL *) 0x7fff5fbfe6c0  
| ► R Exception State Registers  
| ► R Floating Point Registers  
| ► R General Purpose Registers  
| (lldb) po self  
<IFFCIIImageView: 0x10052b380>  
(lldb) p _imageURLString  
(NSString *) $8 = 0x000000100519f10 @"http://images.ap...  
images/hero.png"  
(lldb) |  
All ▾ | ⚡ ⓘ All Output ▾
```

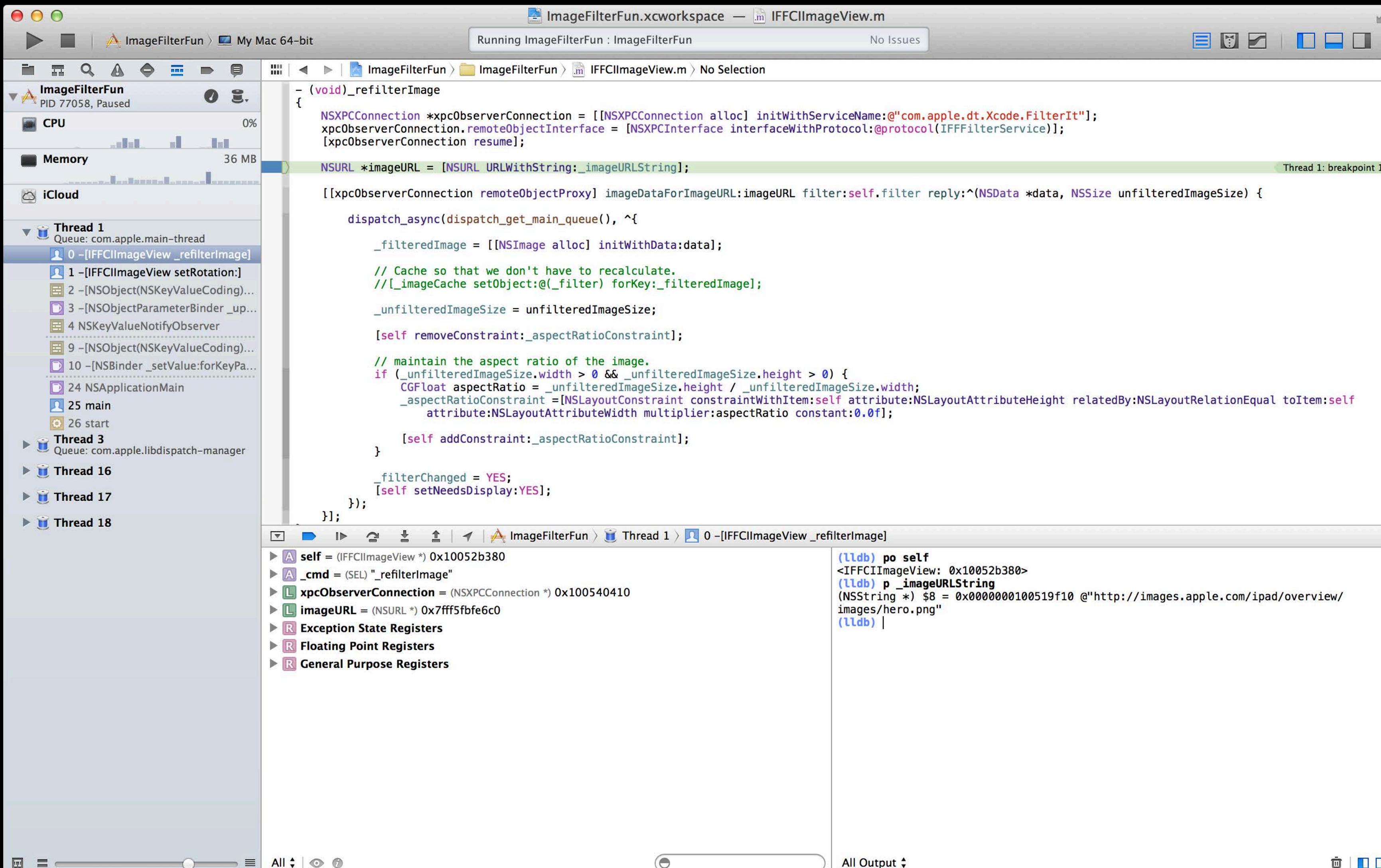
```
1 -[IFFCIIImageView setRotation:] // Cache so that we don't have to recalculate.  
2 -[NSObject(NSKeyValueCoding)... //[_imageCache setObject:@(_filter) forKey:_filteredImage];  
3 -[NSObjectParameterBinder _up...  
4 NSKeyValueNotifyObserver  
-----  
9 -[NSObject(NSKeyValueCoding)...  
10 -[NSBinder _setValueForKeyPa...  
24 NSApplicationMain  
25 main  
26 start  
Thread 3 Queue: com.apple.libdispatch-manager  
Thread 16  
Thread 17  
Thread 18
```

Thread 1 > 0 -[IFFCIIImageView \_refilterImage]

```
(lldb) po self  
<IFFCIIImageView: 0x10052b380>  
(lldb) p _imageURLString  
(NSString *) $8 = 0x000000100519f10 @"http://images.ap...  
images/hero.png"  
(lldb) |
```

All | All Output

# Debug Console



```
_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.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]

po self  
<IFFCIIImageView: 0x10052b380>  
p \_imageURLString  
\$8 = 0x000000100519f10 @"http://images.apple.com/ipad/overview/images/hero.png"

Registers Registers Registers

All Output

```
_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.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 = (NSXPCConnection *) 0x100540410  
(NSURL *) 0x7fff5fbfe6c0  
ate Registers  
nt Registers  
ose Registers
```

(lldb) po self  
<IFFCIIImageView: 0x10052b380>  
(lldb) p \_imageURLString  
(NSString \*) \$8 = 0x000000100519f10 @"http://images.apple.com/ipad/overview/  
images/hero.png"  
(lldb) |

```
_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.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];
```

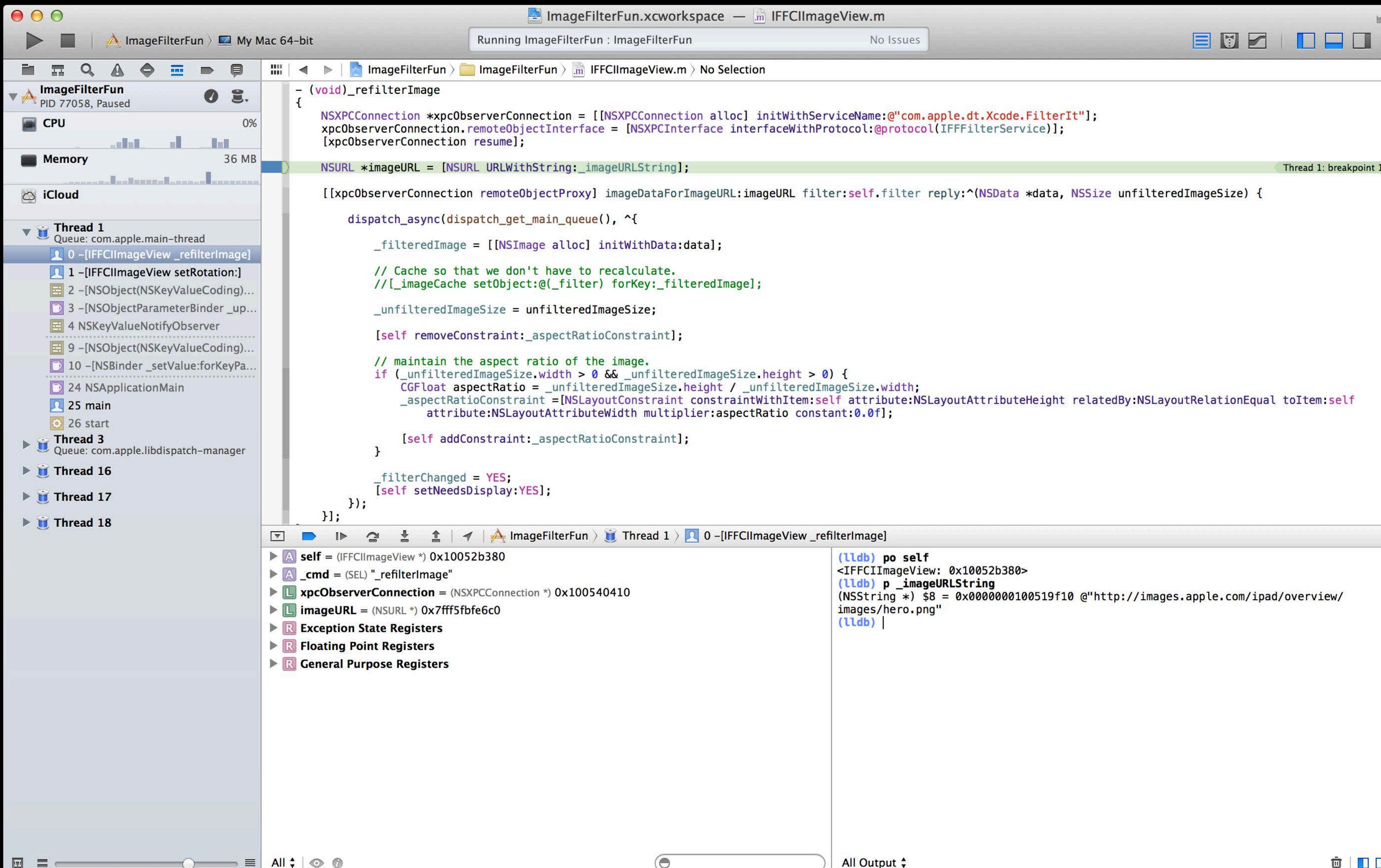
ImageFilterFun > Thread 1 > 0 -[IFFCIIImageView \_refilterImage]

po self  
<IFFCIIImageView: 0x10052b380>  
p \_imageURLString  
\$8 = 0x000000100519f10 @"http://images.apple.com/ipad/overview/images/hero.png"  
|

Registers  
Registers  
Registers

All Output

# Debug Console



# 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.”

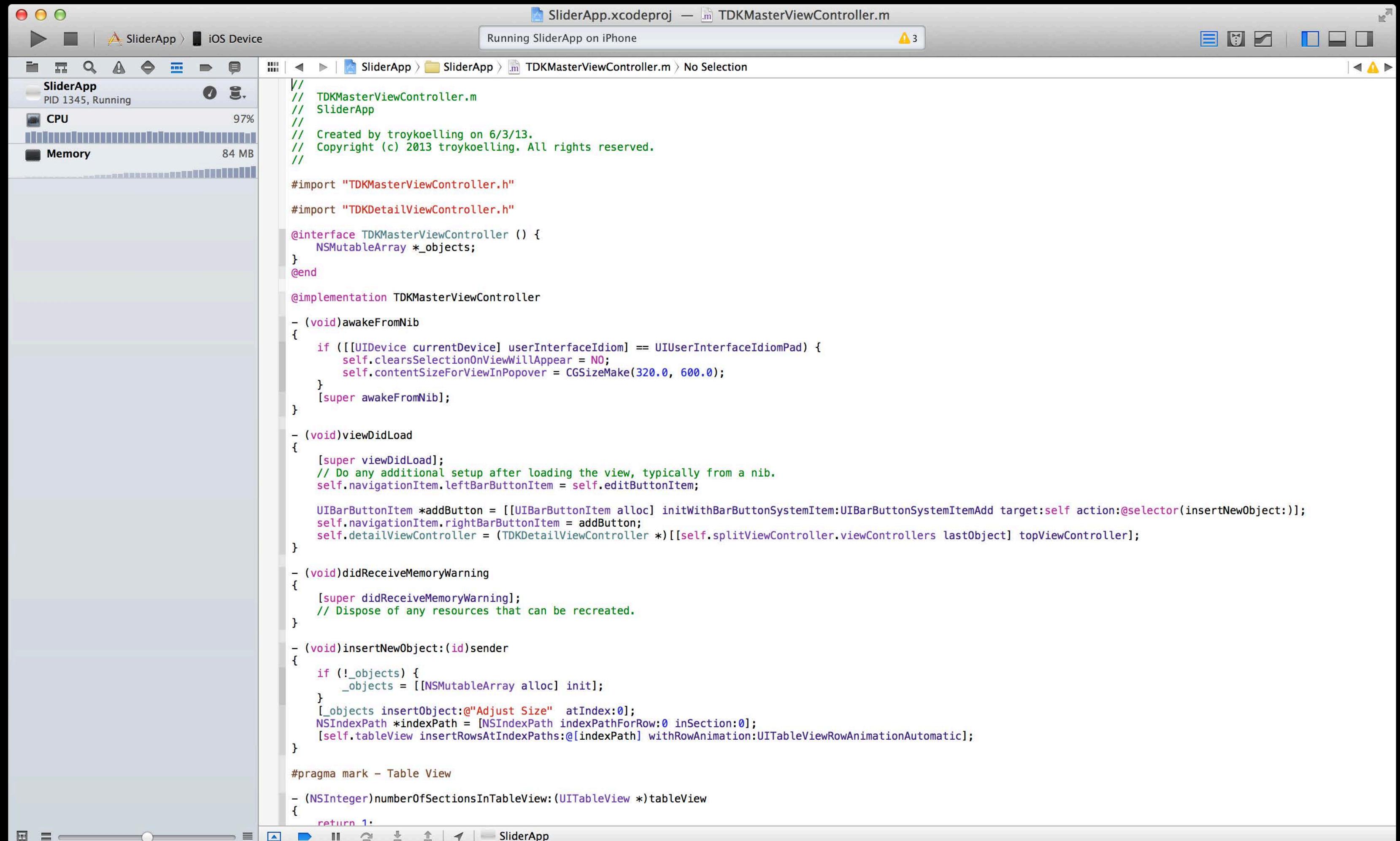
Donald Knuth



Optimize when you  
measure a problem

# Debug Gauges

- CPU
- Memory
- iCloud
- Energy
- OpenGL ES



The screenshot shows the Xcode interface with the following details:

- Top Bar:** Shows "SliderApp.xcodeproj — TDKMasterViewController.m" and "Running SliderApp on iPhone".
- Left Sidebar:** Displays the "SliderApp" target, PID 1345, Running, with sections for CPU (97%) and Memory (84 MB).
- Code Editor:** The file "TDKMasterViewController.m" is open, showing the implementation of the TDKMasterViewController class.
- Bottom Bar:** Standard Xcode navigation buttons and the project name "SliderApp".

```
// 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.
}

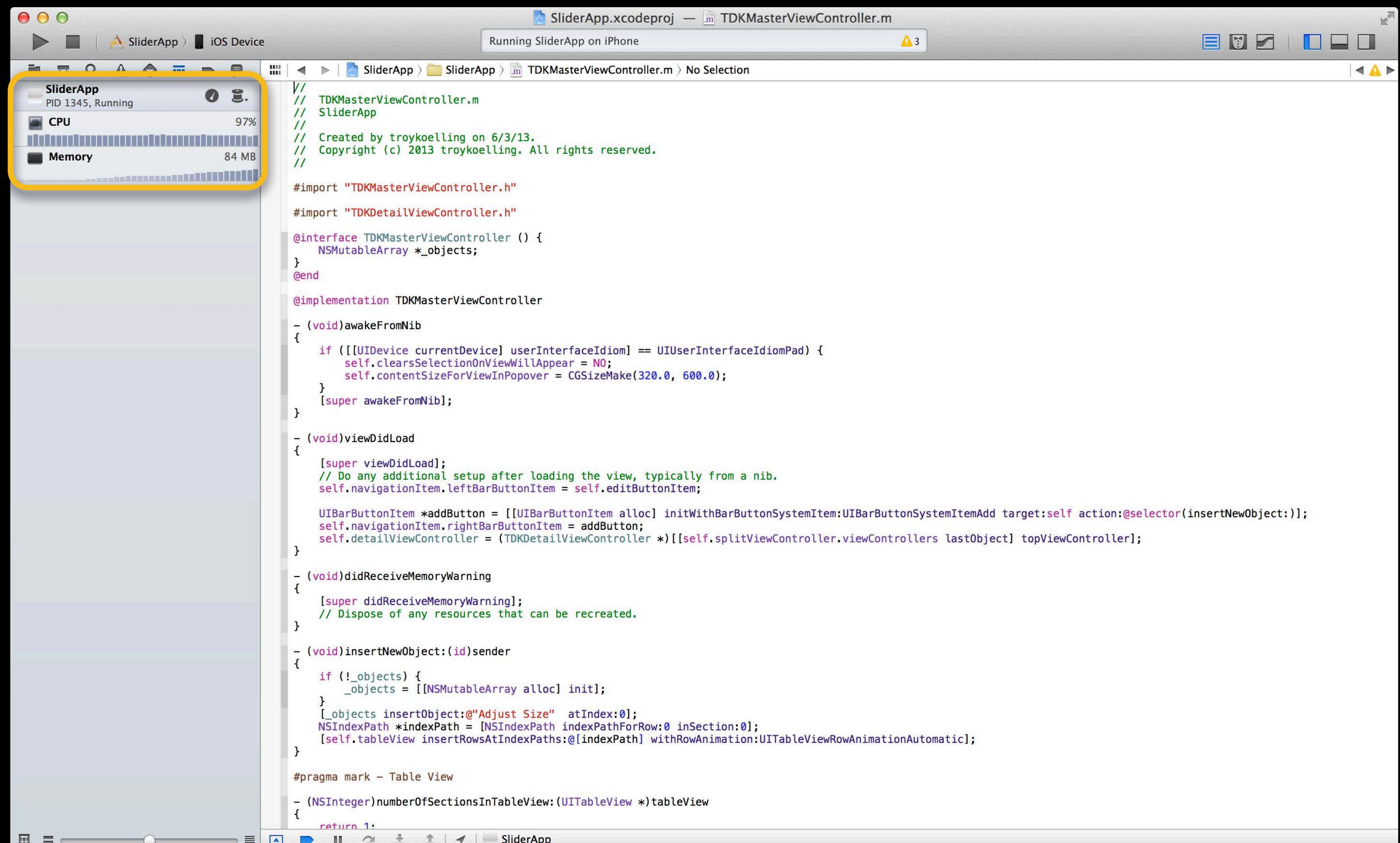
- (void)insertNewObject:(id)sender
{
    if (!_objects) {
        _objects = [[NSMutableArray alloc] init];
    }
    [_objects insertObject:@"Adjust Size" atIndex:0];
    NSIndexPath *indexPath = [NSIndexPath indexPathForRow:0 inSection:0];
    [self.tableView insertRowsAtIndexPaths:@[indexPath] withRowAnimation:UITableViewRowAnimationAutomatic];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

#pragma mark - Table View

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

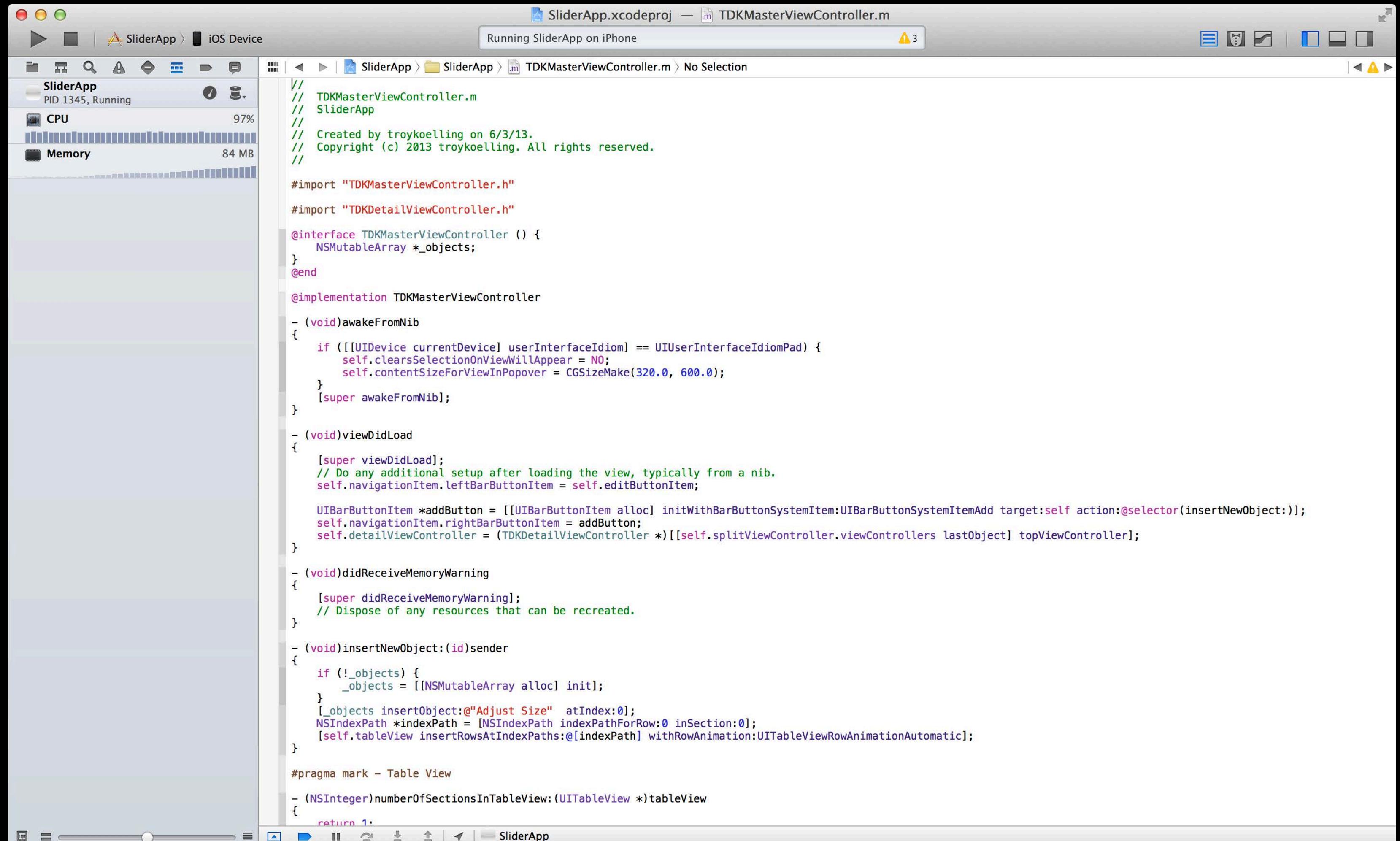
# Debug Gauges

- CPU
- Memory
- iCloud
- Energy
- OpenGL ES



# Debug Gauges

- CPU
- Memory
- iCloud
- Energy
- OpenGL ES



The screenshot shows the Xcode interface with the "SliderApp.xcodeproj" project open. The "TDKMasterViewController.m" file is selected in the project navigator. The left side of the screen features the "Debug Navigator" which displays the "SliderApp" process (PID 1345, Running) with two gauges: "CPU" at 97% and "Memory" at 84 MB. The main editor area contains 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] == 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];
}

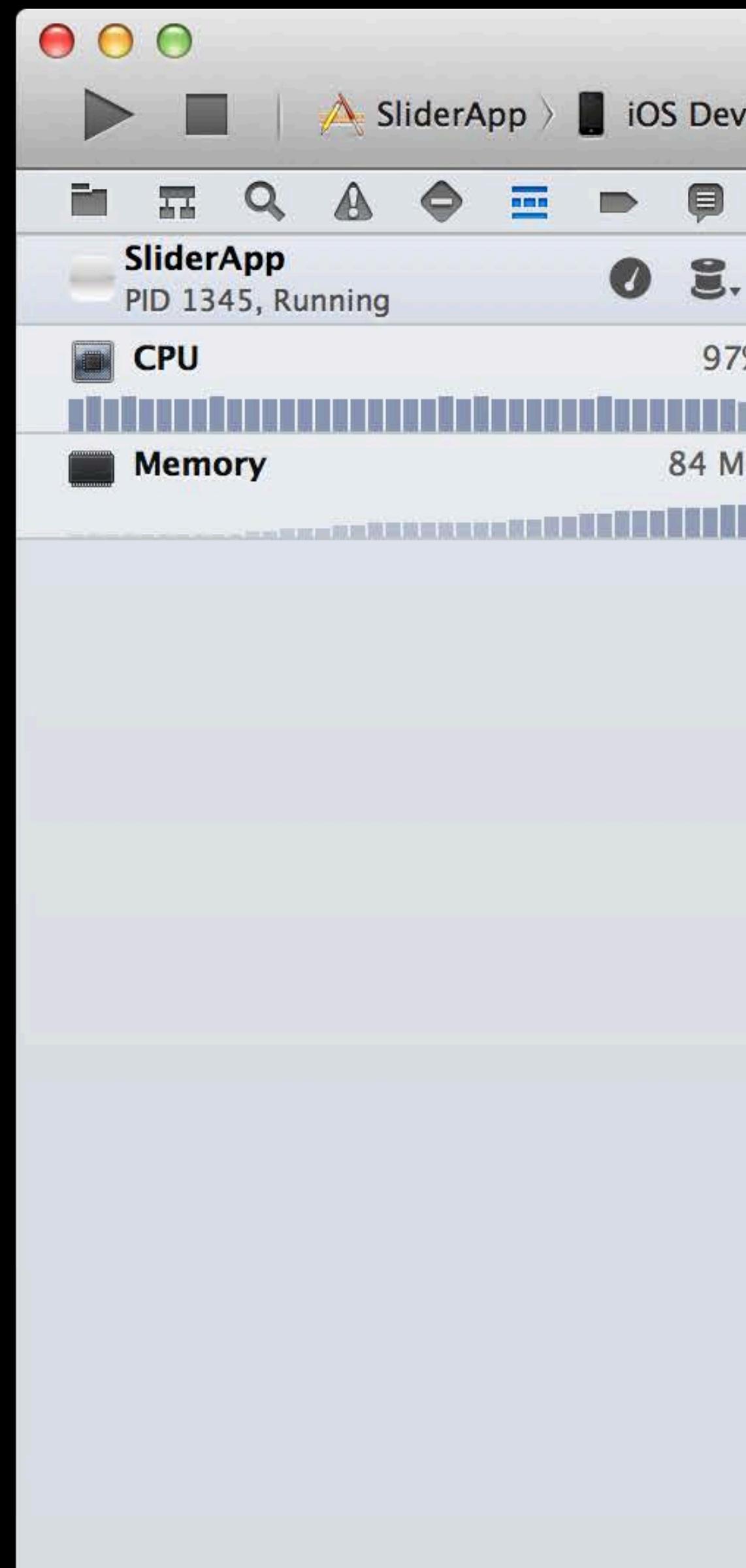
- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (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;
}
```

# Understanding Performance



The screenshot shows the Xcode interface with the following details:

- Project: SliderApp.xcodeproj
- Target: SliderApp
- Platform: iOS Device
- Status: Running SliderApp on iPhone
- Process: SliderApp (PID 1345, Running)
- Metrics:
  - CPU: 97%
  - Memory: 84 MB
- File Navigator: Shows the file structure: SliderApp > SliderApp > TDKMasterViewController.m
- Editor: Displays the 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

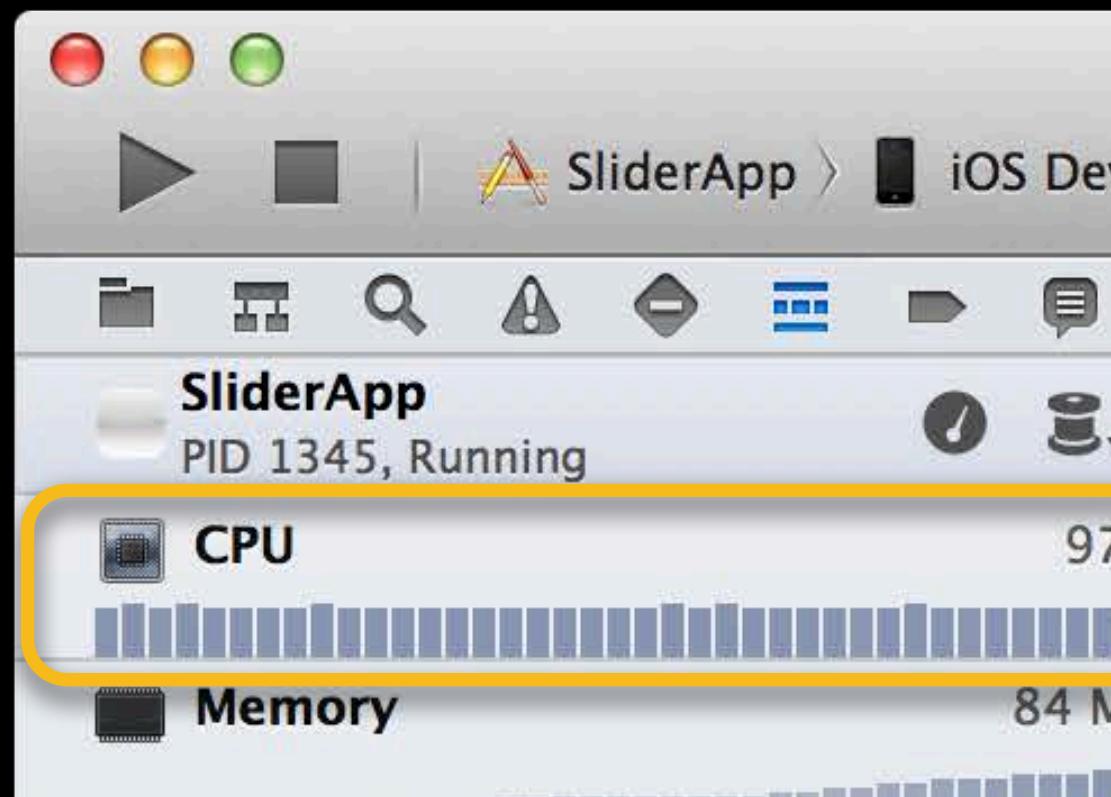
@implementation TDKMasterViewController

- (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:)];
}
```

# CPU Report



The screenshot shows the Xcode interface with the project 'SliderApp.xcodeproj' open. The main window displays the code for 'TDKMasterViewController.m'. In the bottom-left corner, there is a real-time monitoring panel for the running application 'SliderApp' (PID 1345). The CPU usage is highlighted with a yellow box and shows a value of 97%. Below the CPU bar, the memory usage is shown as 84 MB.

```
// 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

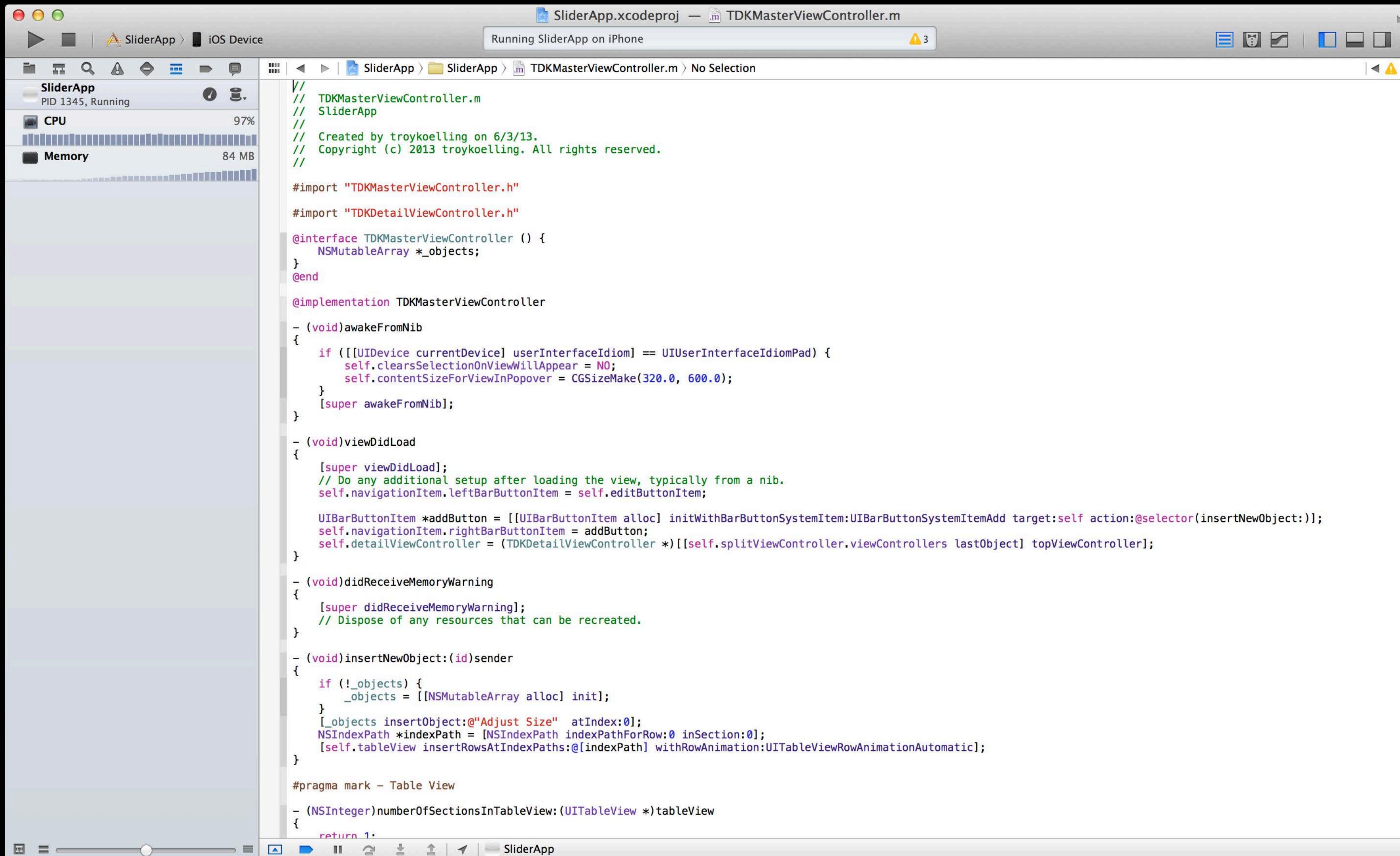
@implementation TDKMasterViewController

- (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:
```

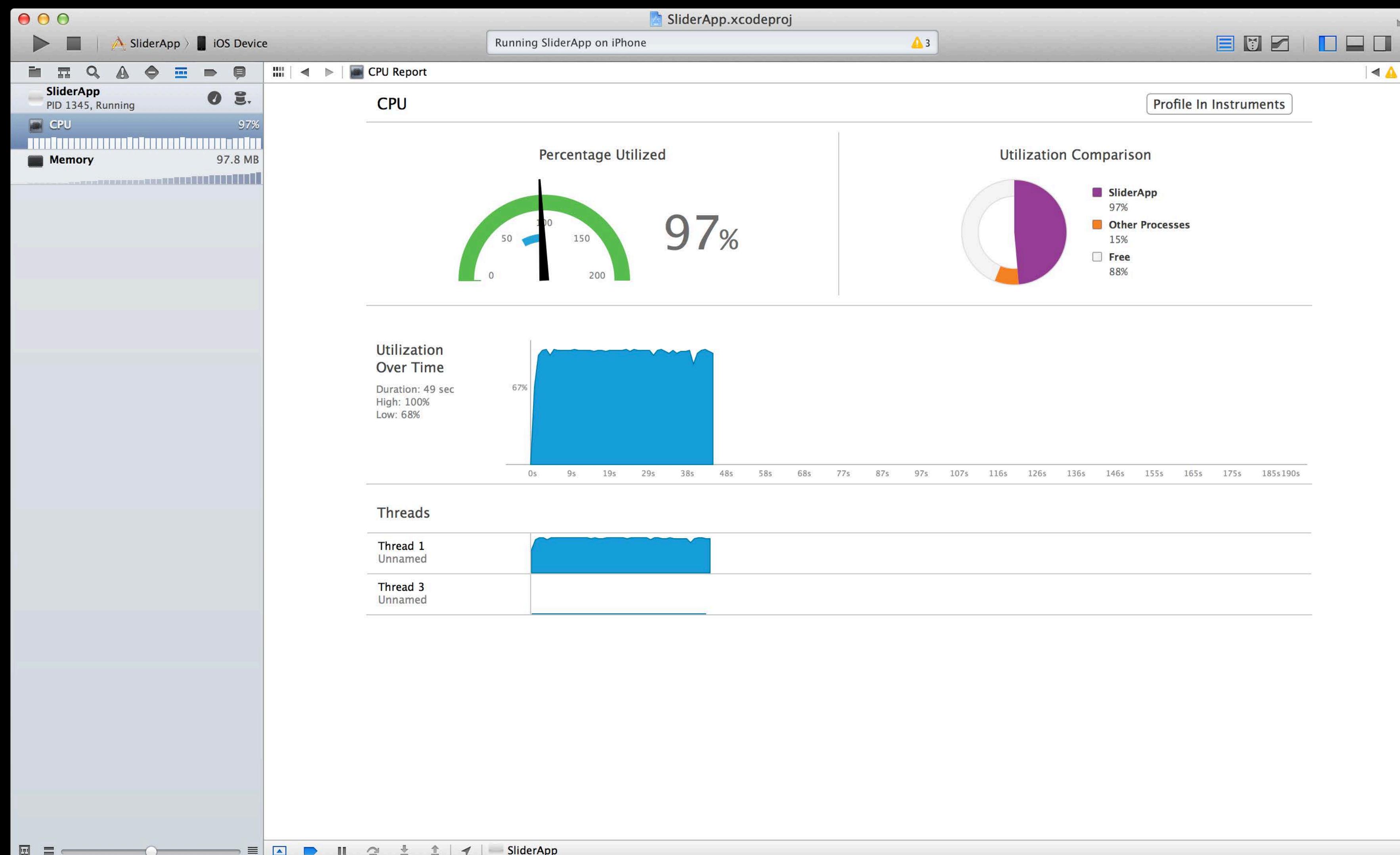
# CPU Report



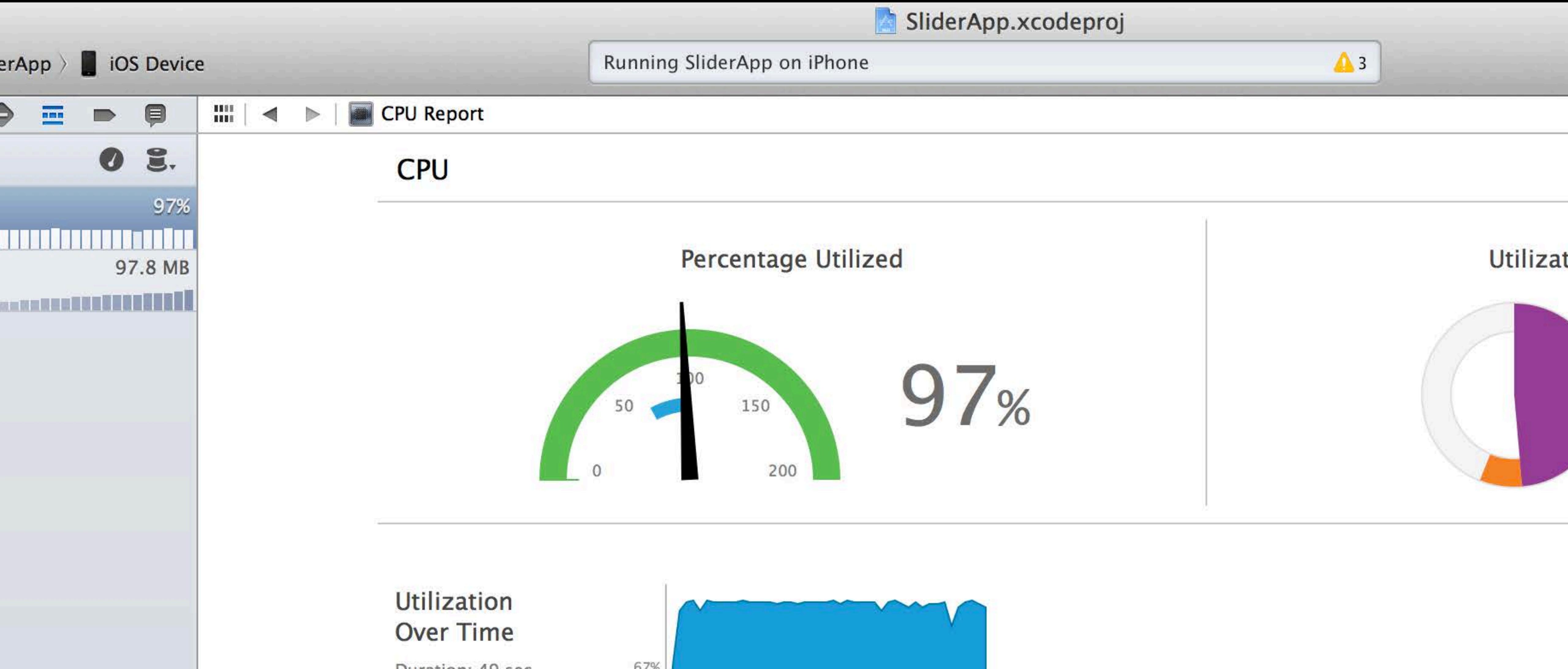
The screenshot shows the Xcode interface with the following details:

- Project:** SliderApp.xcodeproj
- File:** TDKMasterViewController.m
- Build Configuration:** iOS Device
- Build ID:** PID 1345, Running
- CPU Usage:** 97%
- Memory Usage:** 84 MB
- Code View:** The main pane displays the source code for `TDKMasterViewController.m`. The code includes imports for `TDKMasterViewController.h` and `TDKDetailViewController.h`, defines an interface with an `NSMutableArray *_objects` property, and implements methods for `awakeFromNib`, `viewDidLoad`, `didReceiveMemoryWarning`, and `insertNewObject:sender`. It also contains a `#pragma mark - Table View` section and a `numberOfSectionsInTableView:` implementation.
- Toolbars:** Standard Xcode toolbars for file navigation, search, and selection are visible at the top and bottom of the editor window.

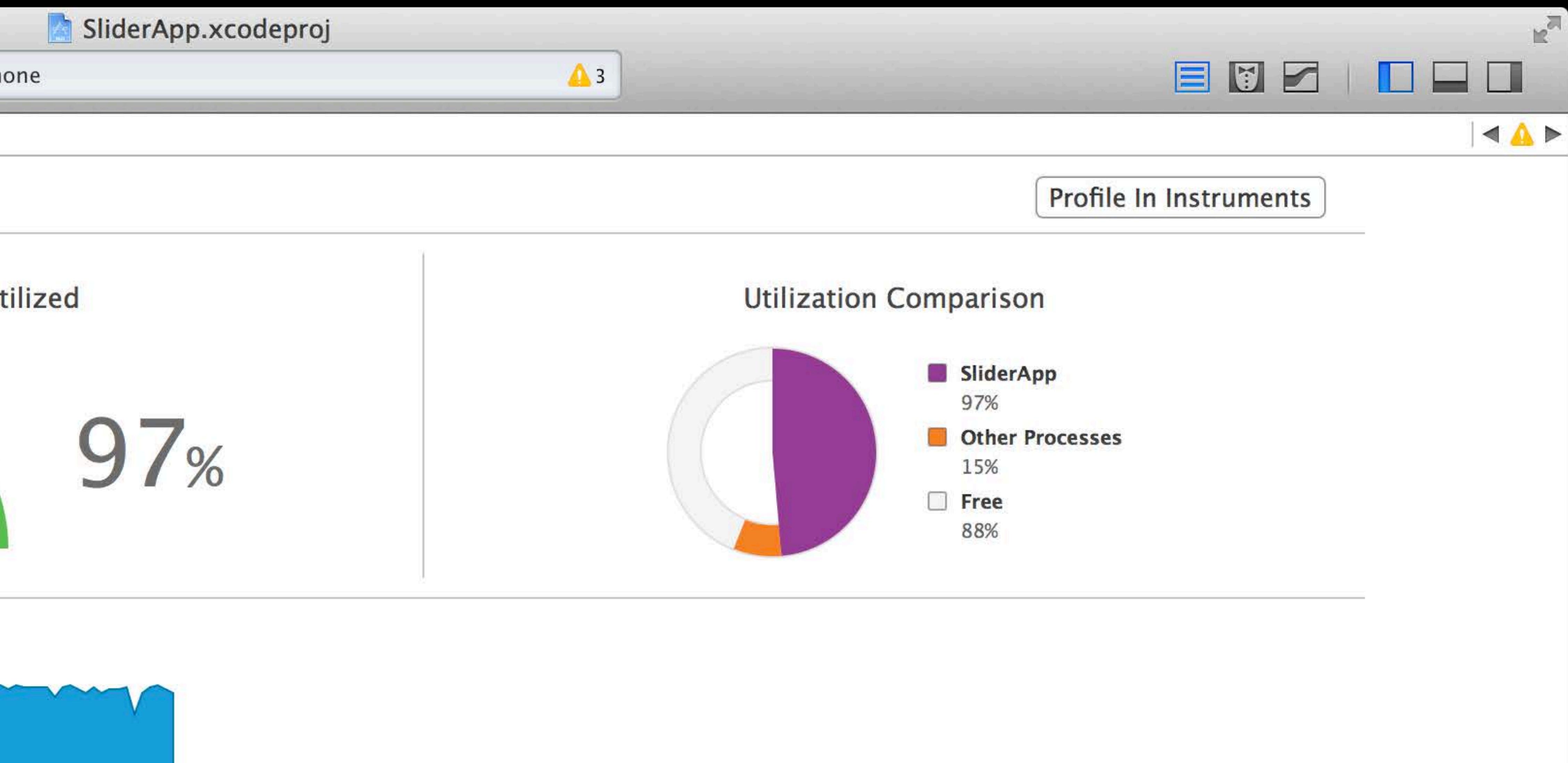
# CPU Report



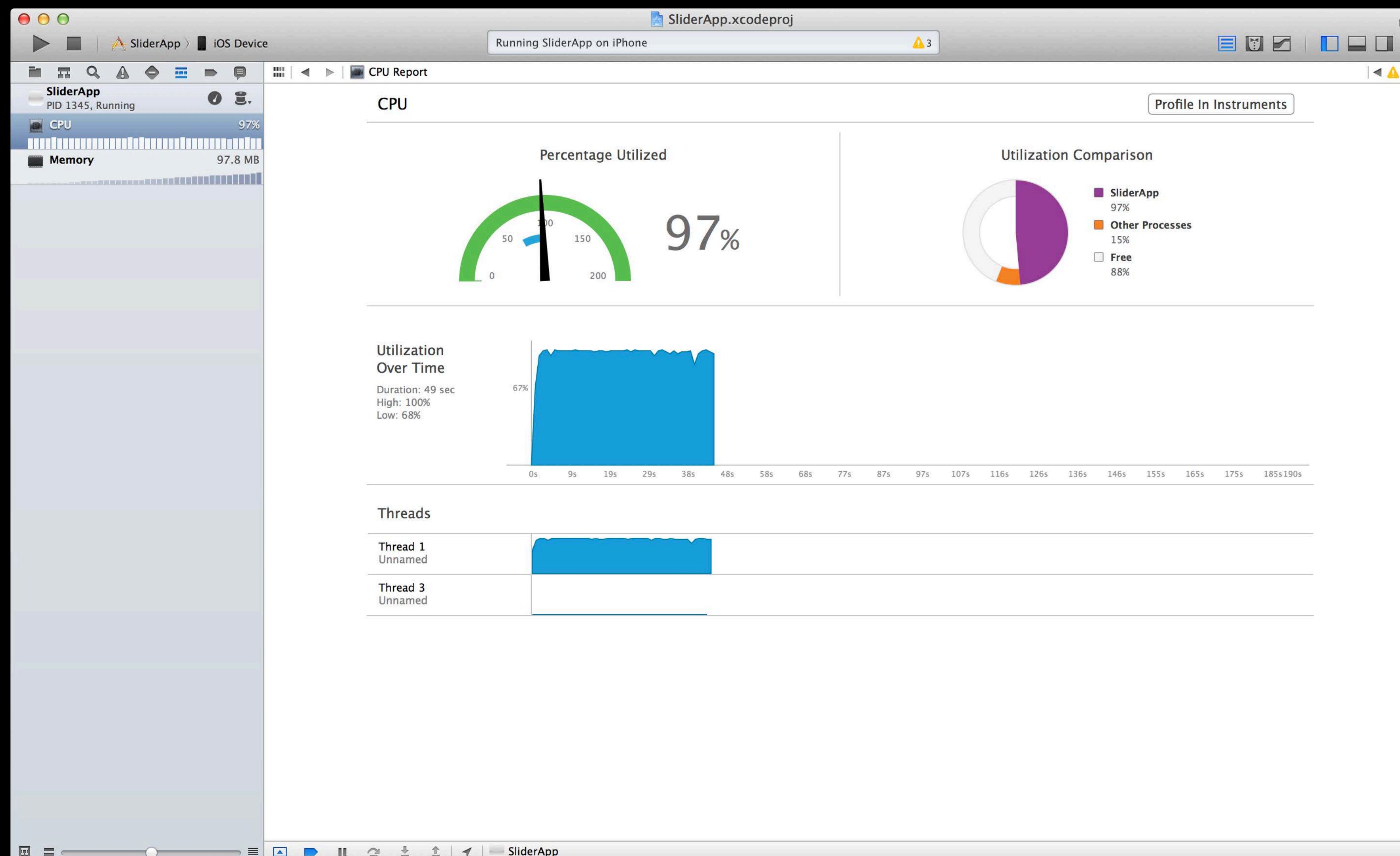
# CPU Report



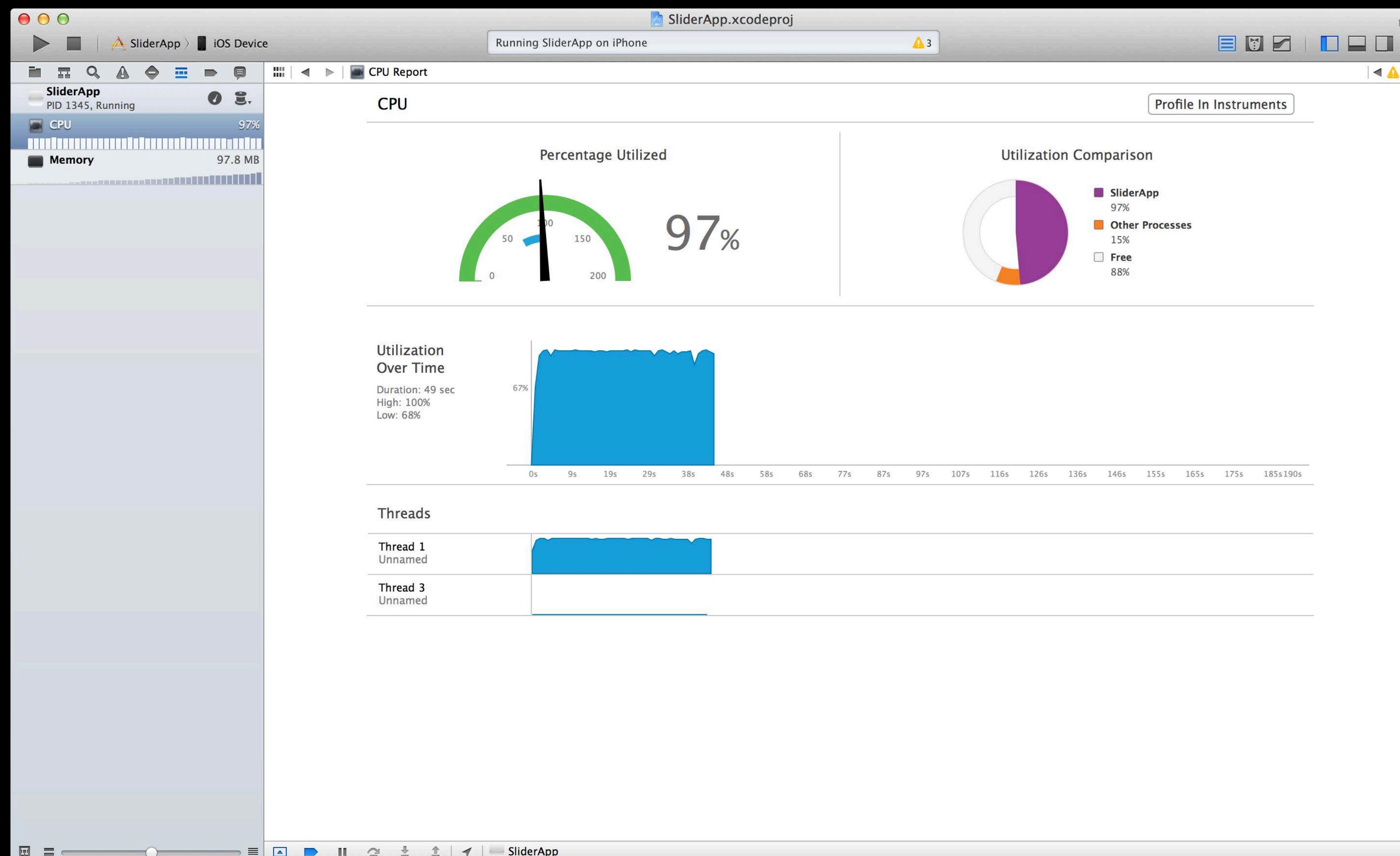
# CPU Report



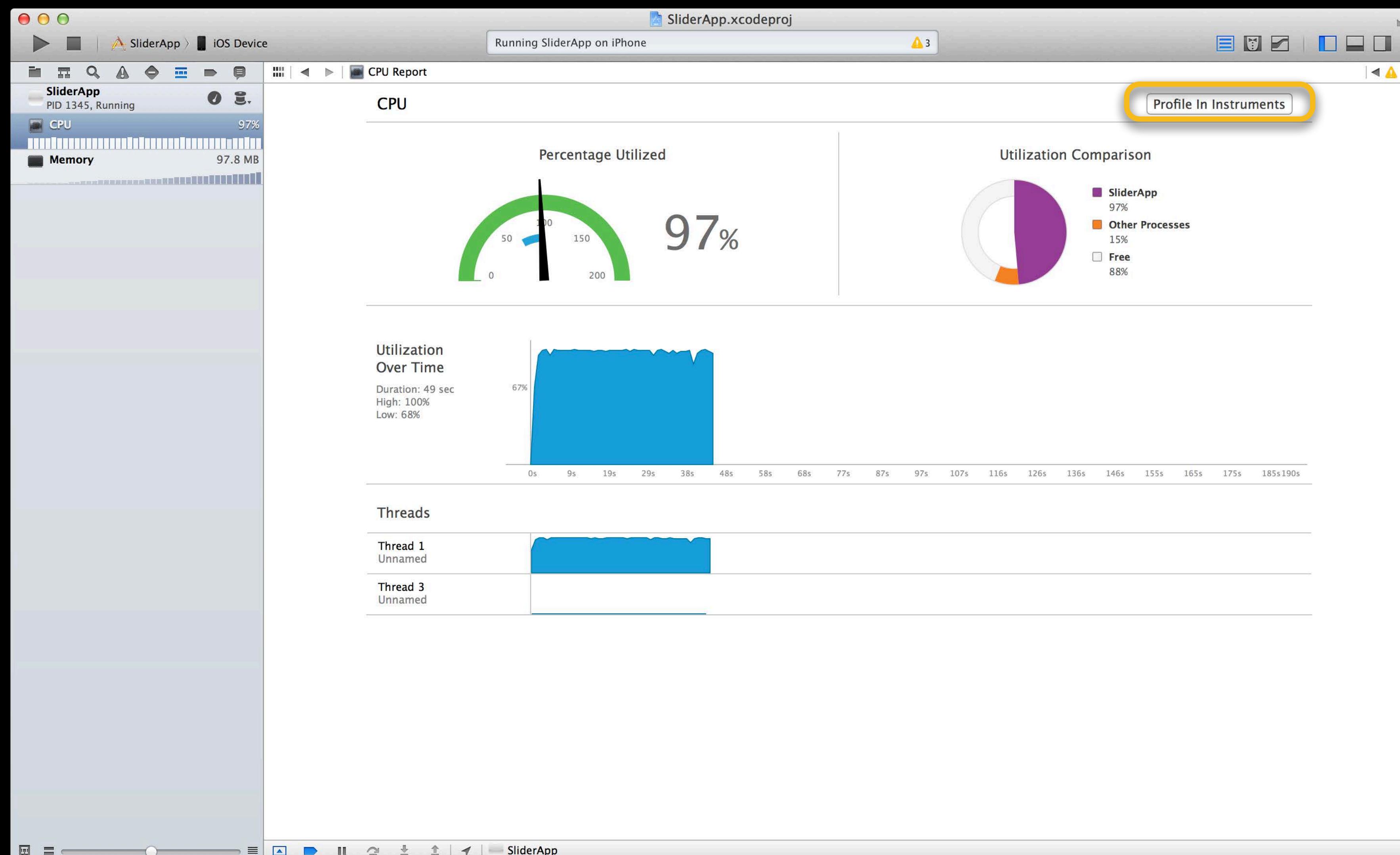
# CPU Report



# CPU Report



# CPU Report



# Memory Report



SliderApp.xcodeproj — TDKMasterViewController.m

Running SliderApp on iPhone

SliderApp PID 1345, Running

CPU 97%

Memory 84 MB

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

#import "TDKMasterViewController.h"

#import "TDKDetailViewController.h"

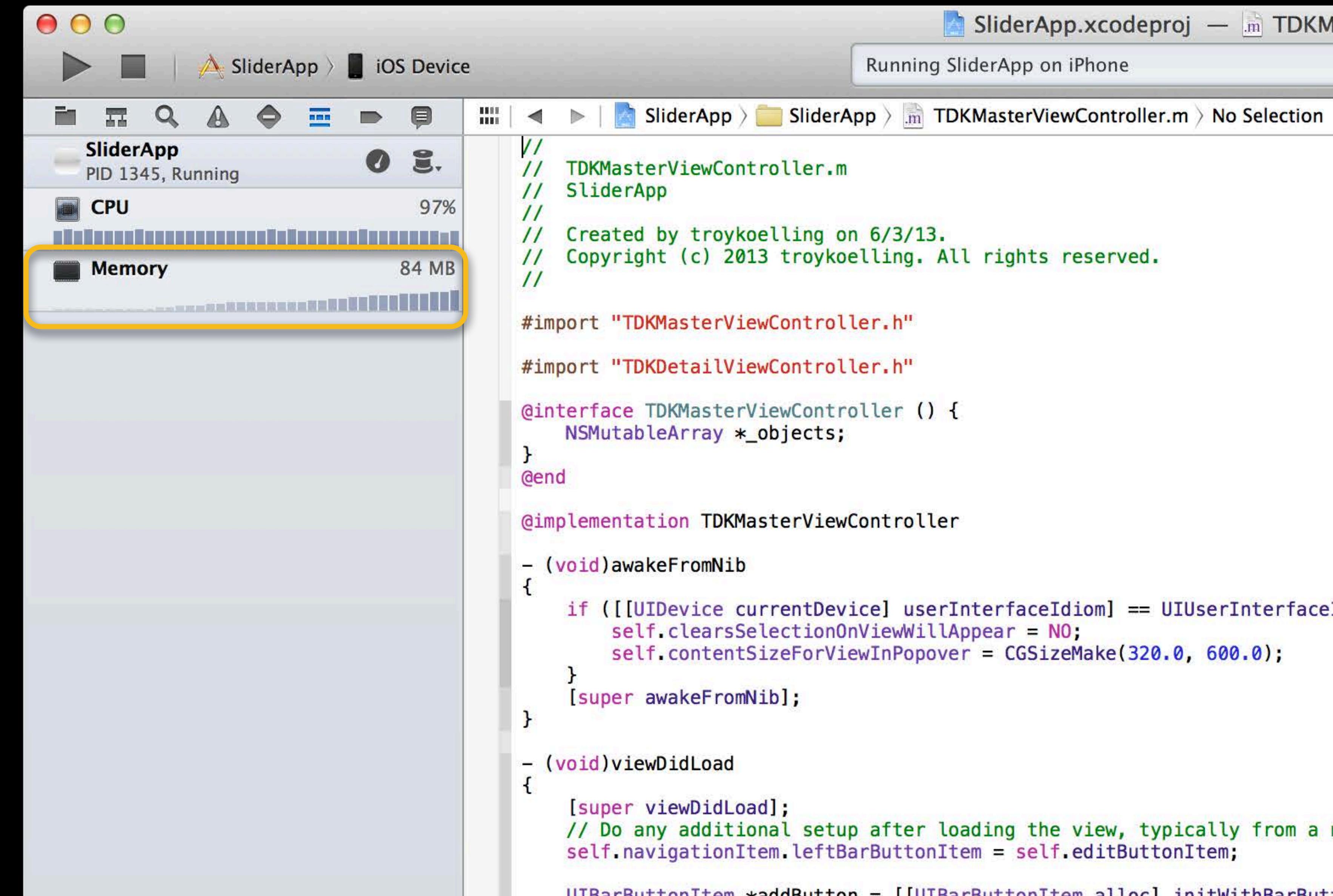
@interface TDKMasterViewController () {
    NSMutableArray *_objects;
}
@end

@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:)];
}
```

# Memory Report



# Memory Report



The screenshot shows the Xcode interface with the following details:

- Project:** SliderApp.xcodeproj
- Target:** SliderApp
- Platform:** iOS Device
- Build ID:** PID 1345, Running
- CPU Usage:** 97%
- Memory Usage:** 84 MB
- File Path:** SliderApp > SliderApp > TDKMasterViewController.m
- Code Preview:** Shows the beginning of the TDKMasterViewController.m file, including imports, a class interface, implementation methods for awakeFromNib and viewDidLoad, and some initial setup code.

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

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

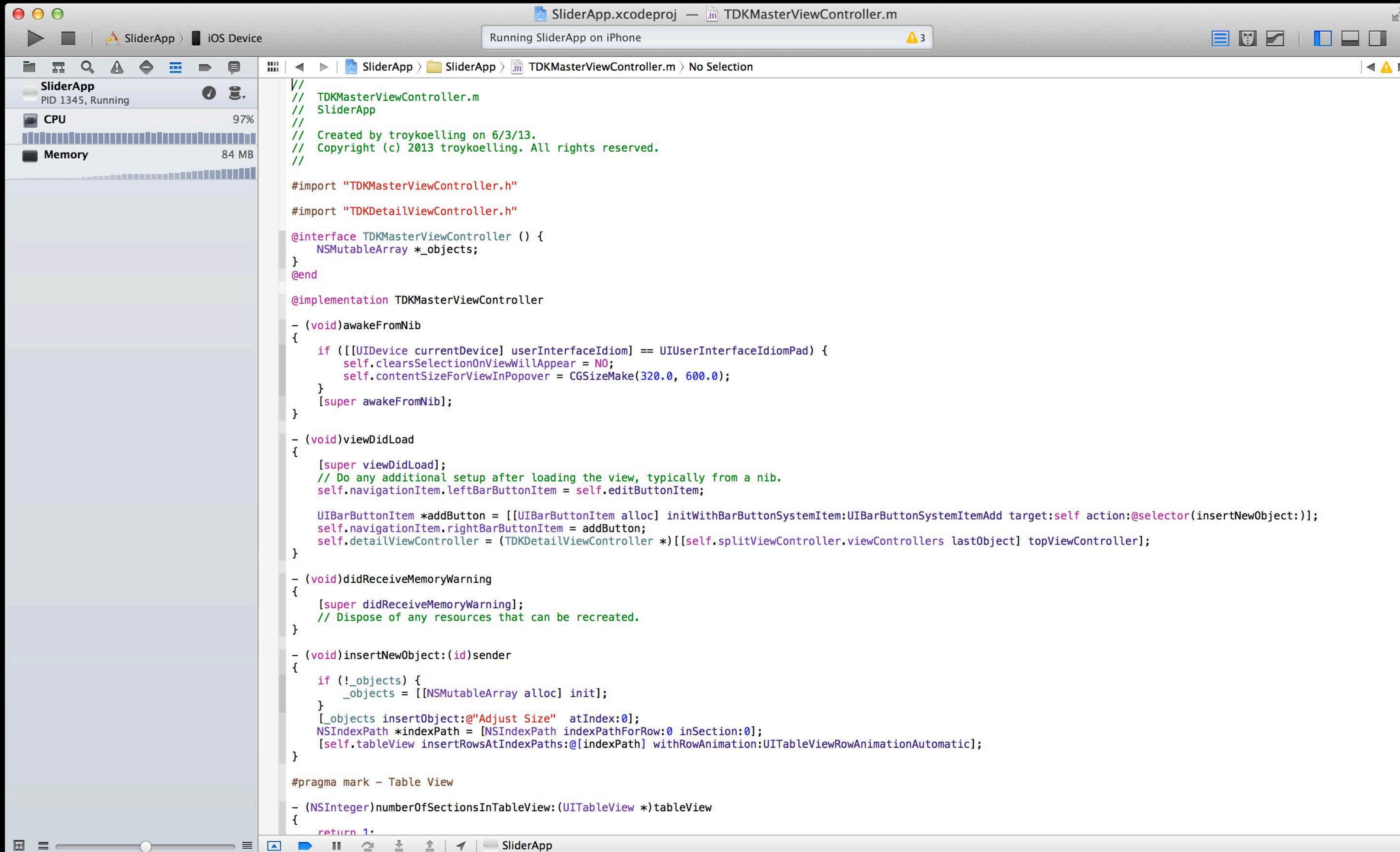
@interface TDKMasterViewController () {
    NSMutableArray *_objects;
}
@end

@implementation TDKMasterViewController

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

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

# Memory Report



```
// 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];
}

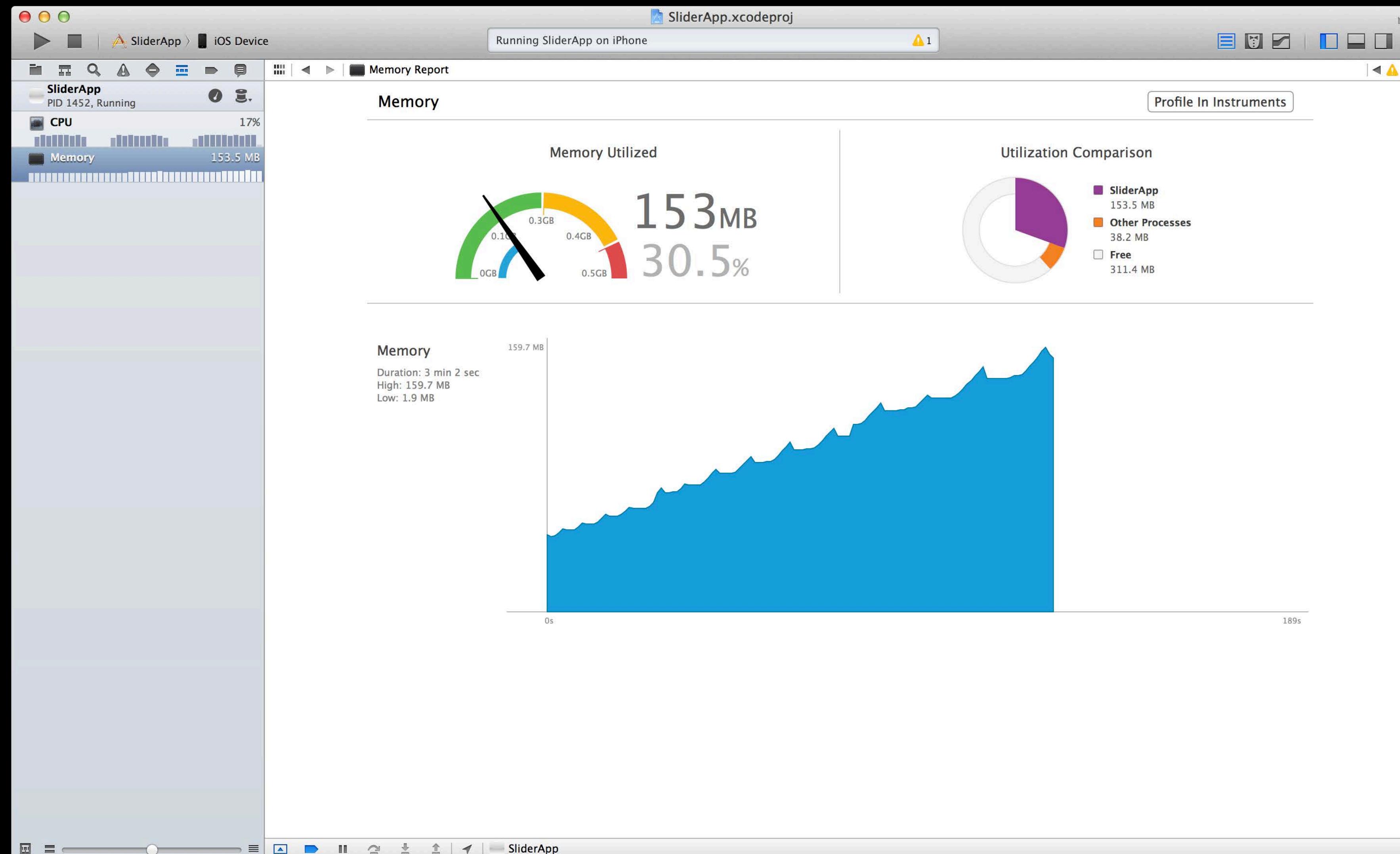
- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (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;
}
```

# Memory Report



# Memory Report

SliderApp.xcodeproj  
Running SliderApp on iPhone 1

iOS Device

Memory Report

Memory

Memory Utilized

153 MB  
30.5%

0GB 0.1GB 0.3GB 0.4GB 0.5GB

17%  
153.5 MB

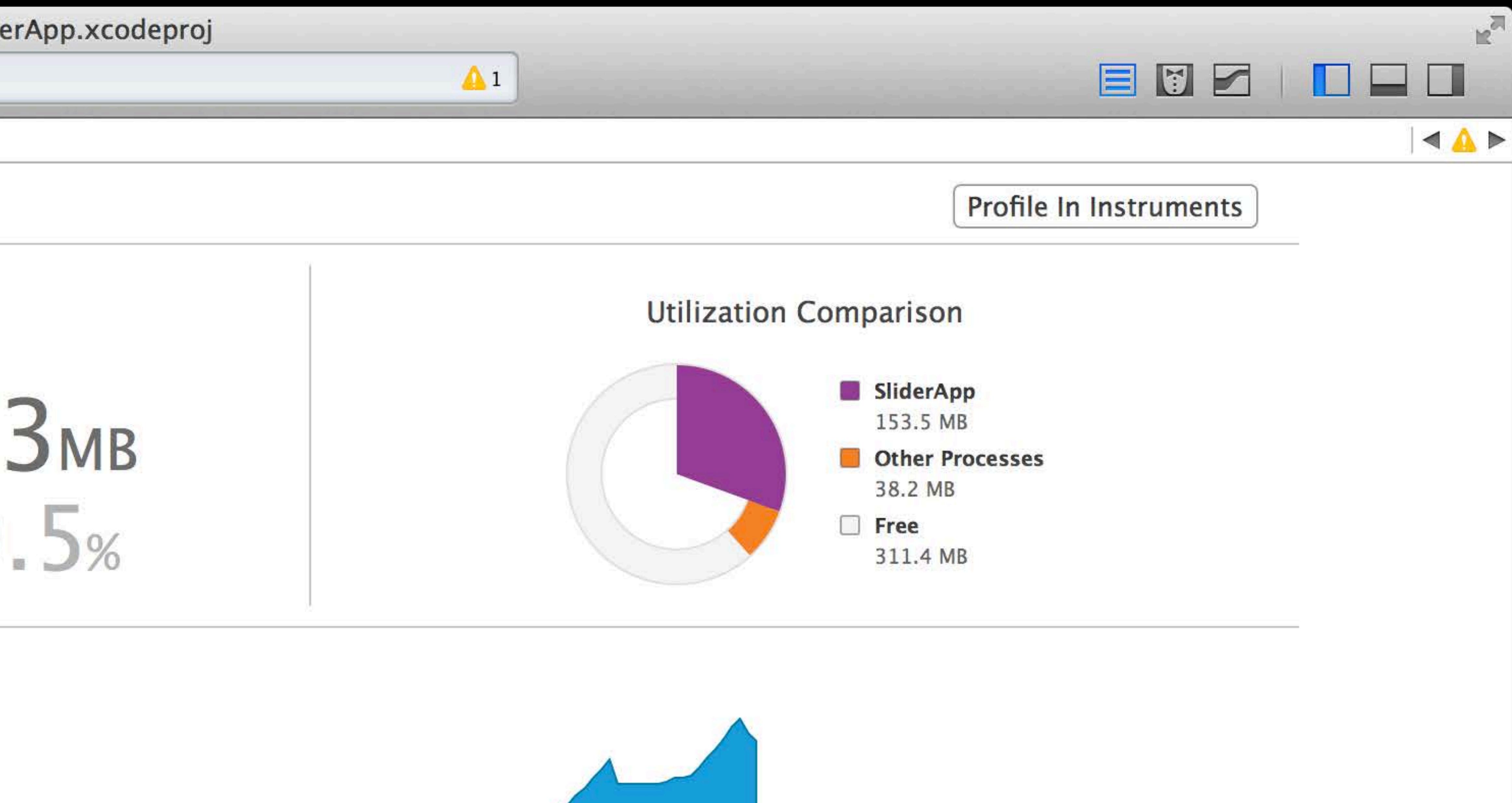
Memory

159.7 MB

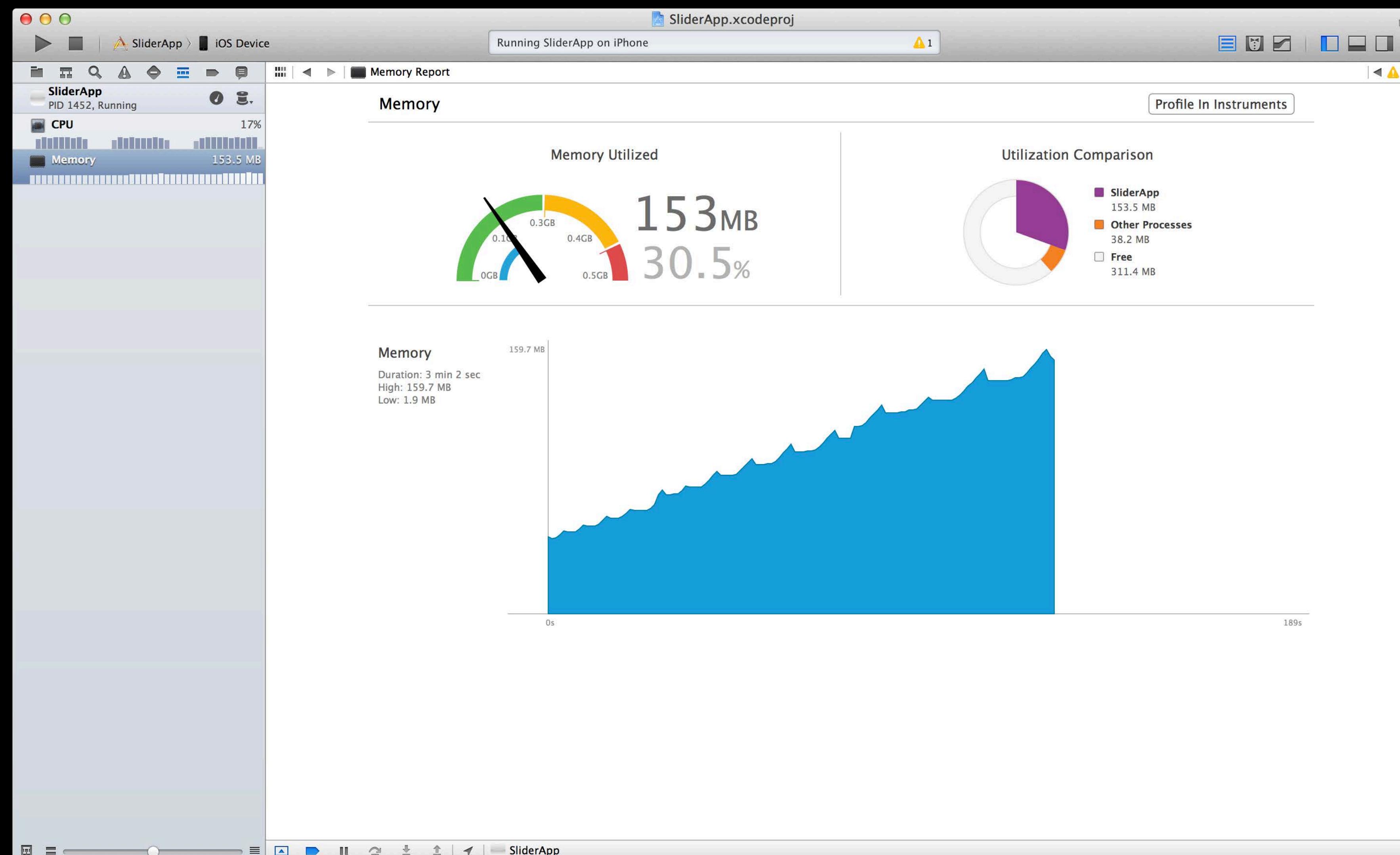
Duration: 3 min 2 sec  
High: 159.7 MB

The screenshot shows the Xcode Memory Report interface for the 'SliderApp' project. At the top, it displays 'SliderApp.xcodeproj' and 'Running SliderApp on iPhone'. Below that, there's a toolbar with icons for Stop, Run, and others, followed by the title 'Memory Report'. The main area is titled 'Memory' and 'Memory Utilized'. It features a large gauge meter with a black needle pointing to 153 MB, which is also displayed as 30.5% of the total memory capacity (0.5GB). The gauge scale marks 0GB, 0.1GB, 0.3GB, 0.4GB, and 0.5GB. To the left of the gauge, there's a vertical bar chart showing memory usage over time, with a peak of 153.5 MB at 17% utilization. At the bottom, there's a summary section with 'Memory' and '159.7 MB', along with a duration of '3 min 2 sec' and a note about a high value of '159.7 MB'.

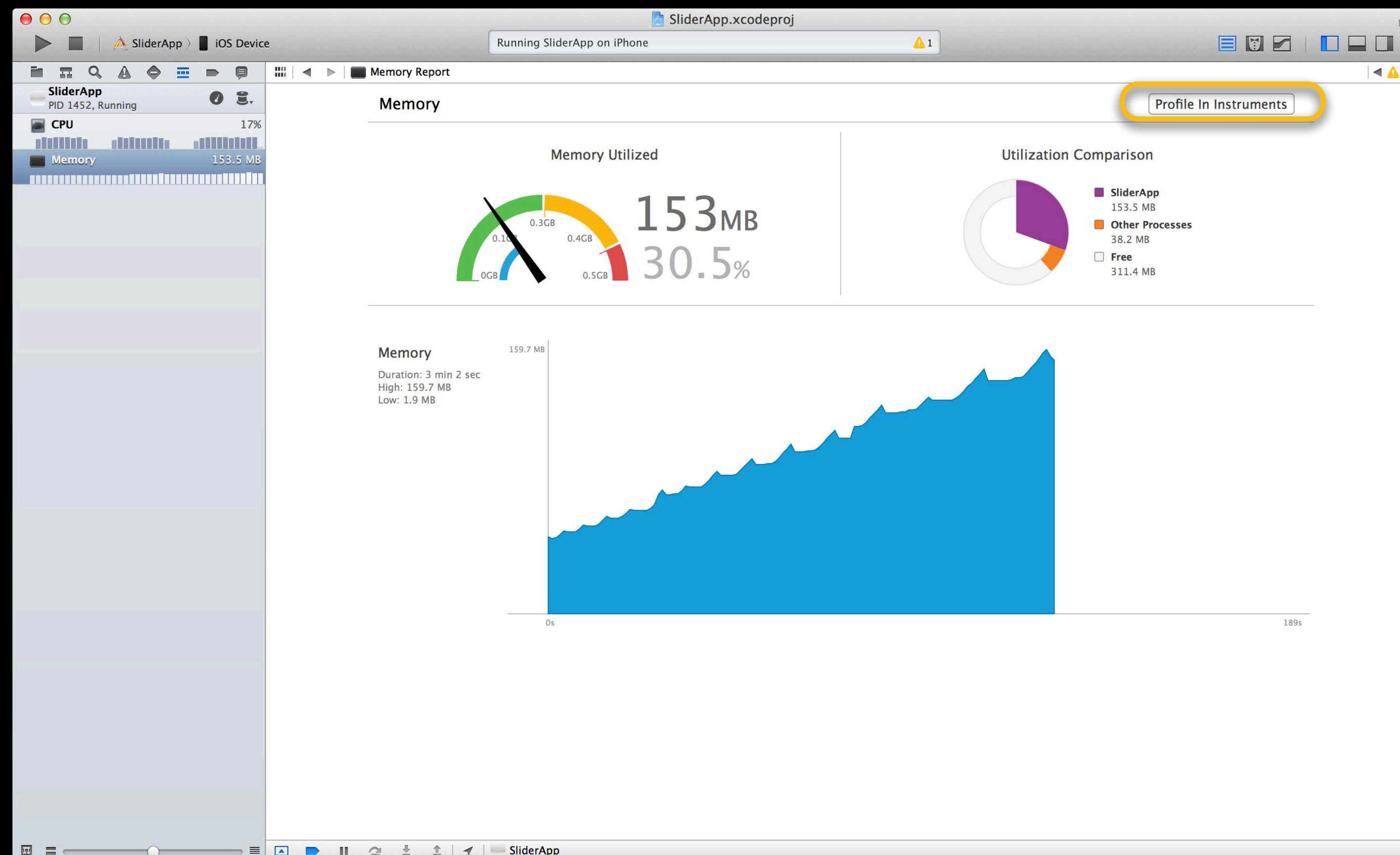
# Memory Report



# Memory Report



# Memory Report



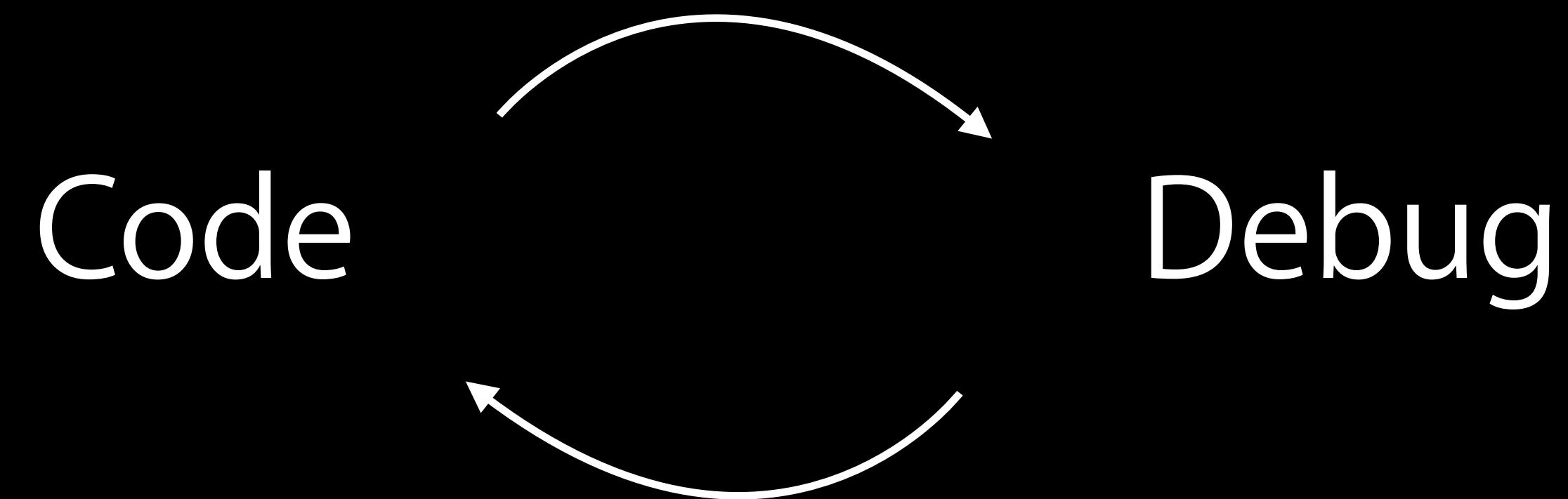
# *Demo*

## Xcode debugging reports

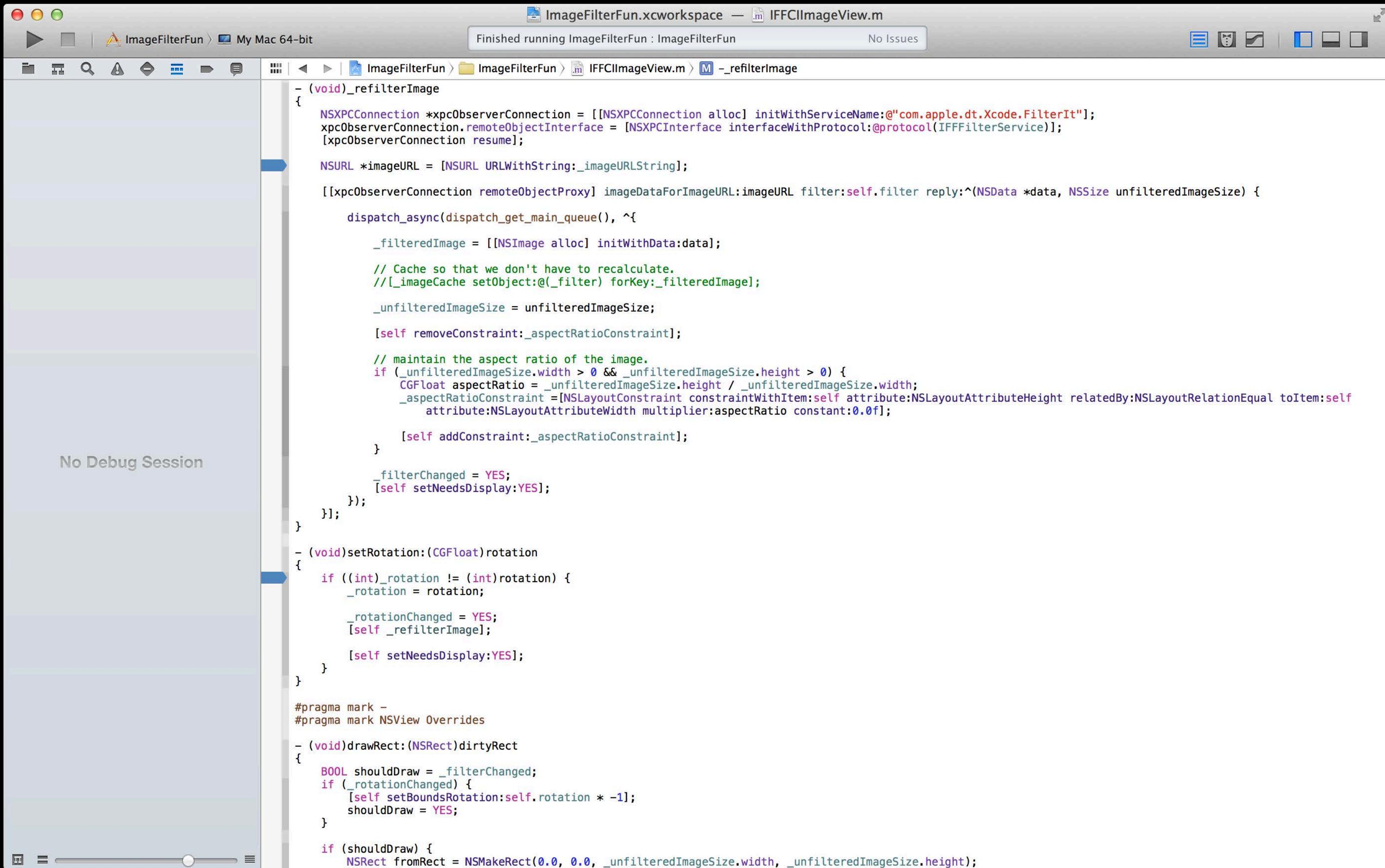
**Han Ming Ong**  
Xcode Debugger UI Engineer

# Data Tips and Quick Look

# Workflows for Debugging



# Debugging Your Code



The screenshot shows the Xcode IDE interface with the following details:

- Title Bar:** ImageFilterFun.xcworkspace — IFFCImageview.m
- Status Bar:** Finished running ImageFilterFun : ImageFilterFun No issues
- Code Editor:** The file IFFCImageview.m is open, showing implementation code for a class. The code includes methods for filtering images and setting rotation, along with various properties and constraints.
- Sidebar:** The sidebar on the left displays "No Debug Session".
- Toolbar:** The top toolbar contains standard Xcode icons for file operations, search, and navigation.

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

    NSURL *imageURL = [NSURL URLWithString:_imageURLString];

    [[xpccObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, NSSize unfilteredImageSize) {
        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.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 NSView 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 a Xcode interface during a debugging session. The main window title is "ImageFilterFun.xcworkspace — IFFCIIImageView.m". The status bar at the top indicates "Running ImageFilterFun : ImageFilterFun" and "No Issues".

The left sidebar displays the project structure under "ImageFilterFun" and monitoring for CPU (0%) and Memory (36 MB). The "Thread 1" section shows the call stack for the current thread:

- 0 -[IFFCIIImageView \_refilterImage]
- 1 -[IFFCIIImageView setRotation:]
- 2 -[NSObject(NSKeyValueCoding)…]
- 3 -[NSObjectParameterBinder \_up…]
- 4 NSKeyValueNotifyObserver
- 9 -[NSObject(NSKeyValueCoding)…]
- 10 -[NSBinder \_setValue:forKeyPa…]
- 24 NSApplicationMain
- 25 main
- 26 start

The code editor shows the implementation of `_refilterImage`. A breakpoint is set at the start of the function, indicated by a red dot and the text "Thread 1: breakpoint 1.1". The code uses NSXPCConnection to communicate with the Xcode FilterIt service.

The bottom right corner shows the LLDB debugger console with the following commands and output:

```
(lldb) po self
<IFFCIIImageView: 0x10052b380>
(lldb) p _imageURLString
$8 = 0x000000100519f10 @"http://images.apple.com/ipad/overview/images/hero.png"
(lldb)
```

# Debugging Your Code

The screenshot shows the Xcode interface during a debugging session. The main window title is "ImageFilterFun.xcworkspace — IFFCImageView.m". The status bar at the top indicates "Running ImageFilterFun : ImageFilterFun" and "No Issues". The left sidebar displays the project structure under "ImageFilterFun" and monitoring for CPU (0%) and Memory (36 MB). The central pane shows the code for `IFFCImageView.m`. A breakpoint is set at line 1.1, indicated by a red dot and the text "Thread 1: breakpoint 1.1" in the gutter. The stack trace for Thread 1 shows the call stack from step 0 to step 26, starting with `-[IFFCImageview _refilterImage]`. Other threads listed are Thread 3, Thread 16, Thread 17, and Thread 18. The bottom navigation bar includes standard Xcode icons for file operations.

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

    NSURL *_imageURL = [NSURL URLWithString:_imageURLString];
    [[xpccObserverConnection remoteObjectProxy] imageDataForImageURL:imageURL filter:self.filter reply:^(NSData *data, NSSize unfilteredImageSize) {

        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.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 NSView Overrides

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

    if (shouldDraw) {
        // Drawing code here
    }
}
```

# Data Tips

```
_unfilteredImageSize = size;
```



► (width=1216, height=811)



# Data Tips

```
_unfilteredImageSize = size;
```

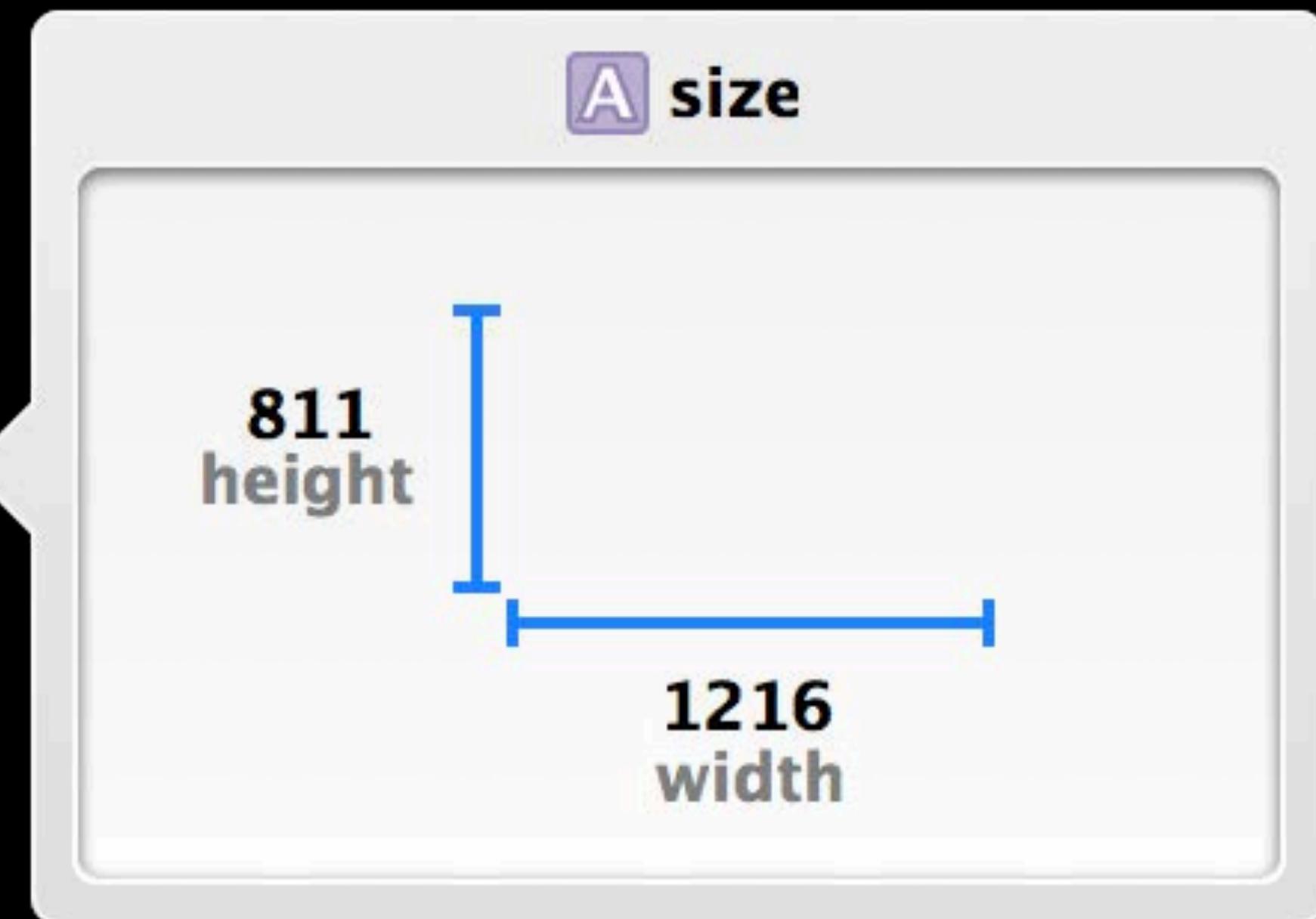
► (width=1216, height=811)

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

# Quick Look of Variables

```
_unfilteredImageSize = size;
```

► (width=1216, height=811)



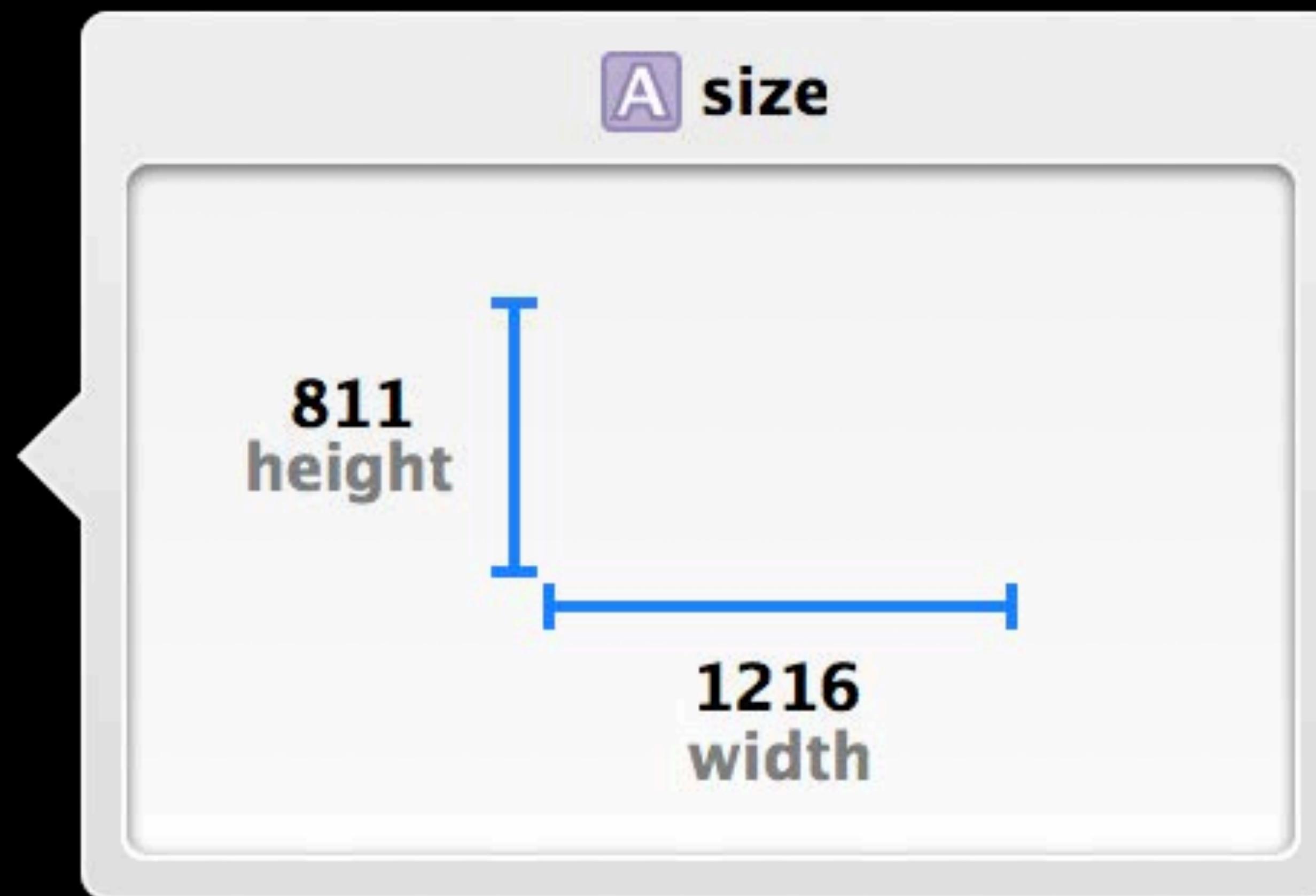
# *Demo*

## Data Tips and Quick Look

**Christopher Friesen**  
Xcode Debugger UI Engineer

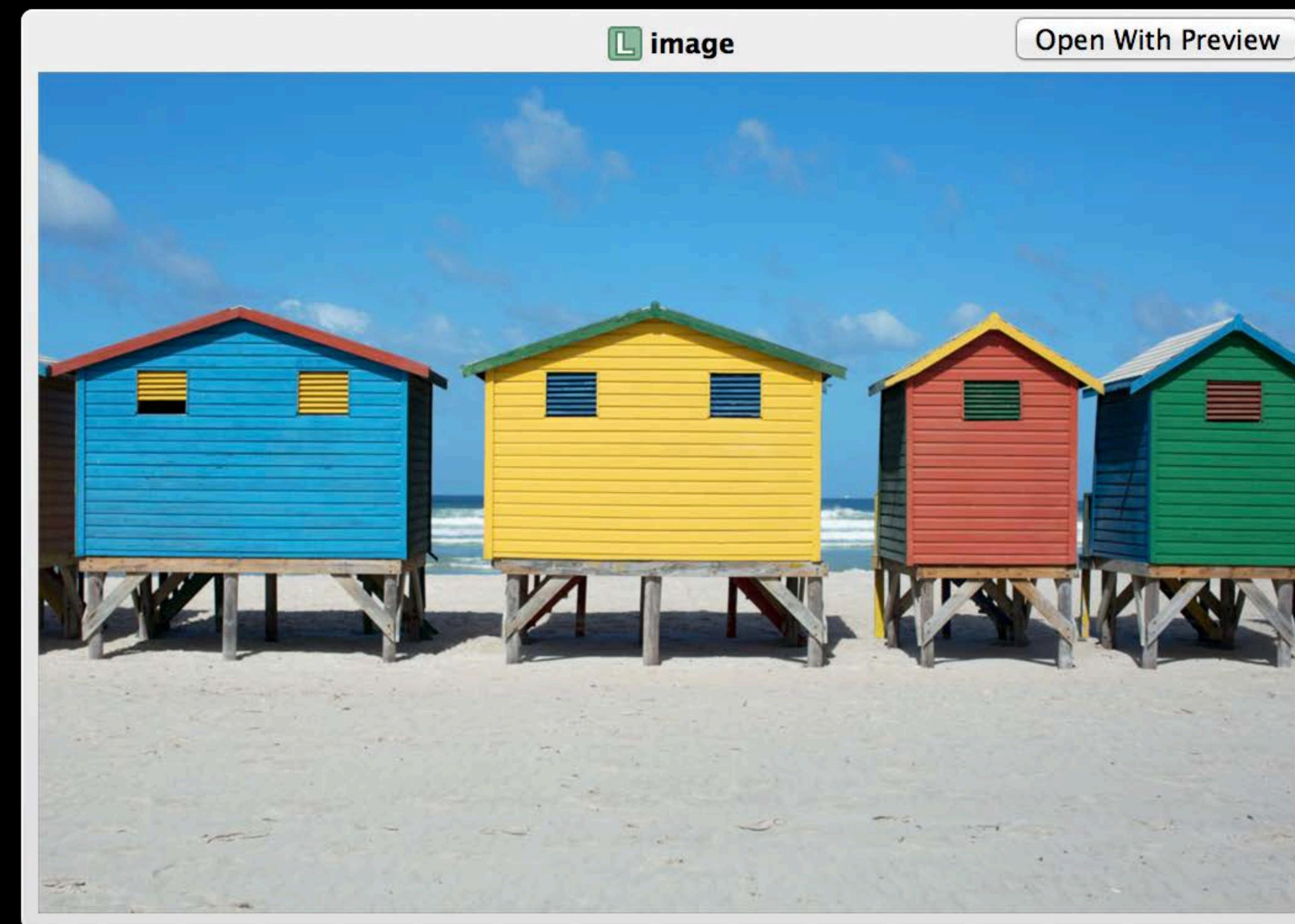
# Quick Look in Xcode

## NSSize and CGSize



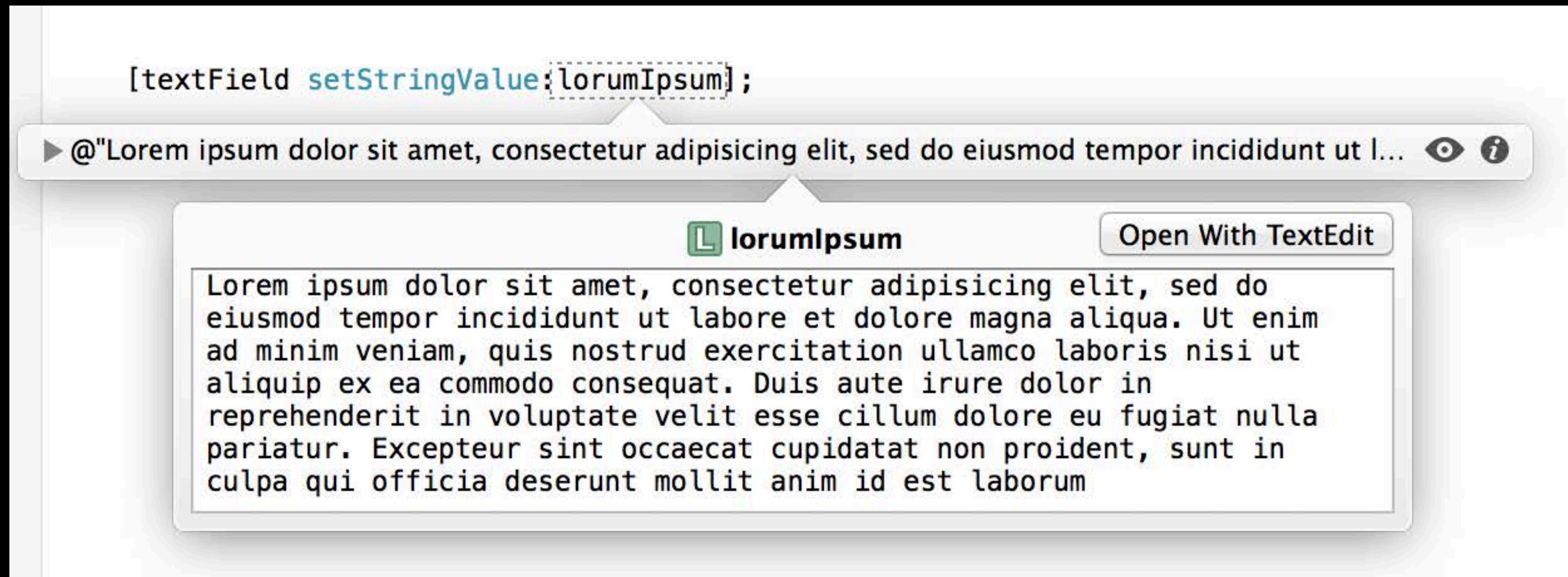
# Quick Look in Xcode

## UIImage, NSImage, CGImageRef, and CIIimage



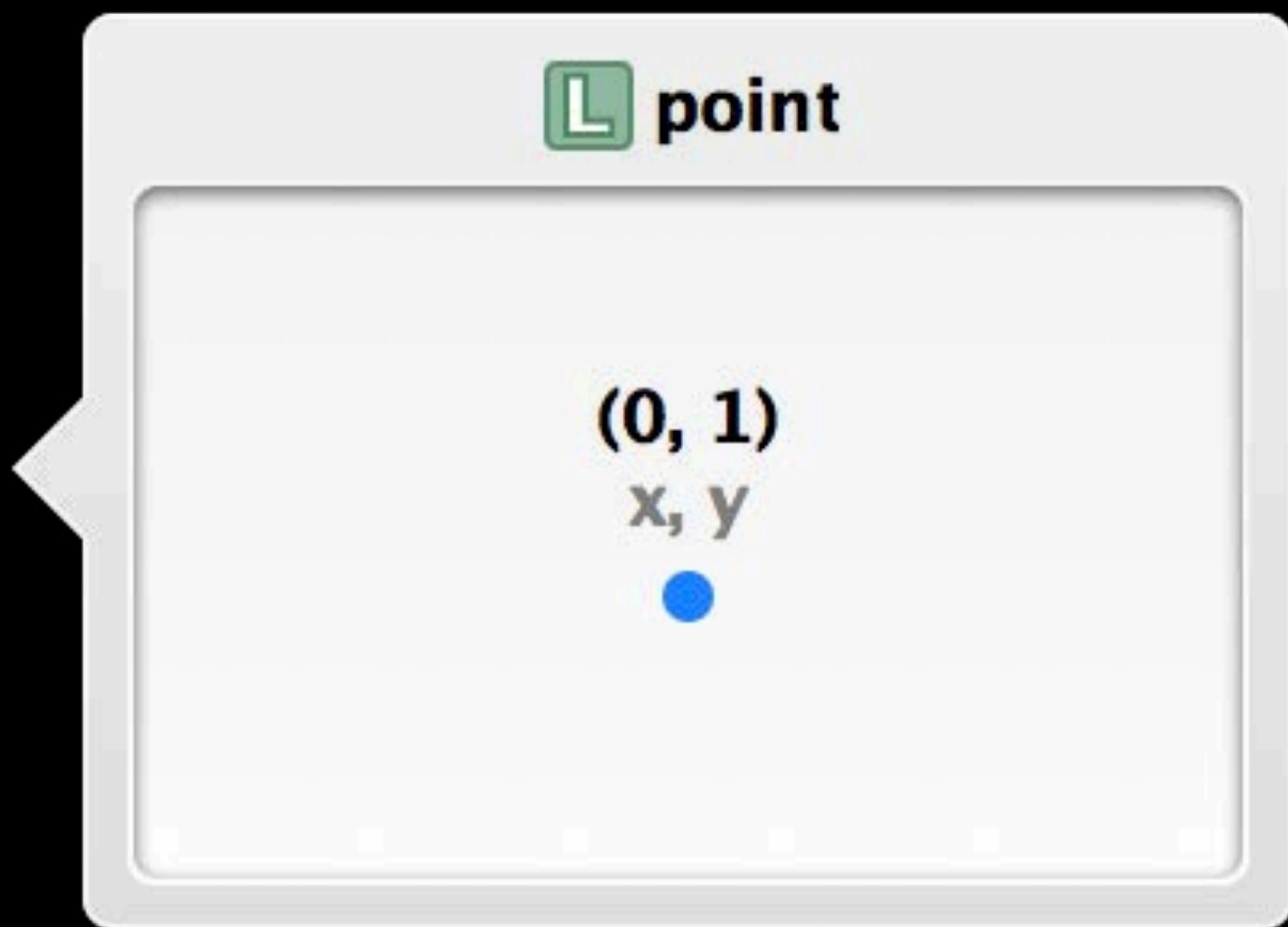
# Quick Look in Xcode

NSString and NSAttributedString, including long text



# Quick Look in Xcode

NSPoint and CGPoint



# Quick Look in Xcode

## NSData

L data

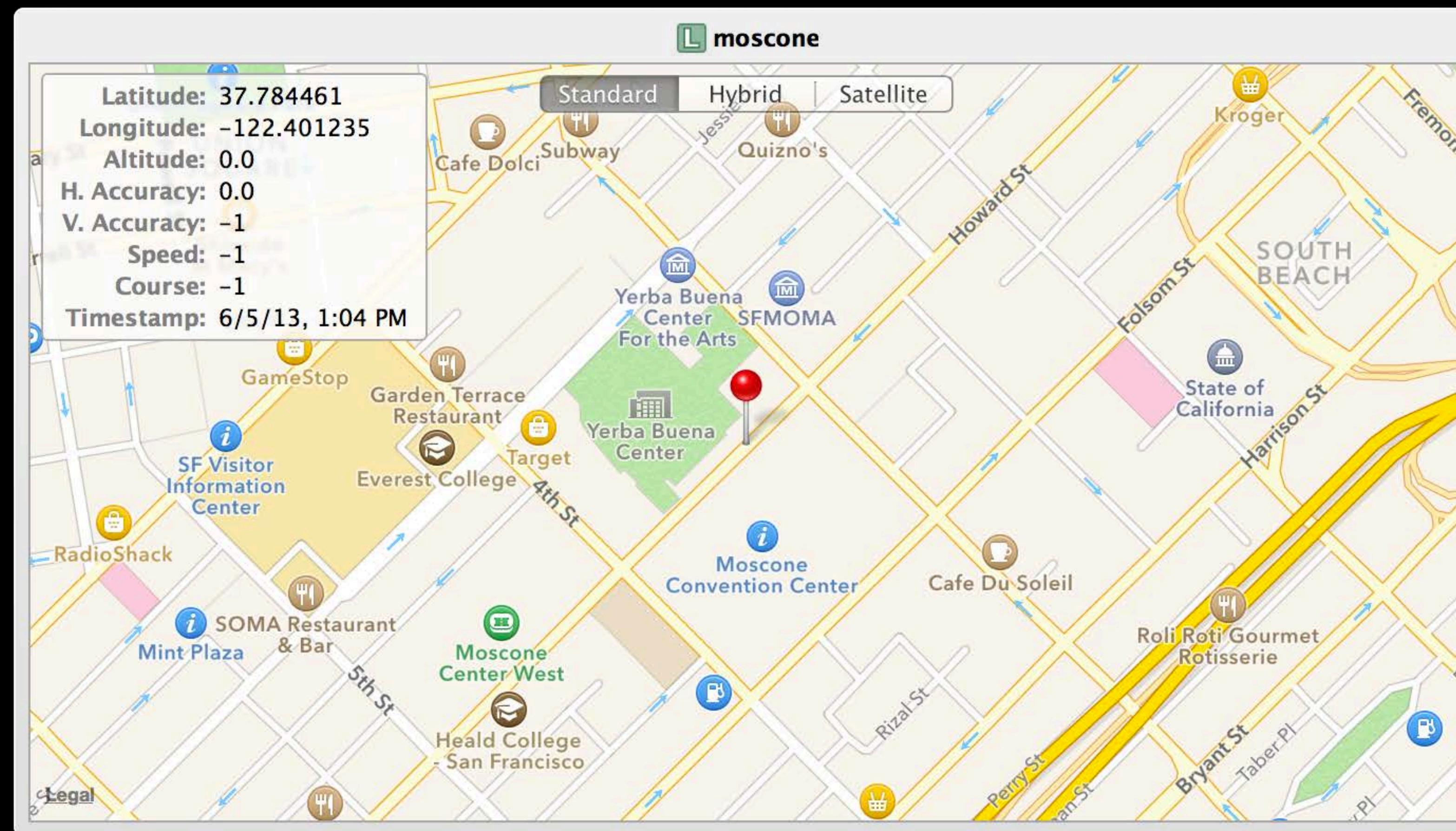
4519206912	3C 3F 78 6D 6C 20 76 65 72 73 69 6F 6E 3D 22 <?xml version="
4519206927	31 2E 30 22 20 65 6E 63 6F 64 69 6E 67 3D 22 1.0" encoding="
4519206942	55 54 46 2D 38 22 3F 3E 0A 3C 21 44 4F 43 54 UTF-8"?>. DOCT</td
4519206957	59 50 45 20 70 6C 69 73 74 20 50 55 42 4C 49 YPE plist PUBLI
4519206972	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 >..<key>ImageUR
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/Library/Developer/
4519207182	72 61 72 79 2F 44 65 76 65 6C 6F 70 65 72 2F rary/Developer/

Bytes 1 – 512 of 7196589

Memory Page

# Quick Look in Xcode

## CLLocation, CLLocationCoordinate2D



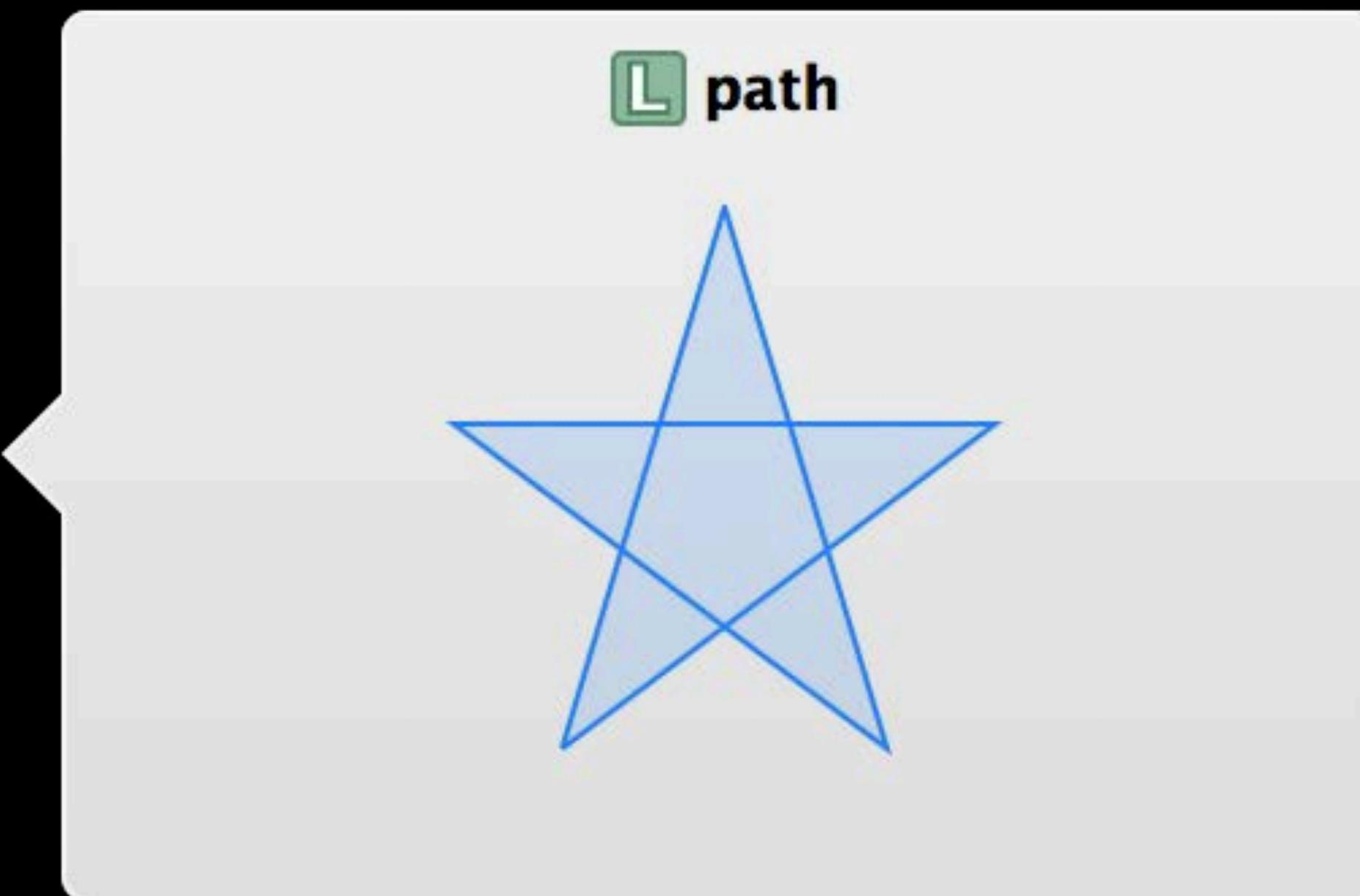
# Quick Look in Xcode

## UIColor, NSColor, and CGColor



# Quick Look in Xcode

UIBezierPath and NSBezierPath

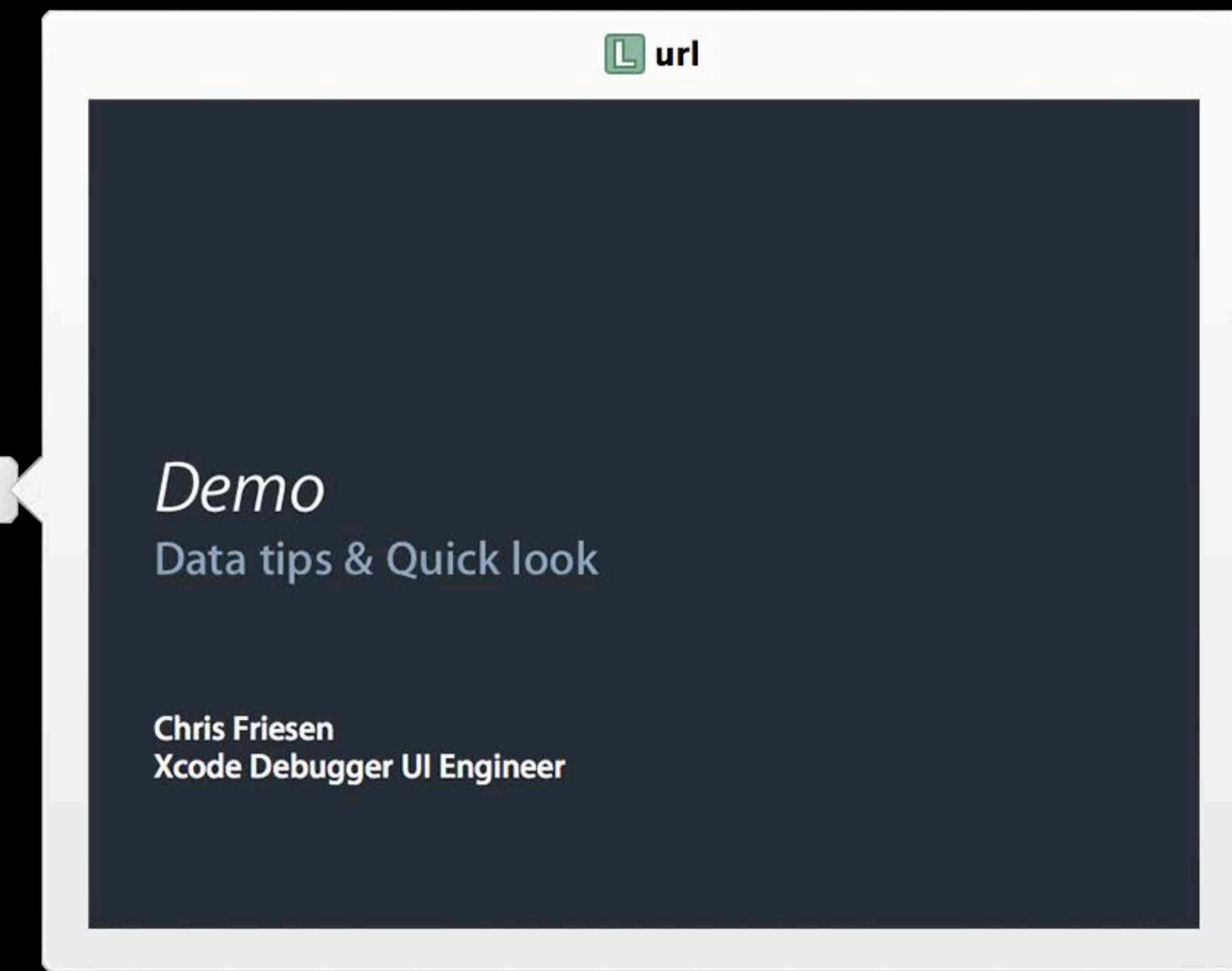


# Quick Look in Xcode

NSURL

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

►(NSURL \*) 0x101838bf0 ⚡ ⓘ

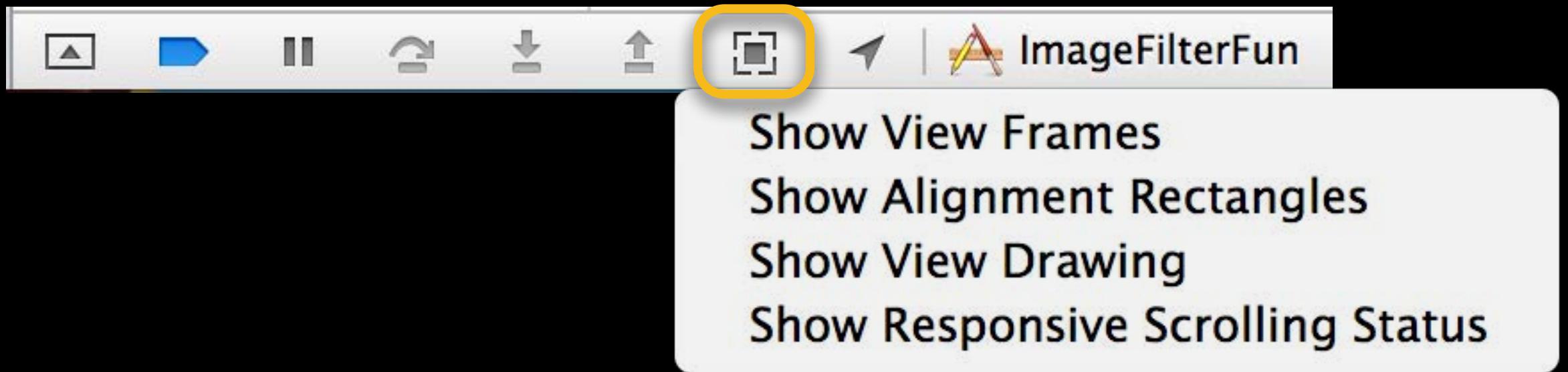




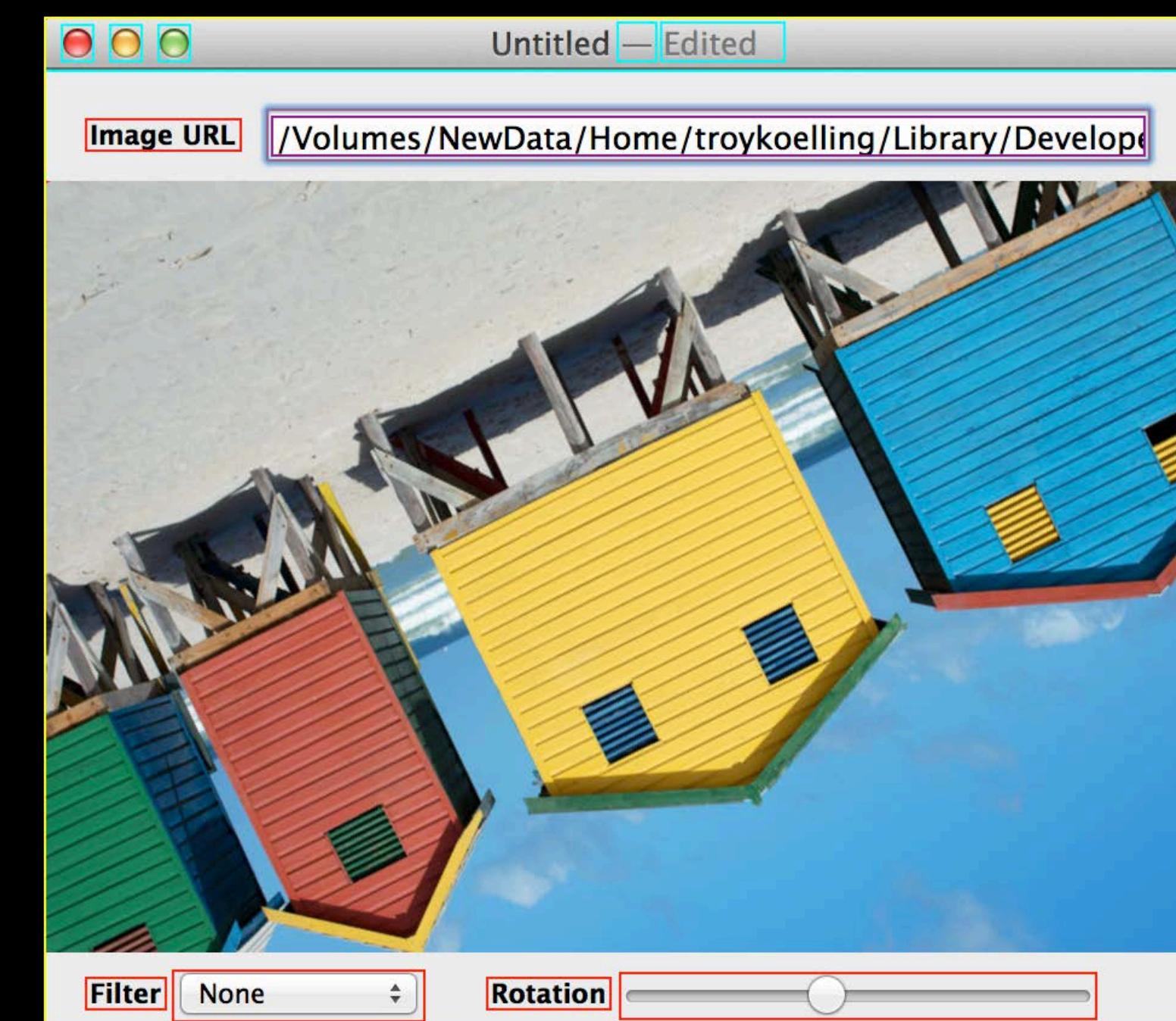
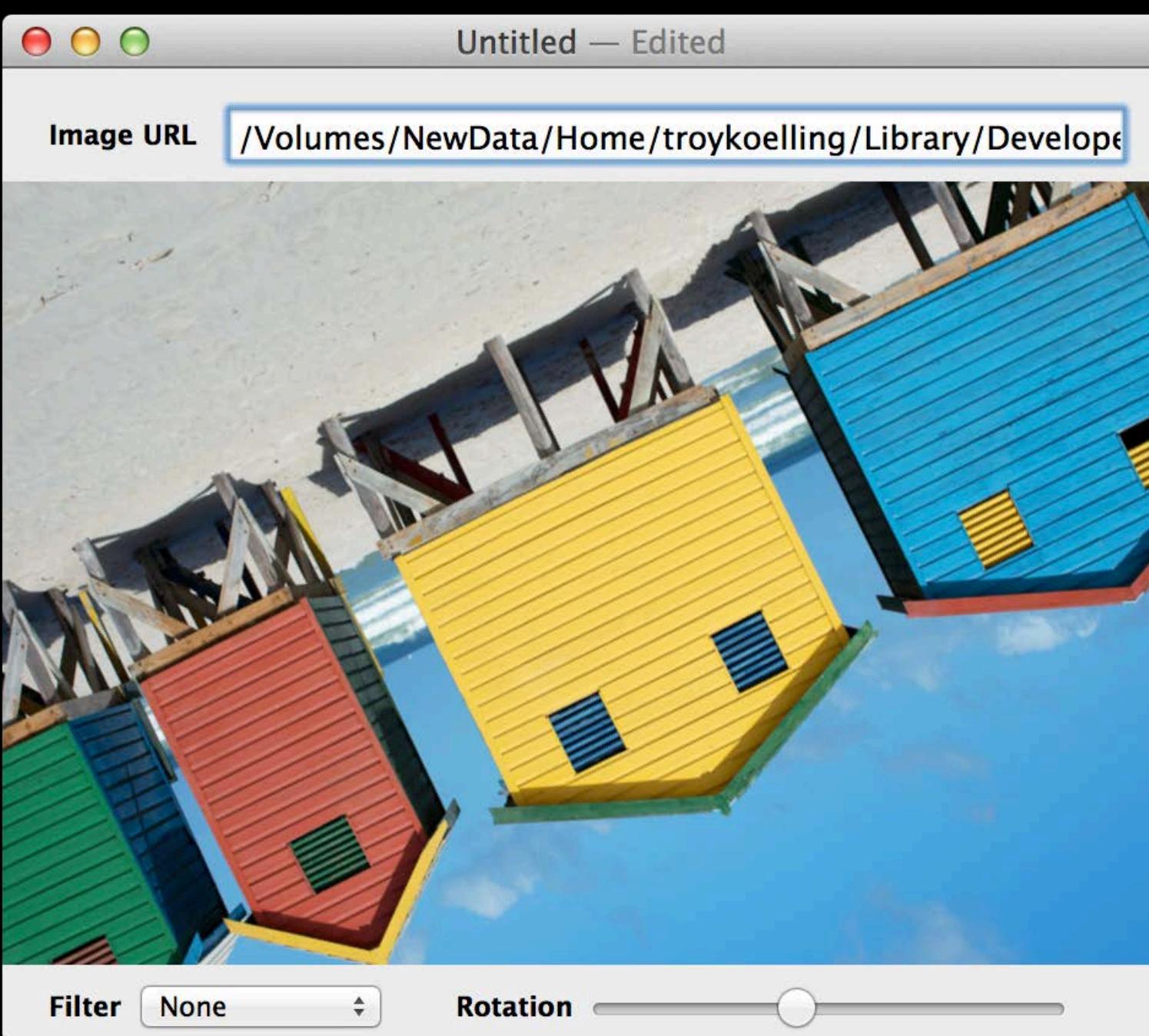
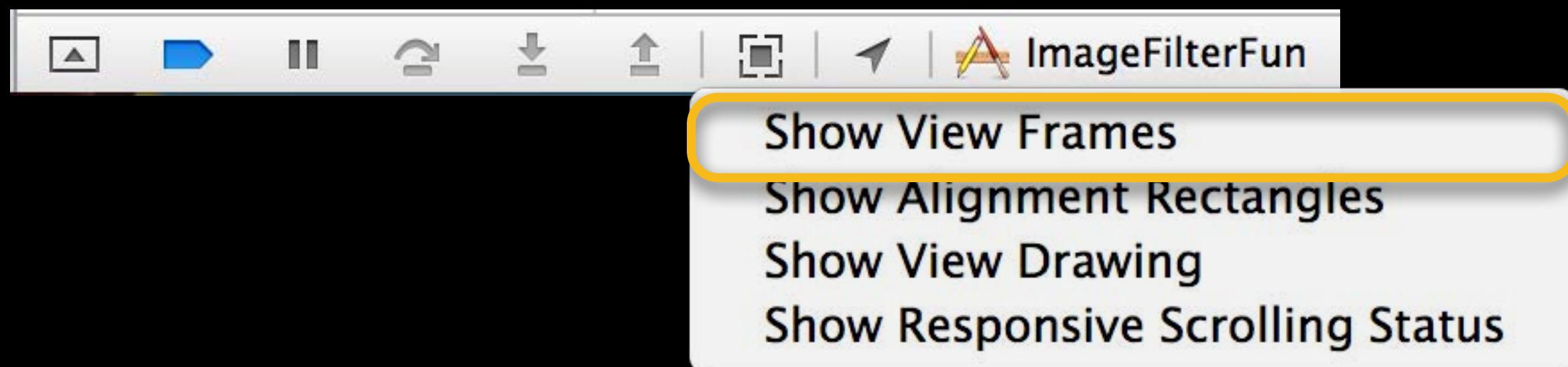
# Drawing Diagnostics



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



# Drawing Diagnostics (AppKit)



# Printing a UIView's Hierarchy



(lldb)

# Printing a UIView's Hierarchy



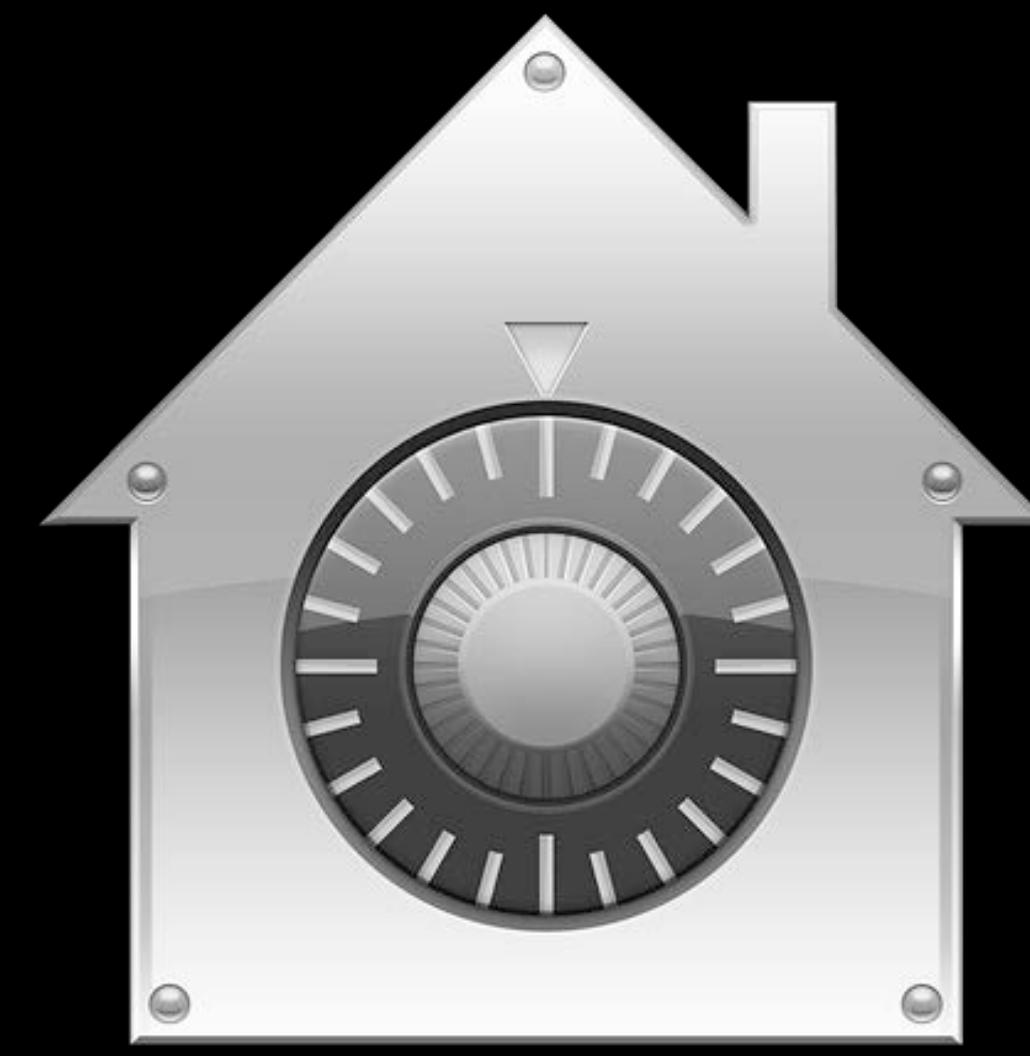
```
(lldb) po [self.view recursiveDescription]
(id) $1 = 0x07b55860 <UIView: 0x6e60e90; frame = (20 20; 280 224); auto
| <UIRoundedRectButton: 0x6e5f830; frame = (97 90; 86 44); opaque = NO;
|   | <UIGroupTableViewCellBackground: 0x6e60150; frame = (0 0; 86 44);
|   | <UIImageView: 0x6e60b40; frame =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

(lldb)

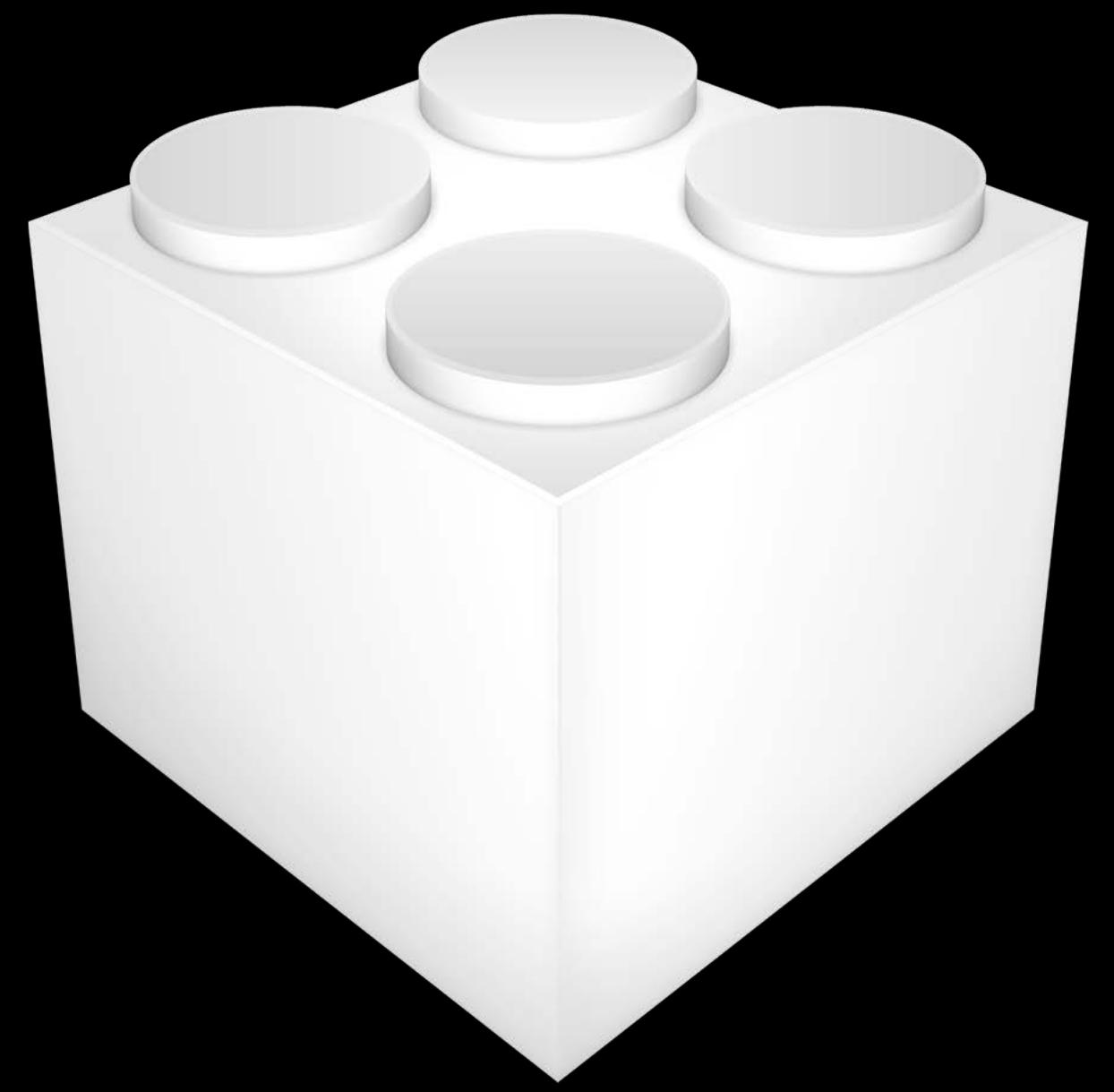
# XPC

## Debugging with Xcode 5

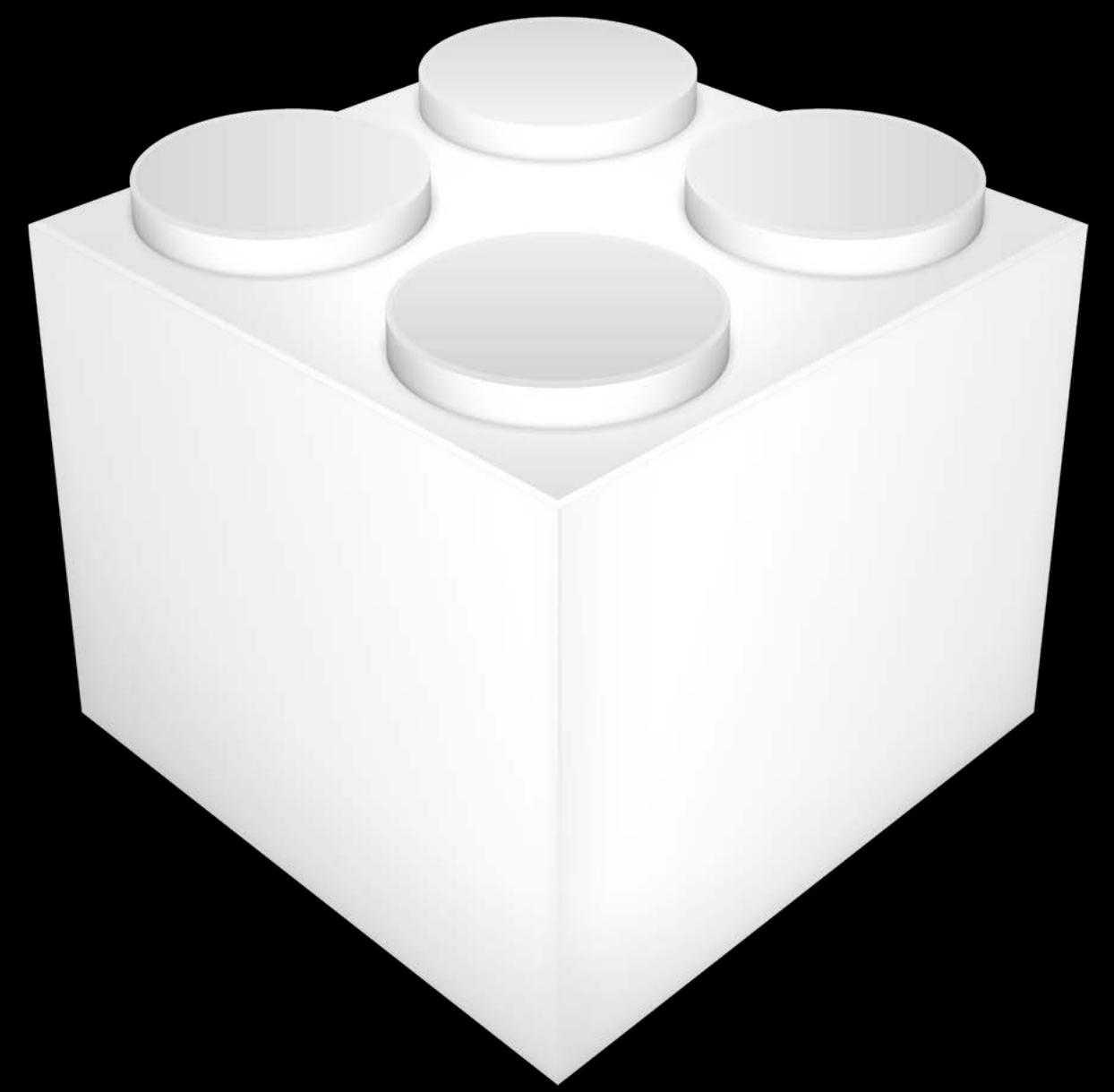
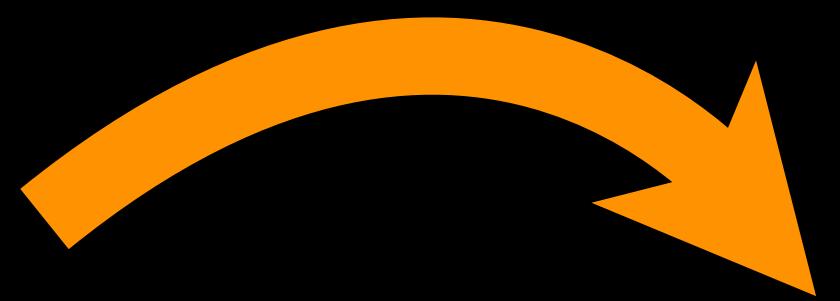
# XPC



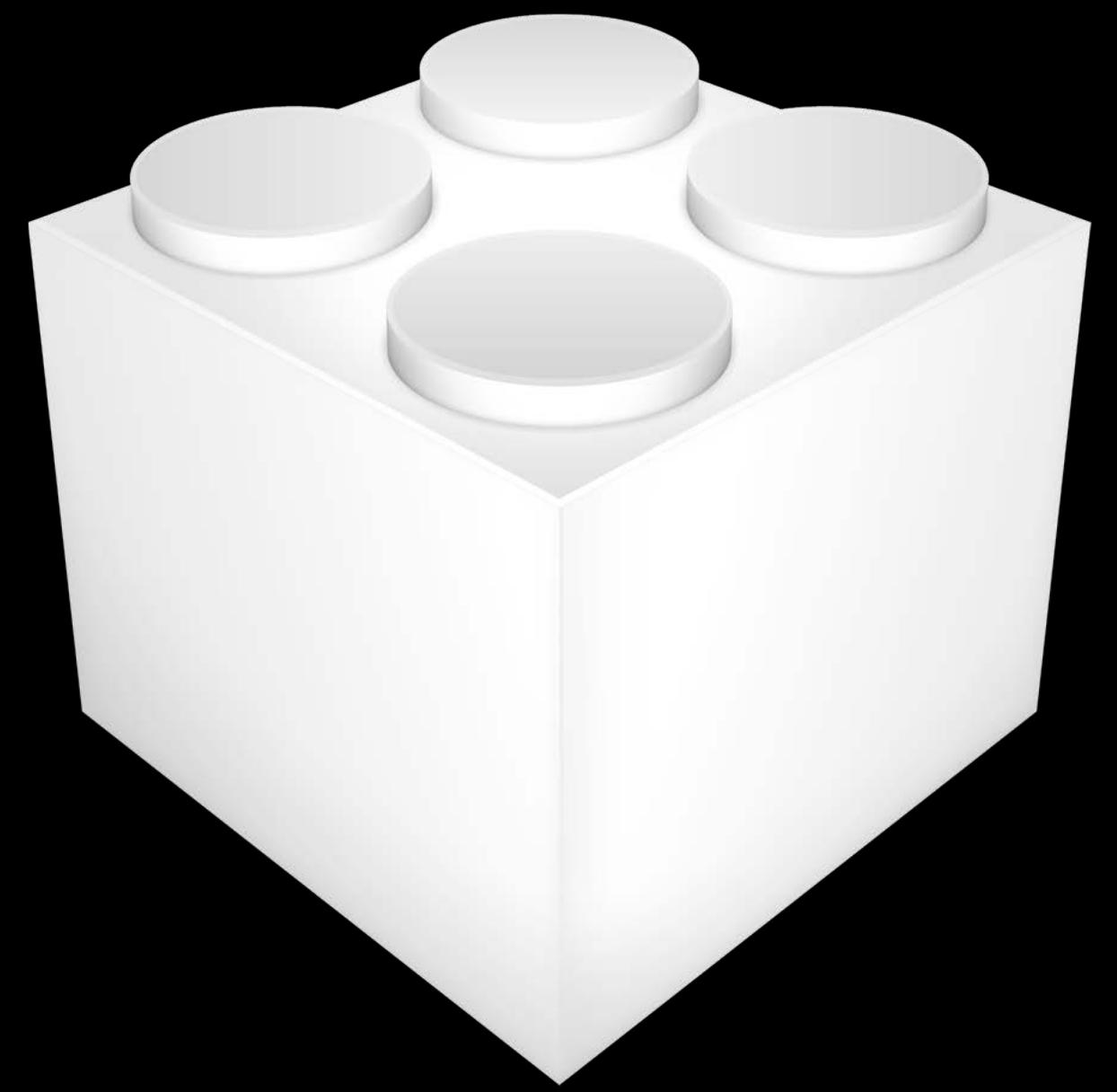
# XPC



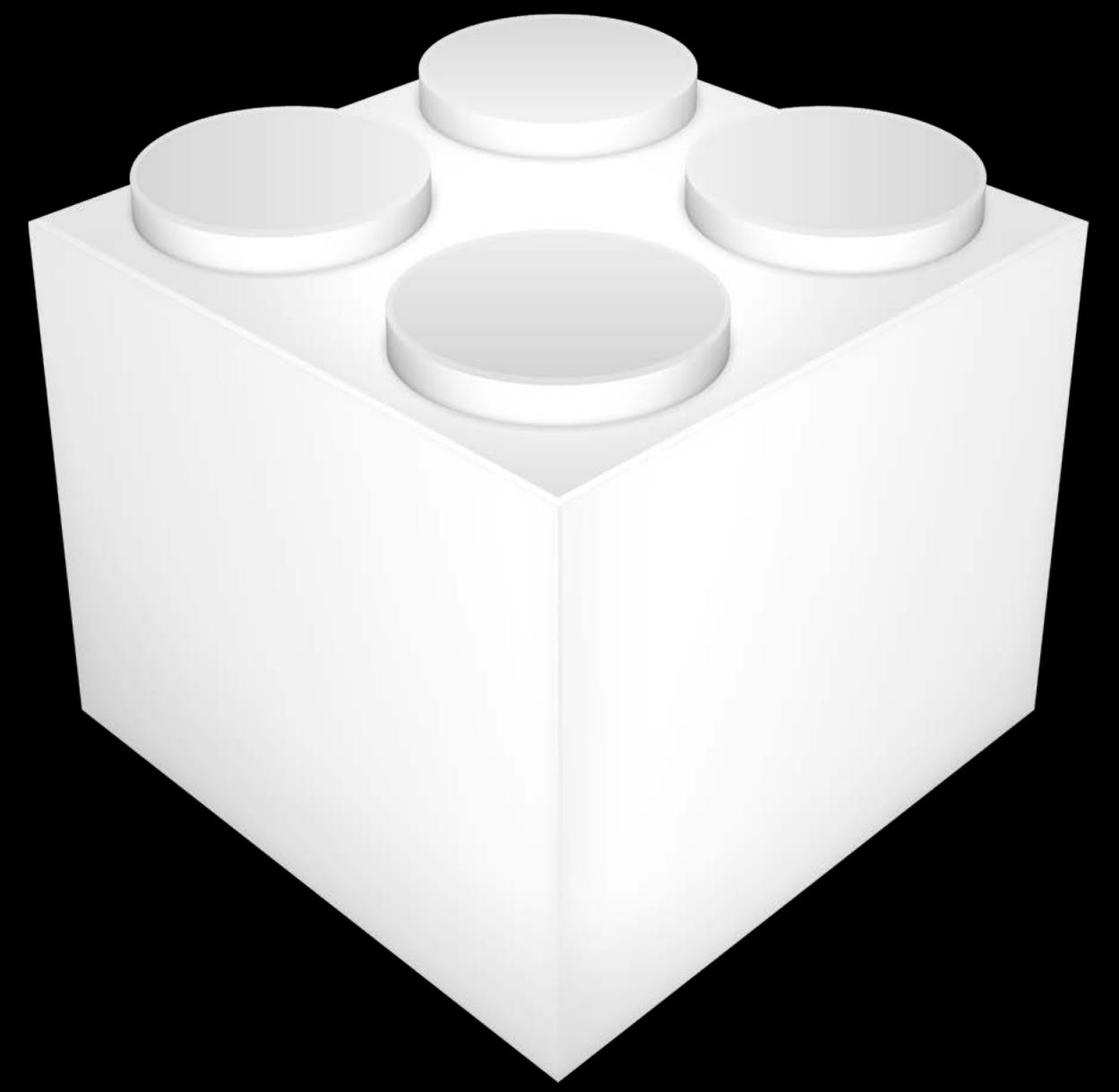
# XPC



# XPC



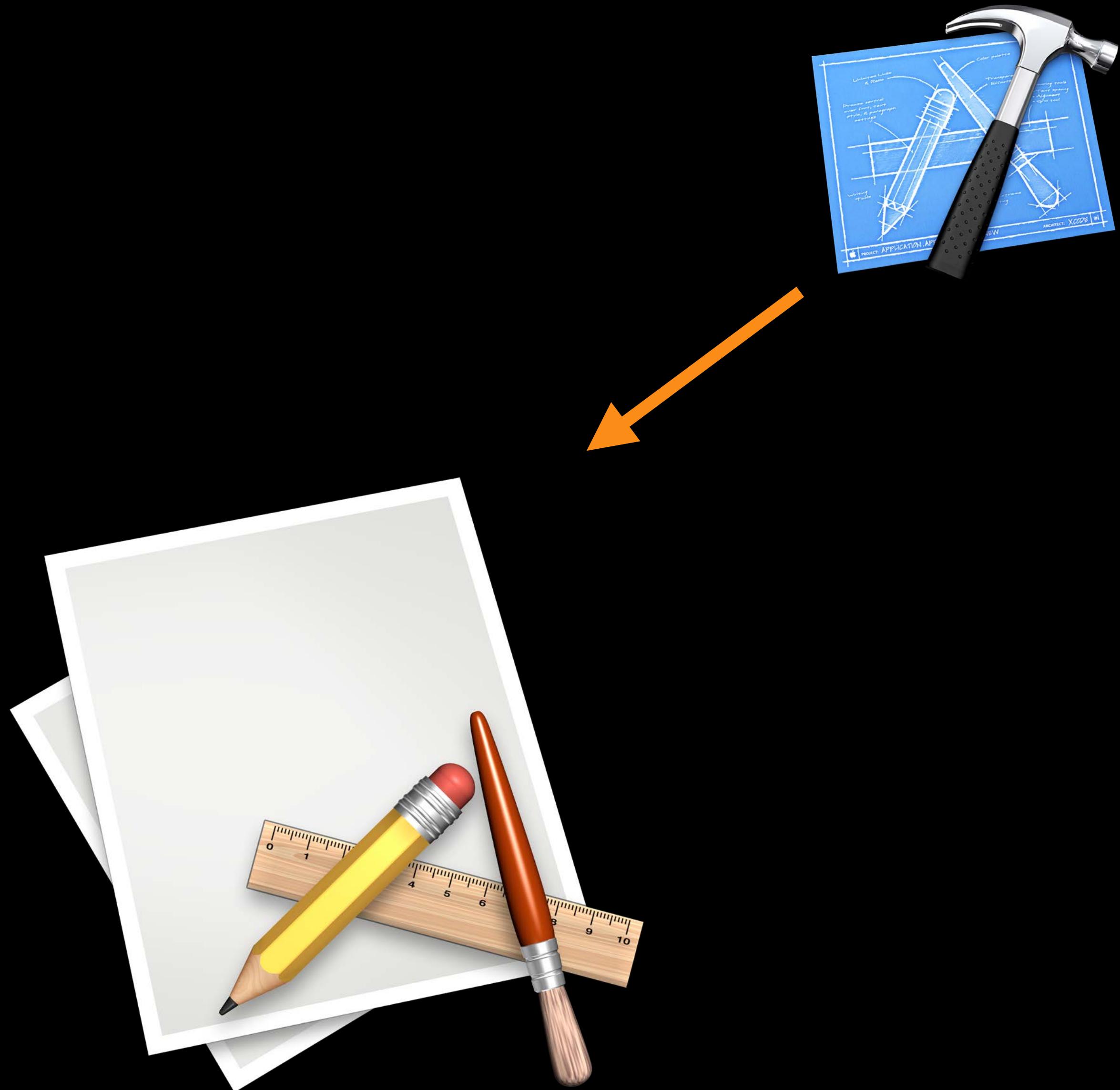
# XPC



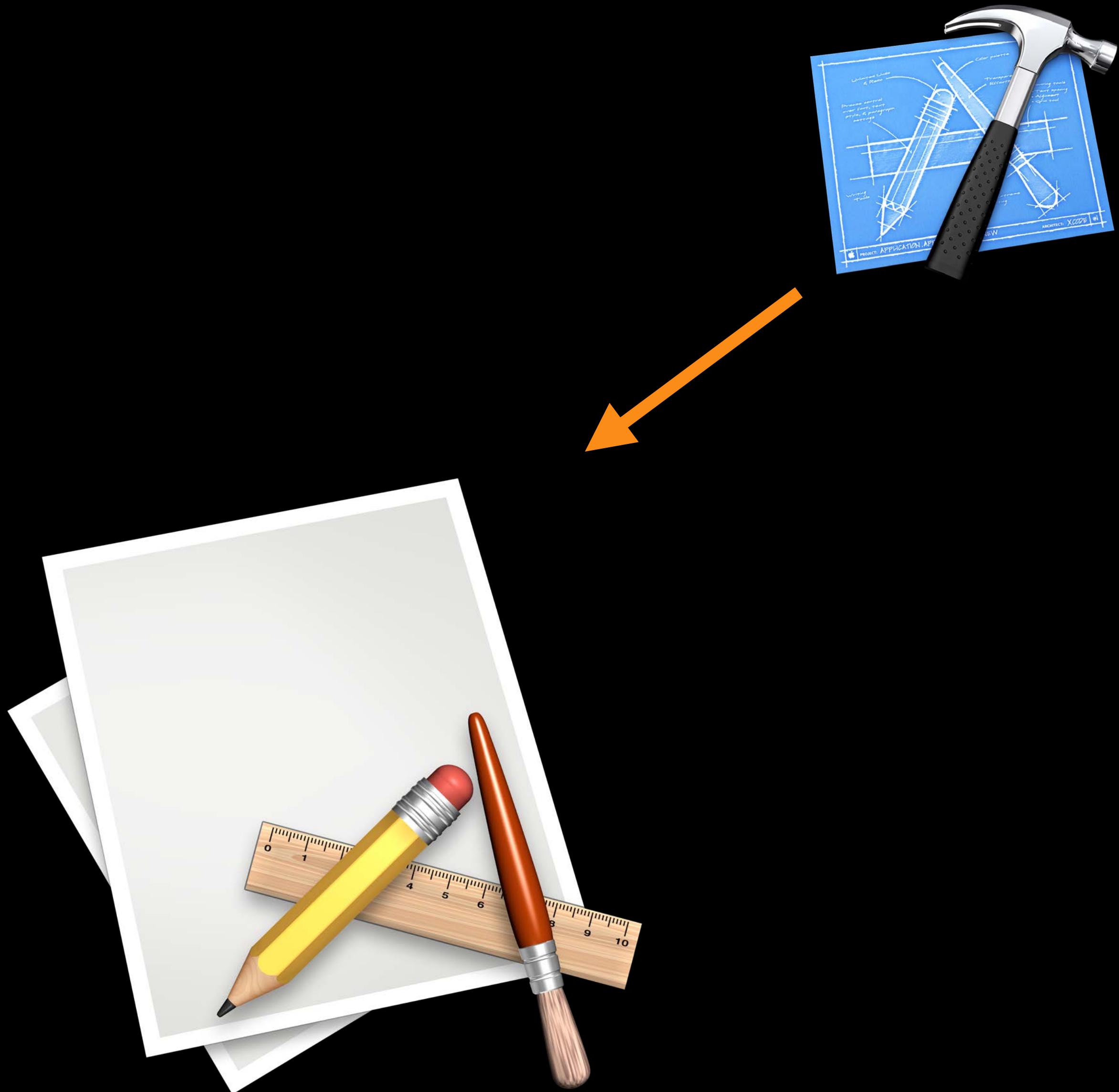
# XPC



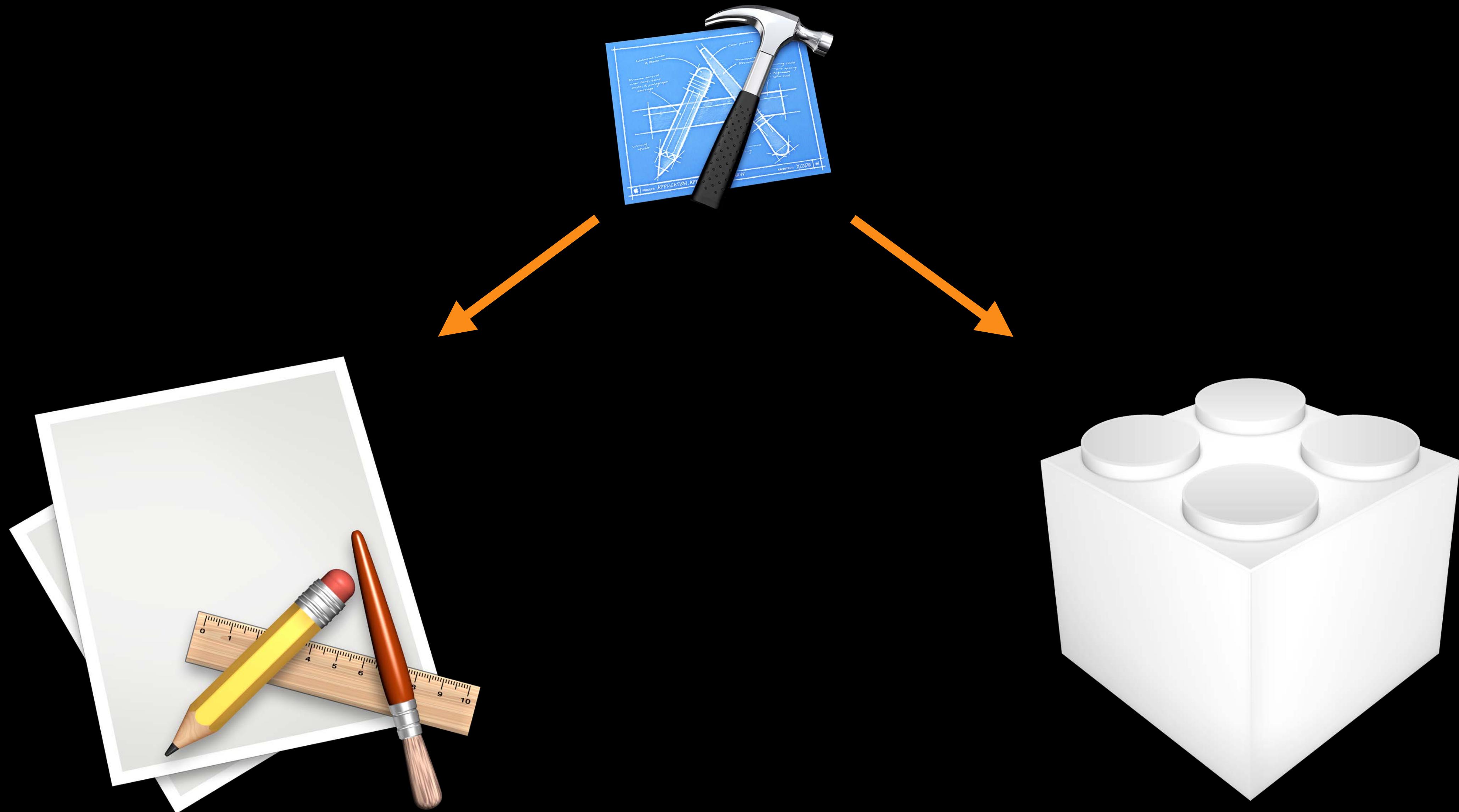
# XPC



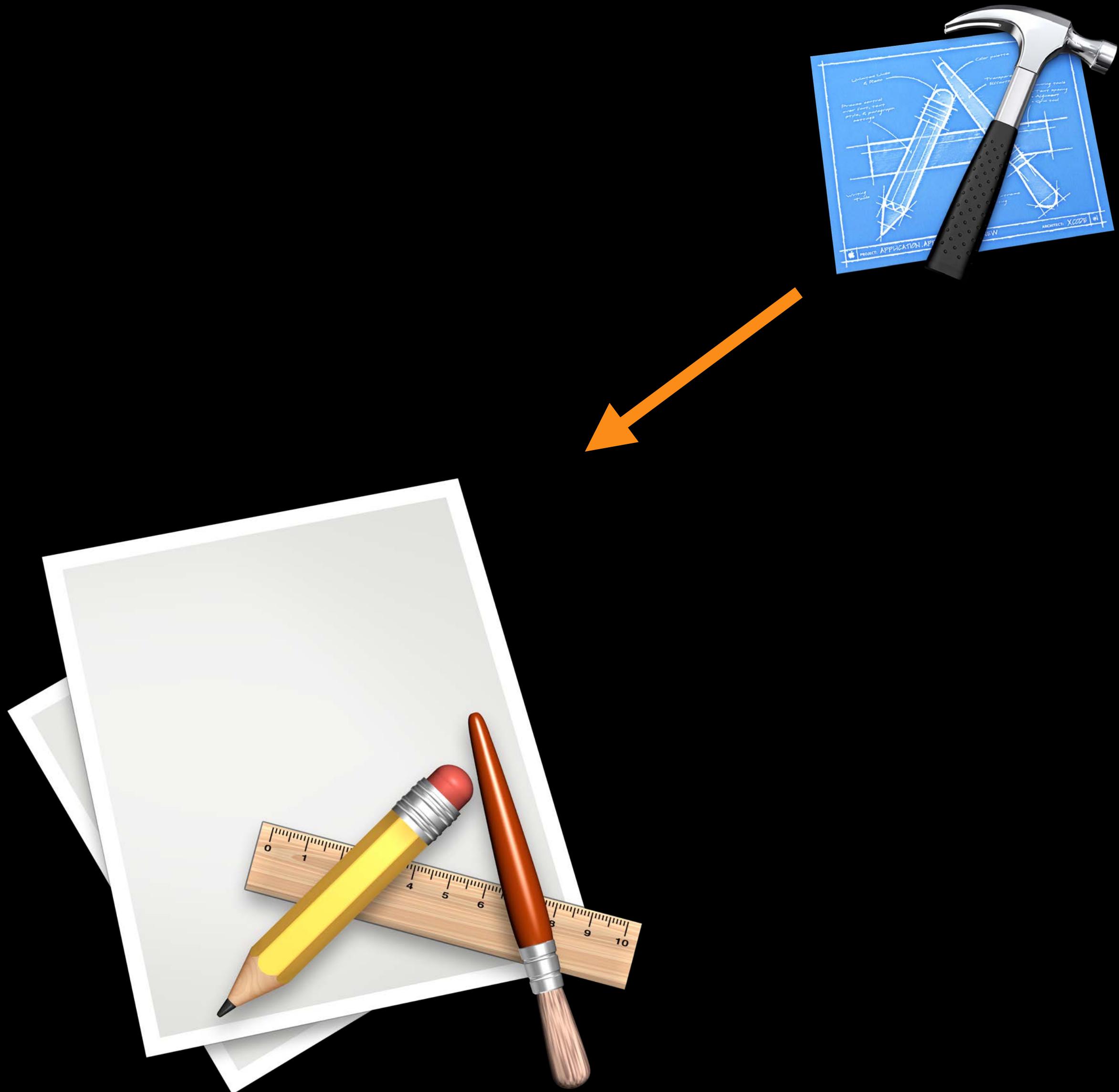
# XPC



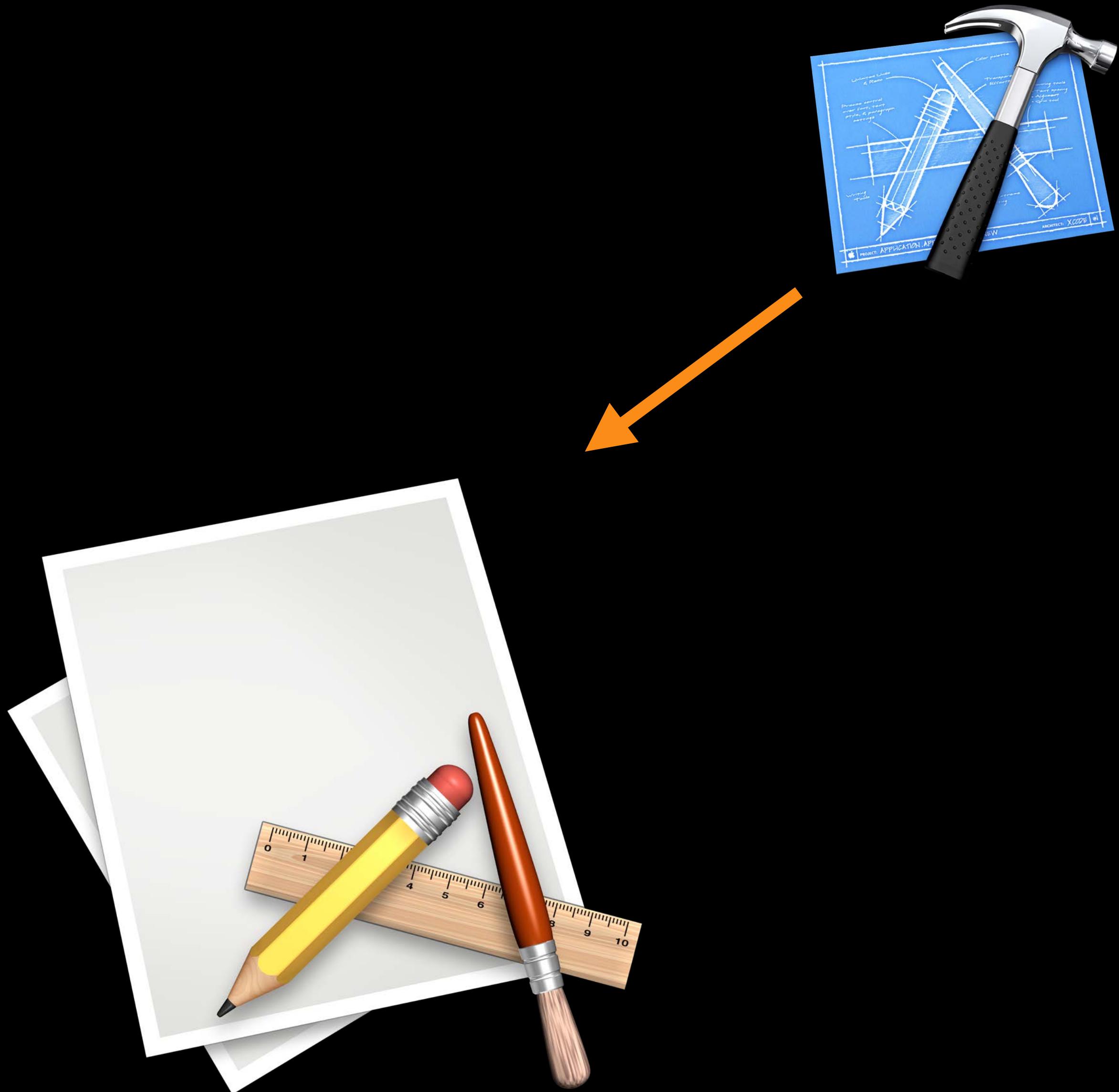
# XPC



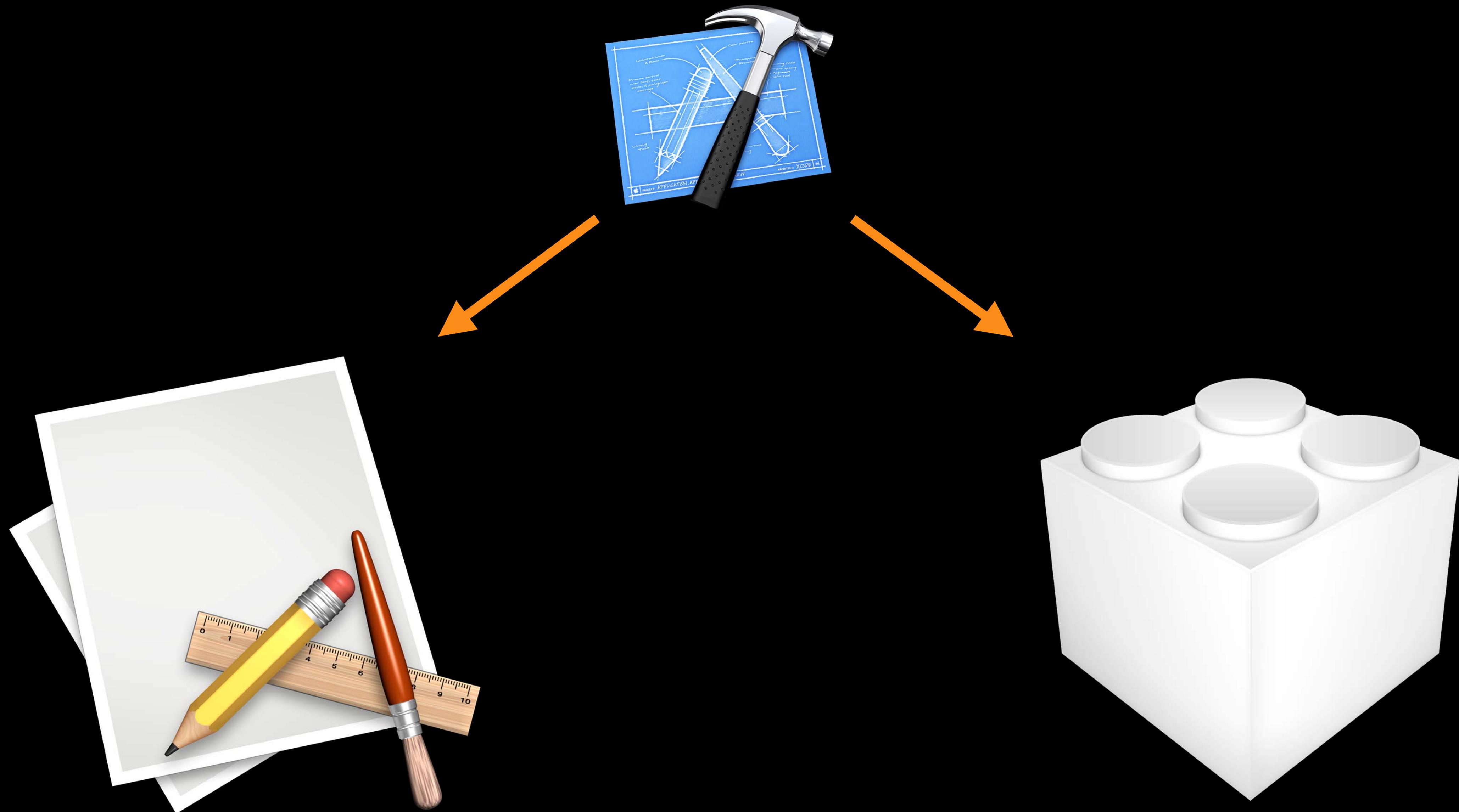
# XPC



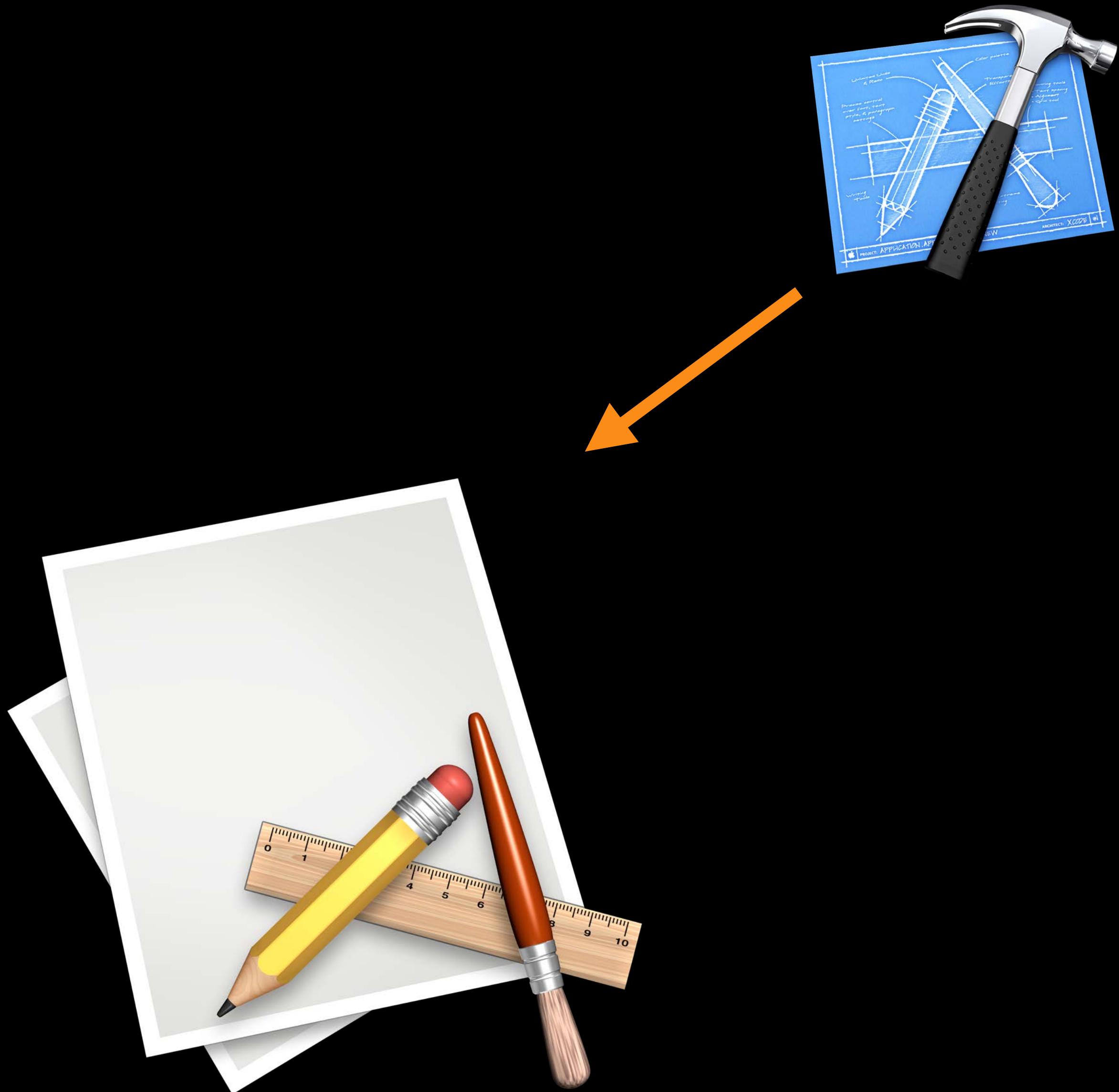
# XPC



# XPC



# XPC



# *Demo*

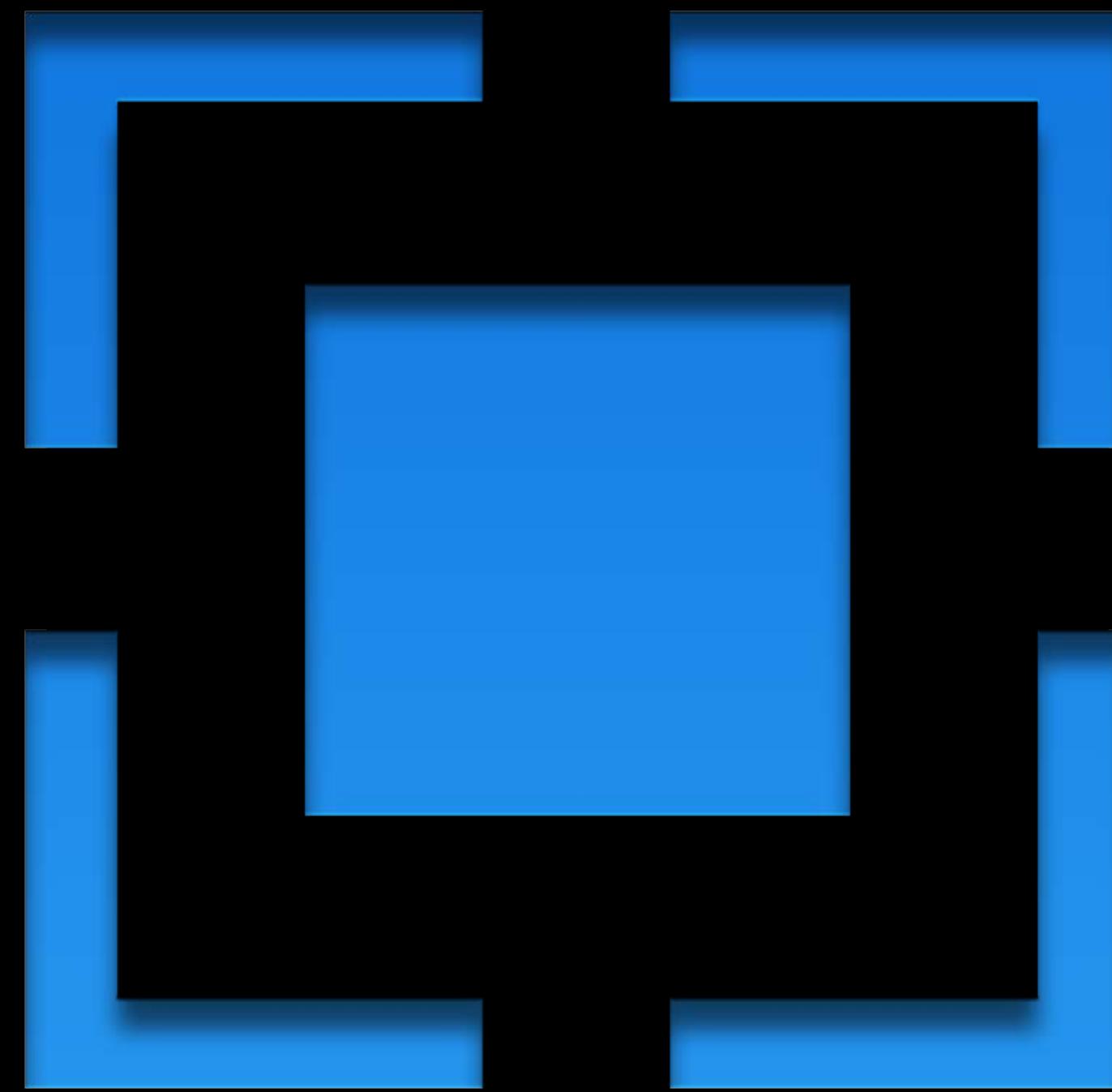
## XPC debugging

**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	

