Introduction to PyObjC

Author

Bob Ippolito

Conference

PyCon DC, March 2005

Intended Audience

- Python developers using Mac OS X 10.3 or later
- Spies from the Linux and Win32 camps
- Hopefully a GNUstep porter/maintainer

Topics

- Installing PyObjC
- Why Bother?
- Interface Builder
- Objective-C Primer
- Crossing the Bridge
- Your First Application
- Help!
- Who's Using This Stuff?

Installing PyObjC

Install Xcode:

http://developer.apple.com/

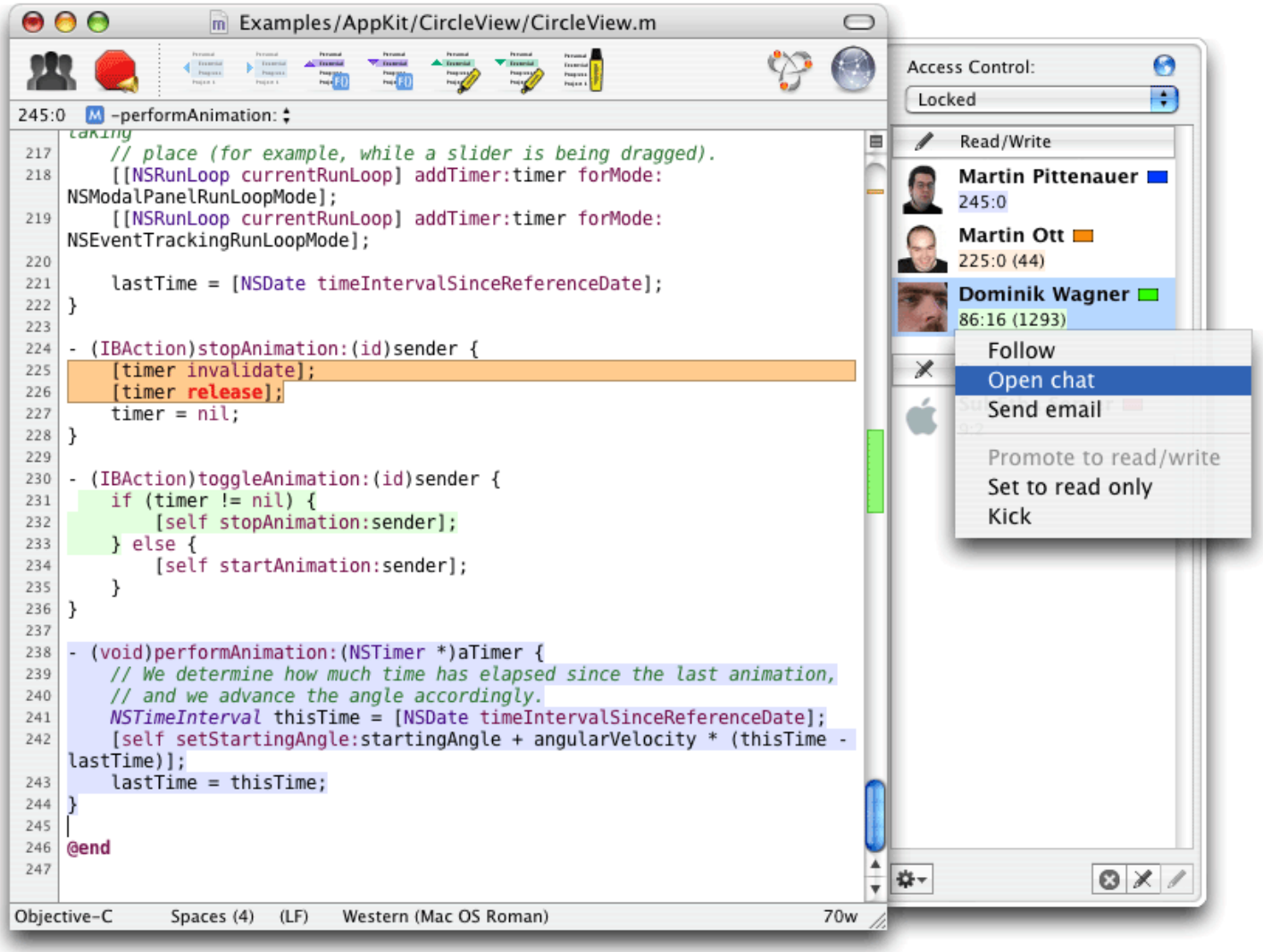
Install PyObjC:

http://pyobjc.sourceforge.net/

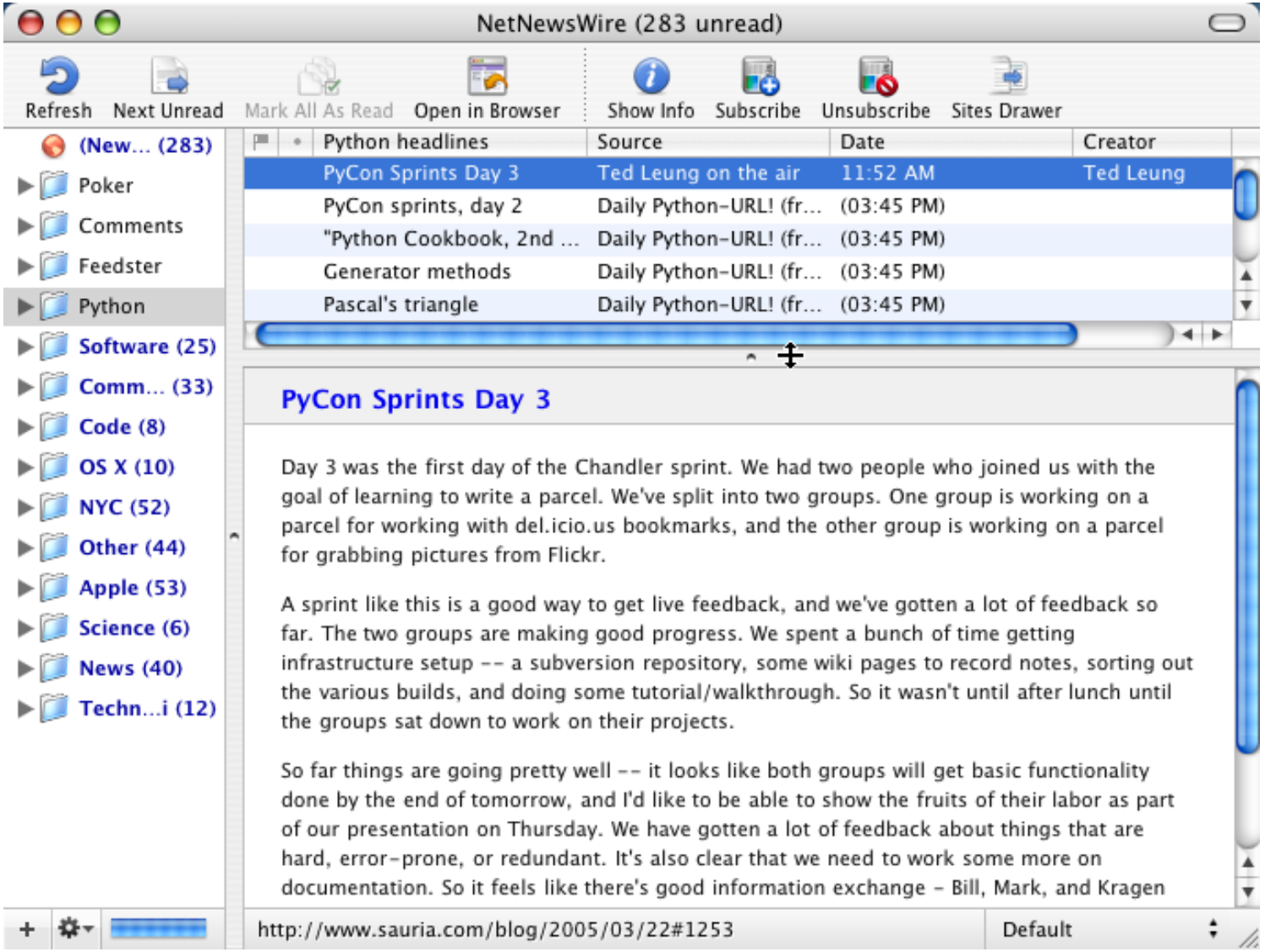
Why Bother?

- You paid for that Mac
- The tools kick ass
- Apple (often) writes good code
- The tools kick ass
- Objective-C and Python are friends

SubEthaEdit is Cocoa...



So is NetNewsWire...



Interface Builder

- Design your interface
- ... using a well designed interface
- Don't write so much code
- Plug objects together
- Manages an *object graph*
- ... think pickle

Objective-C

- True superset of C
- Everything is not an object
- Looks kinda like Smalltalk

Classes

- Flat Namespace
- Single Inheritance
- ... with Categories and Protocols
- Classes are objects
- Instance Variables

Objective-C Interface

```
@interface MyClass : NSObject
{
    int myInt;
}
+(id)myClassWithInt:(int)anInt;
-(int)myInt;
@end
```

Objective-C Implementation

```
@implementation MyClass

+(id)myClassWithInt:(int)anInt;
{
    self = [[self alloc] init];
    intInstanceVariable = anInt;
    return self;
}

-(int)myInt
{
    return myInt;
}

@end
```

Objects

- Separate alloc/init
- Everything is an accessor
- ... except when using Key-Value Coding
- Reference counted
- ... but we take care of that
- ... except where Apple doesn't

Messages

- Target
- Selector
- Arguments
- nil receives anything

Exceptions

- Exceptions are exceptional
- Expect bad code to just crash
- ... even from Python

Crossing the Bridge

- All NSString are *UNICODE*
- str is not safely bridged to anything!
- int, long, float work magically
- ... for value and object types
- None is just like nil
- ... except you can't send messages to it!

Bridged Messaging

Objective-C:

```
[aMutableArray addObject:@"someObject"]
```

- Separate the selector:from the:arguments
- Smash_the_colons_
- Ditch.the_brackets_(and, add, arguments)

Python:

```
aMutableArray.addObject_(u'someObject')
```

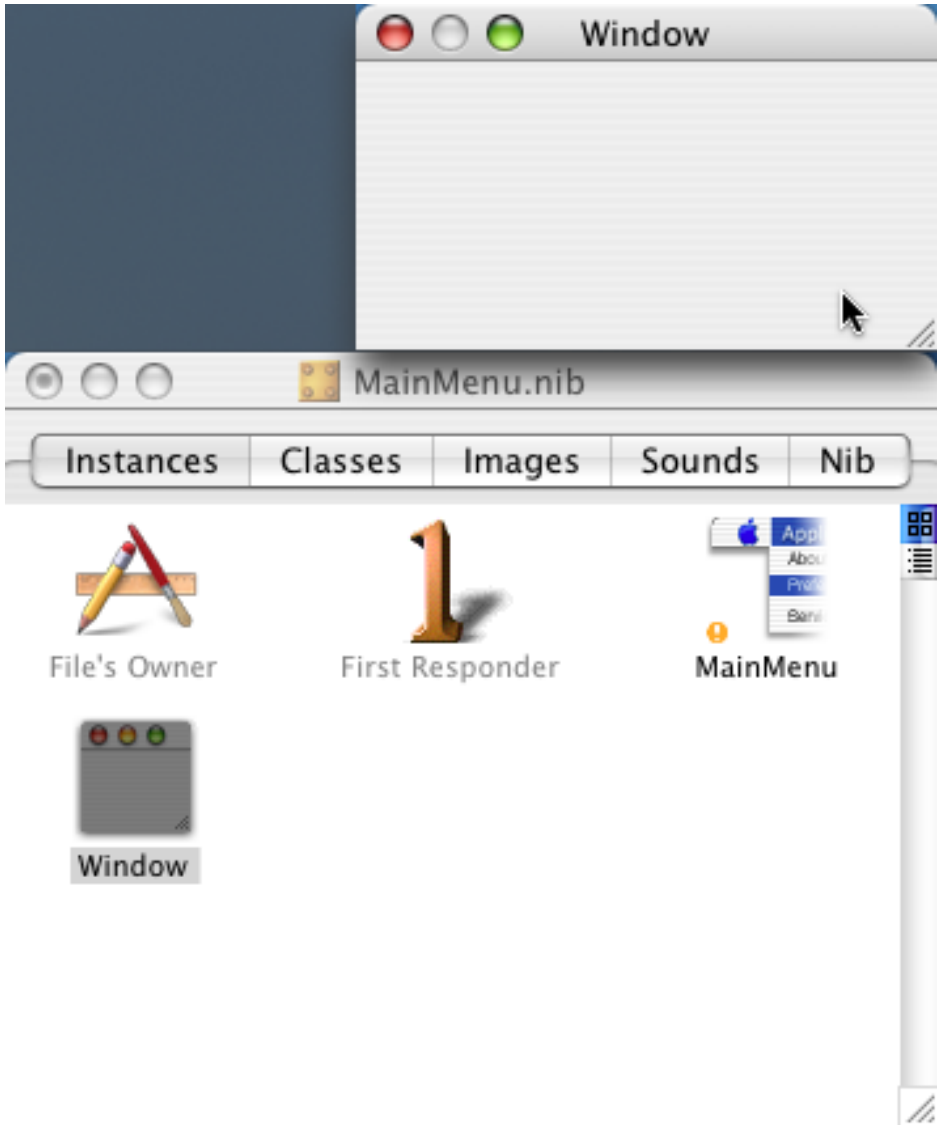
Key-Value Coding

- Kinda like getattr protocol
- Accessor
- ivar
- valueForUndefinedKey: (like __getattr__)

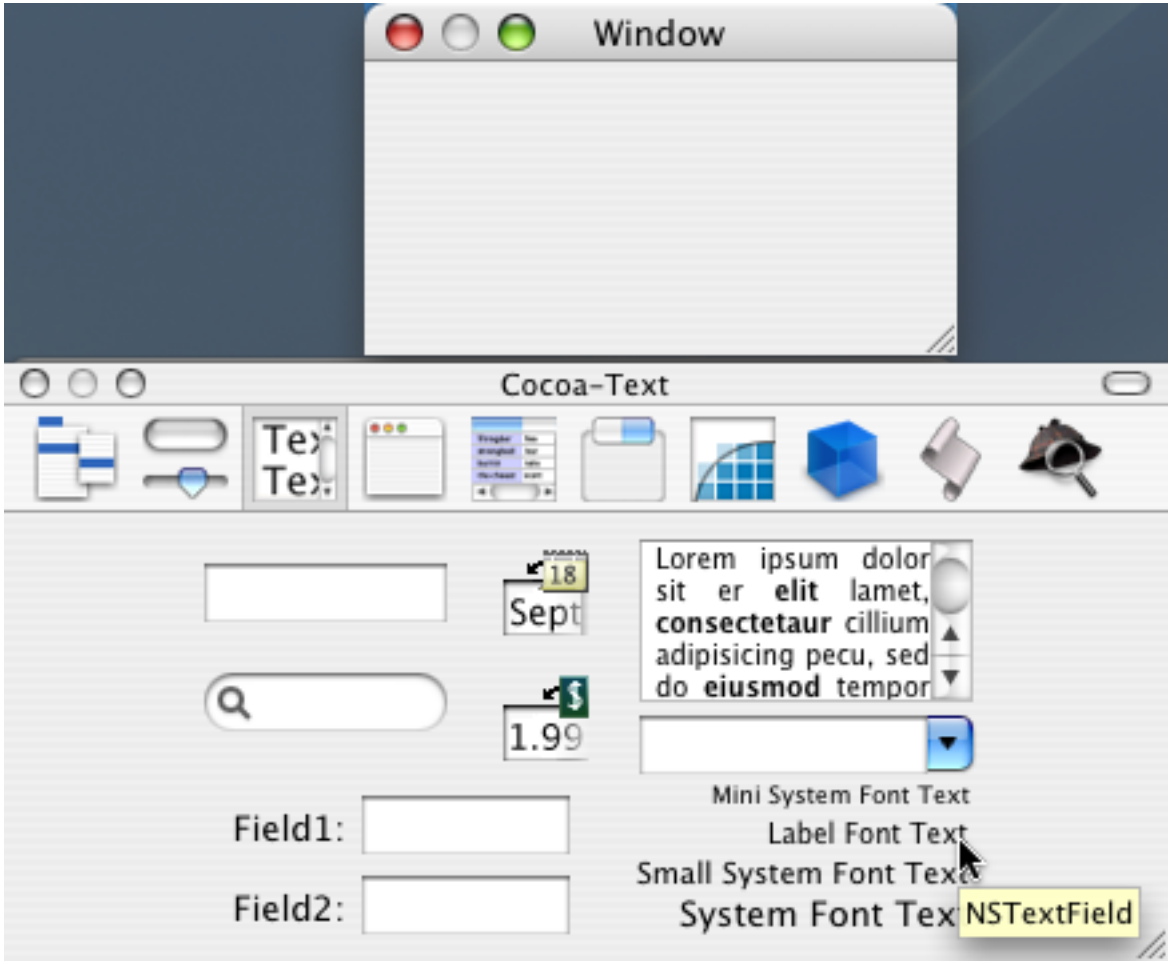
Making Money

- Currency Converter
- Using Cocoa Bindings
- Almost entirely in Interface Builder

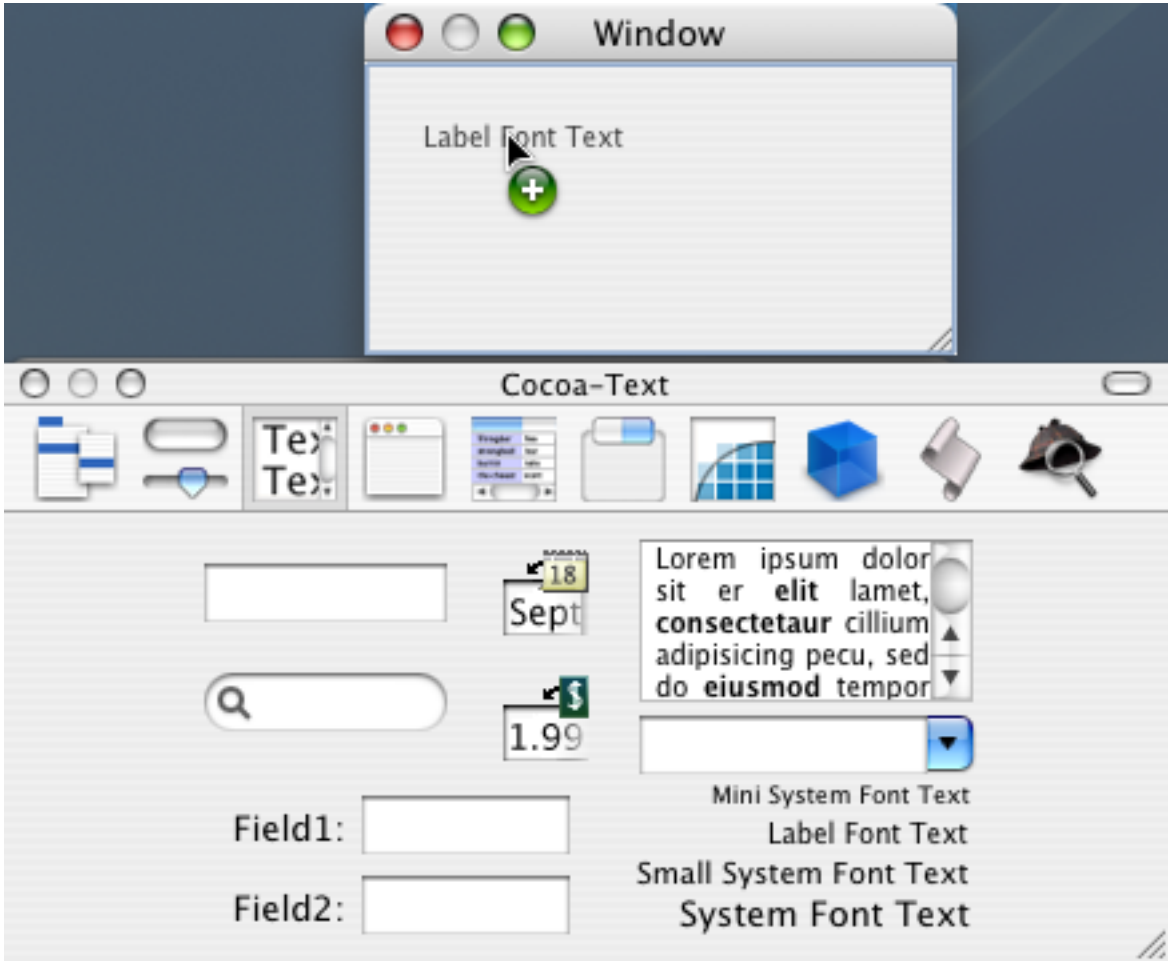
New Application in IB



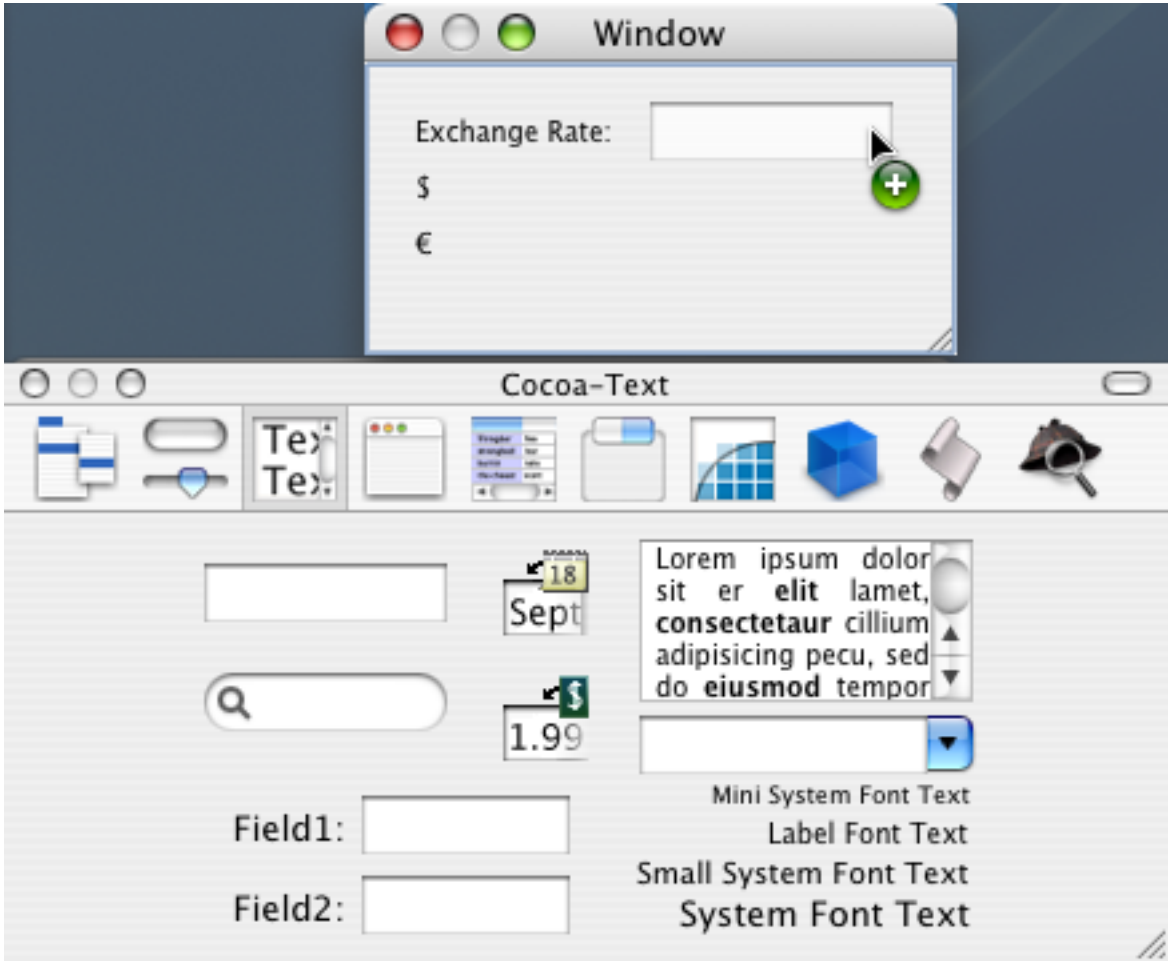
Create an NSTextField



Drag to the NSWindow

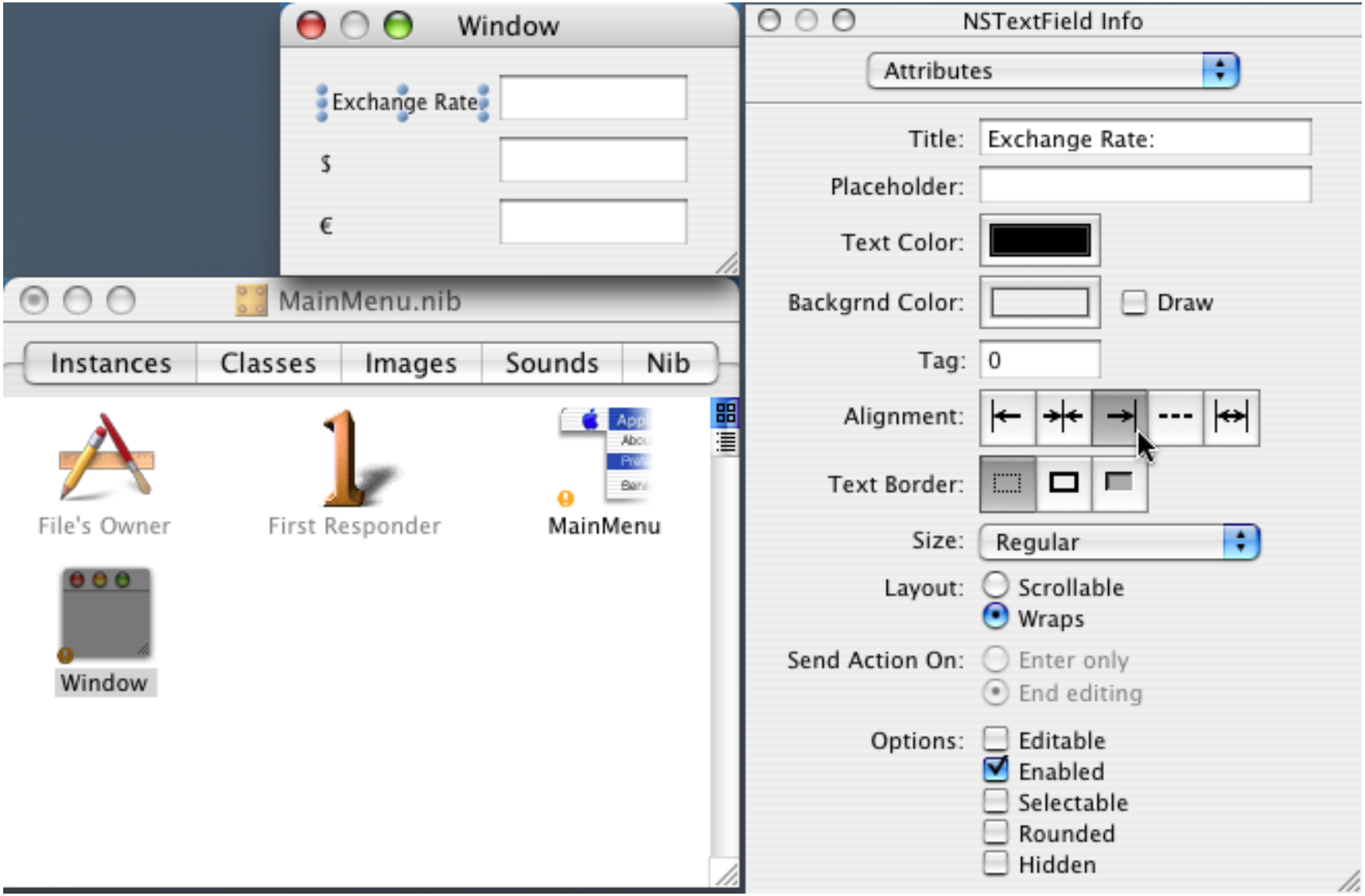


Create the input NSTextFields

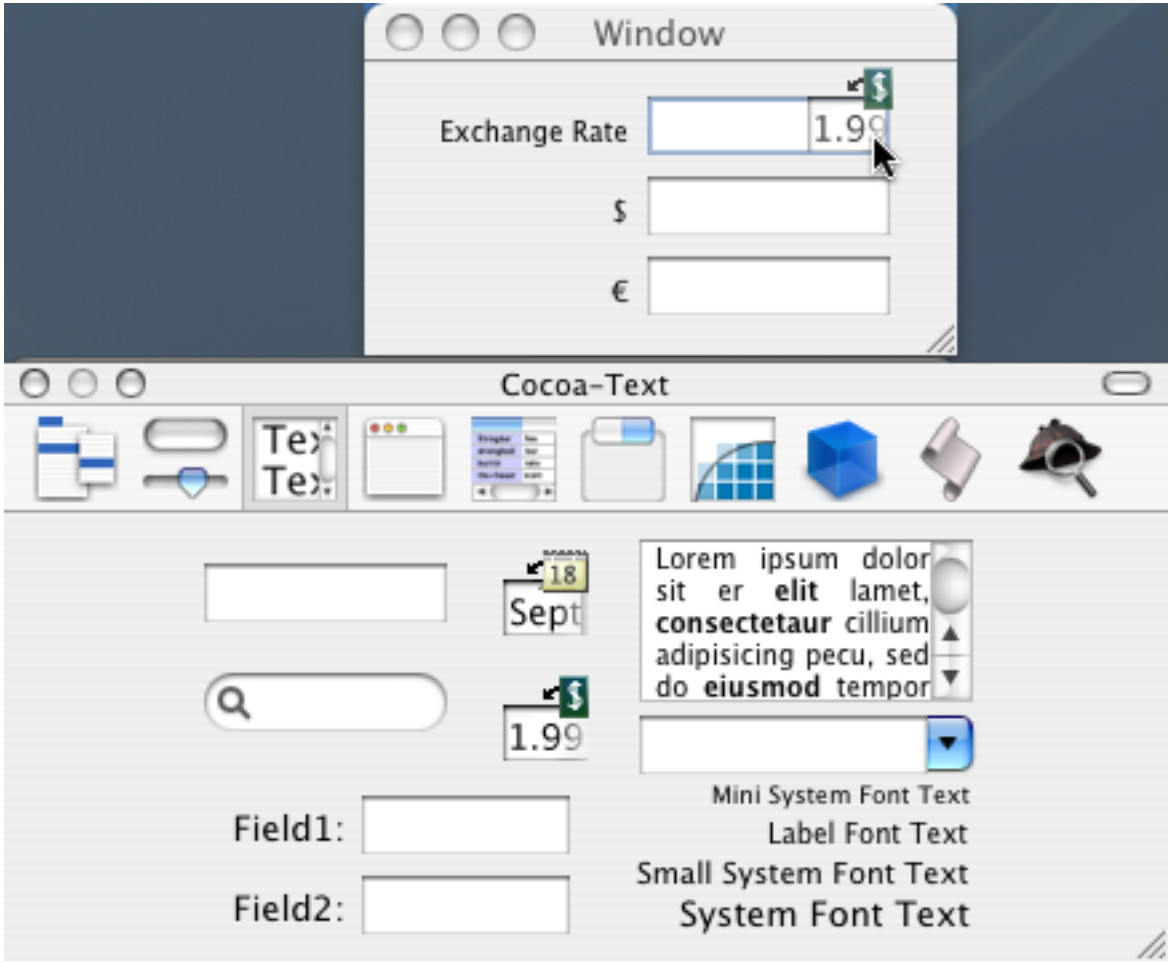


Almost finished UI Layout

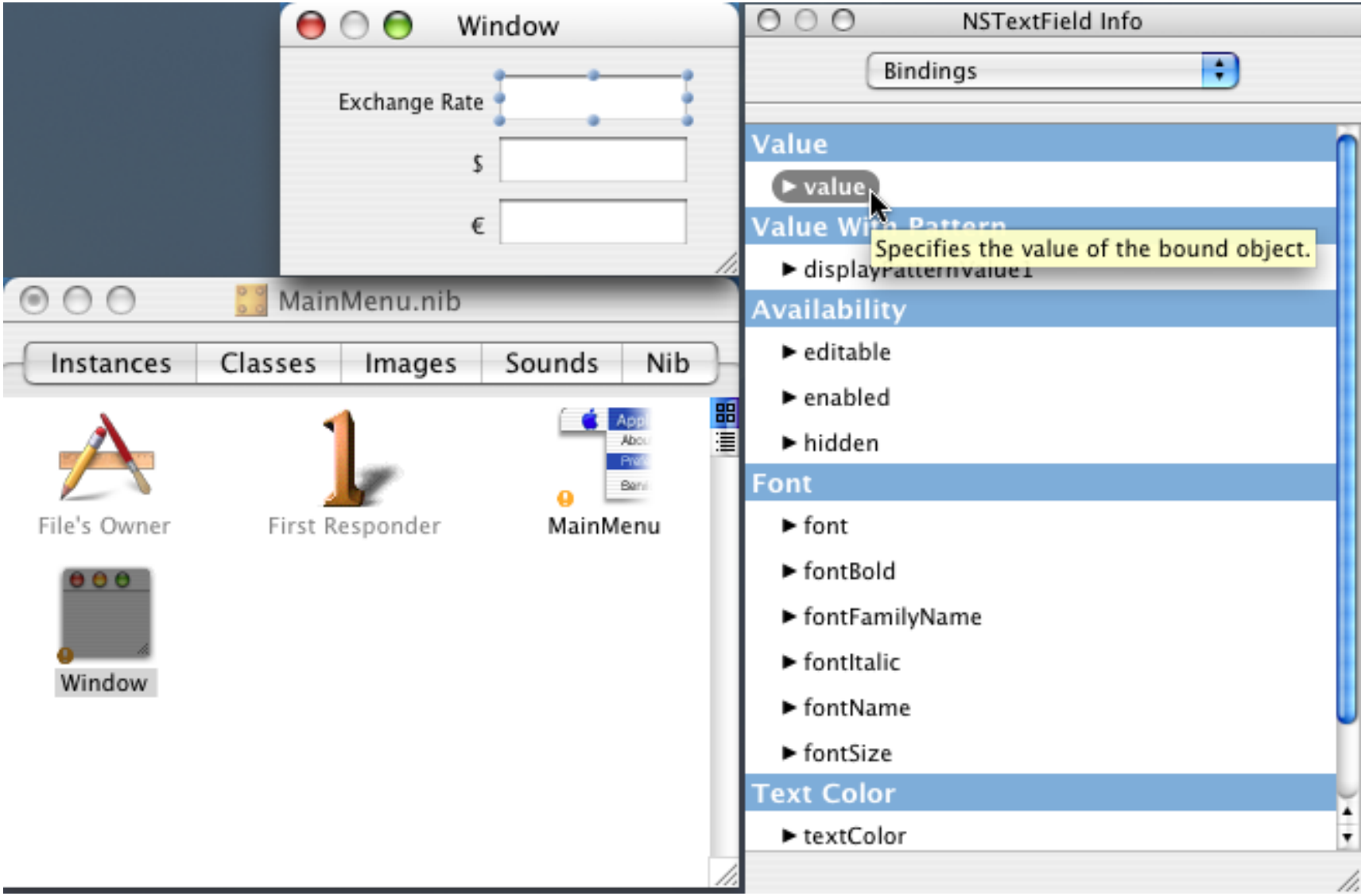
Align the labels



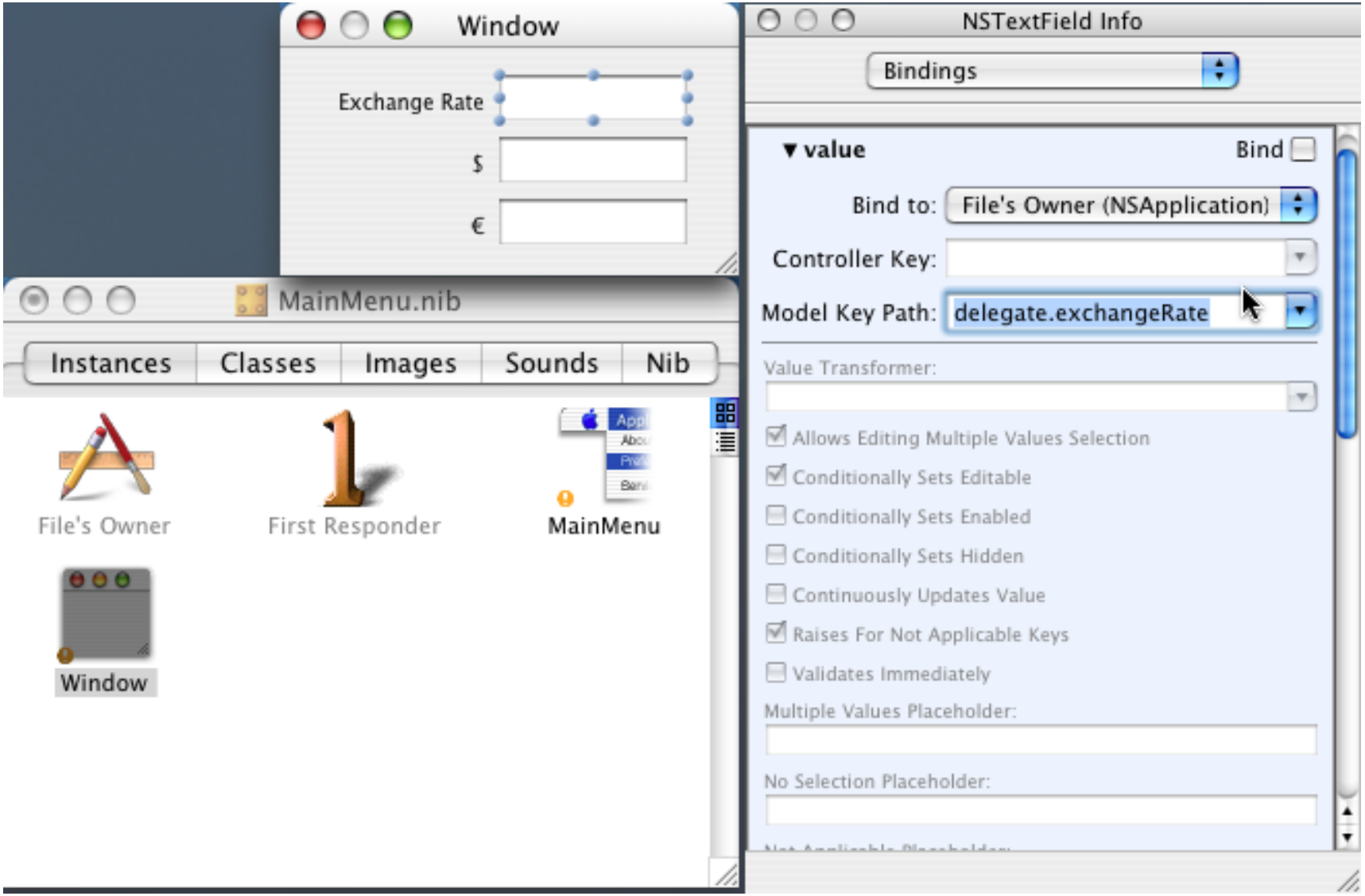
Use currency NSNumberFormatters



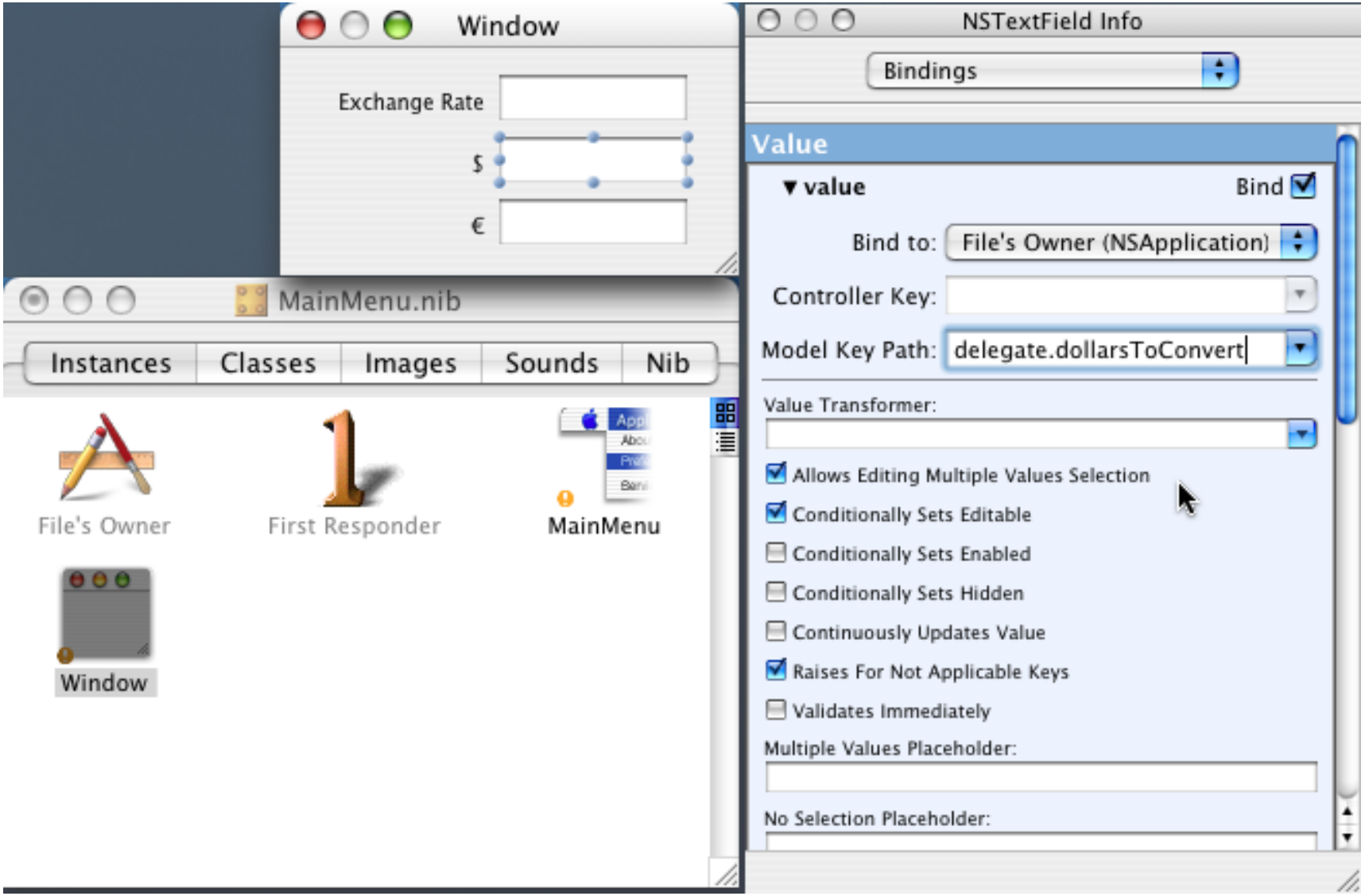
Set up the Bindings



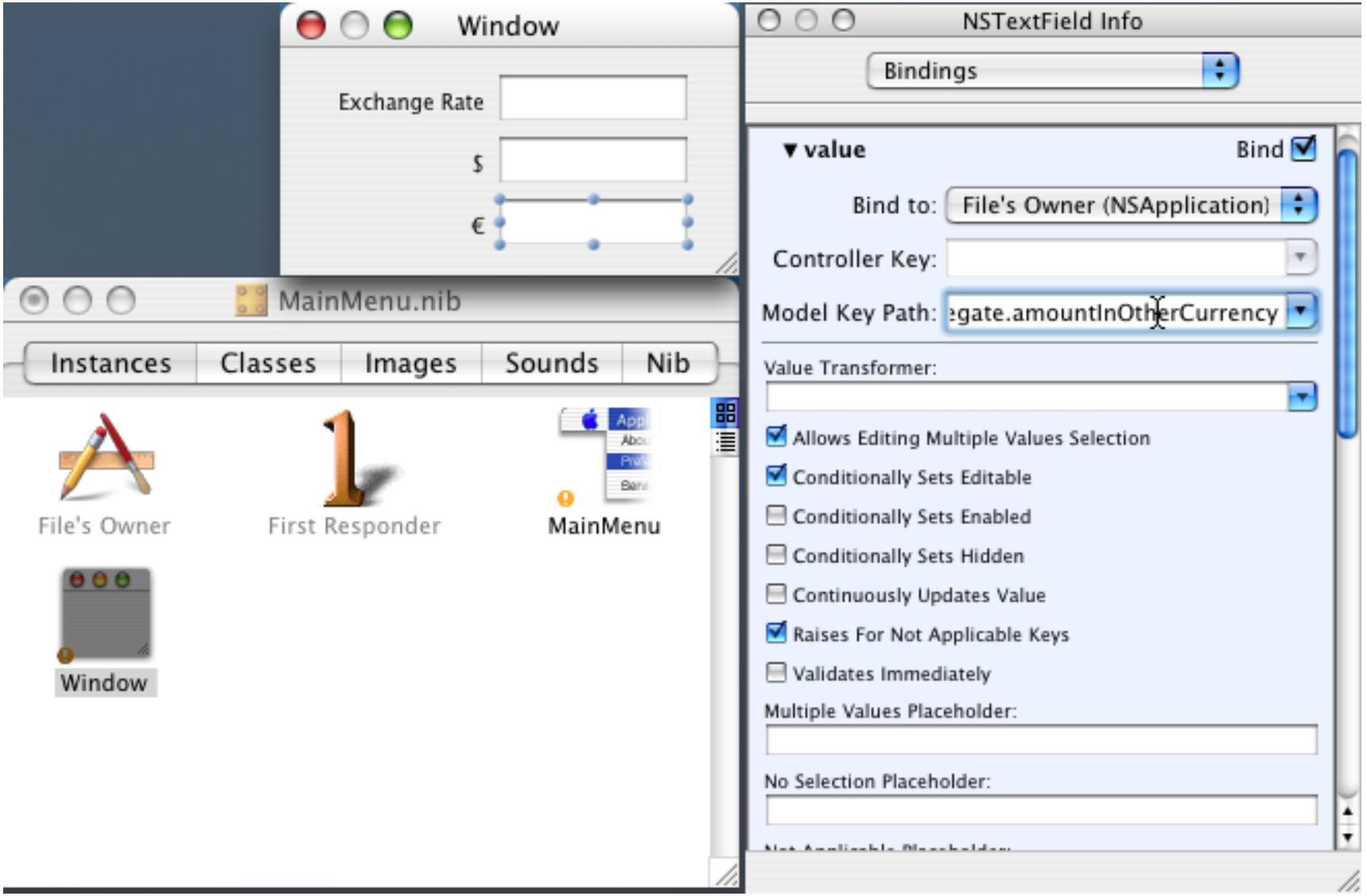
To point to your delegate



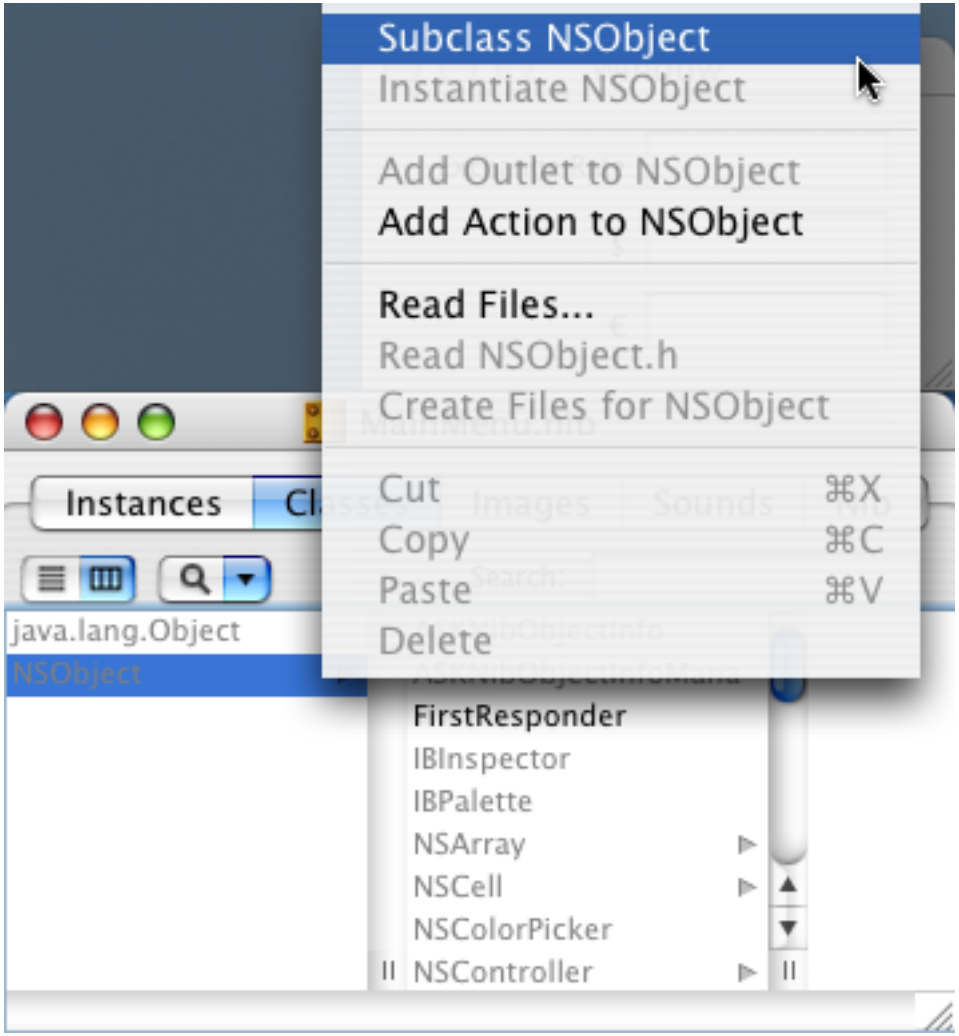
Dollars binding...



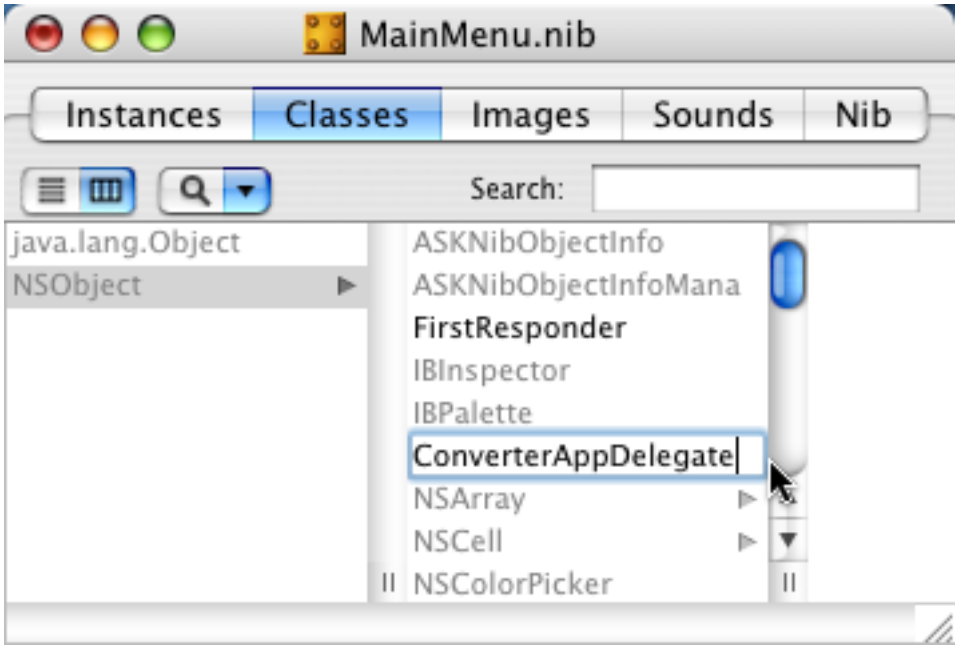
Other Currency Binding...



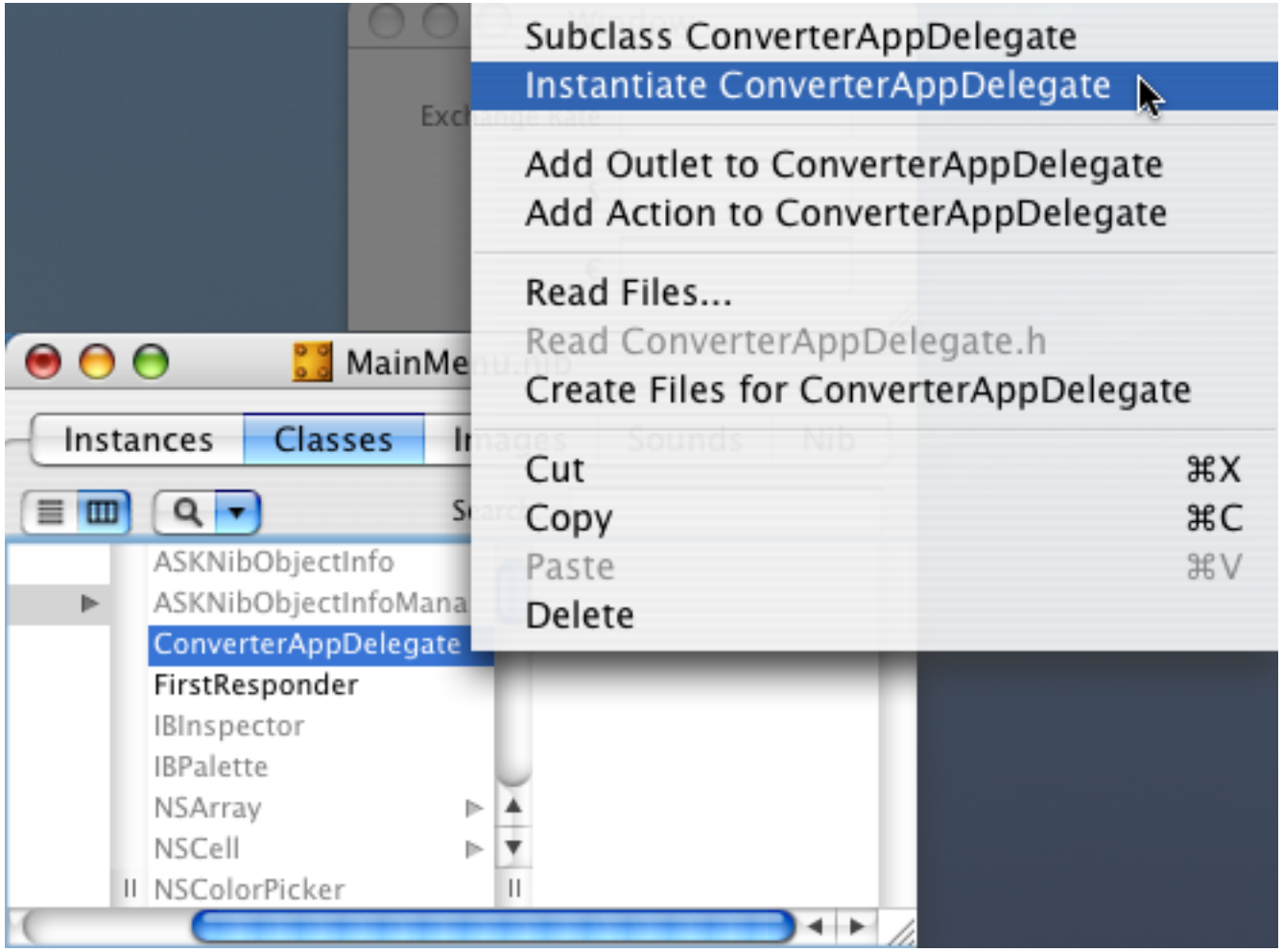
Subclass NSObject



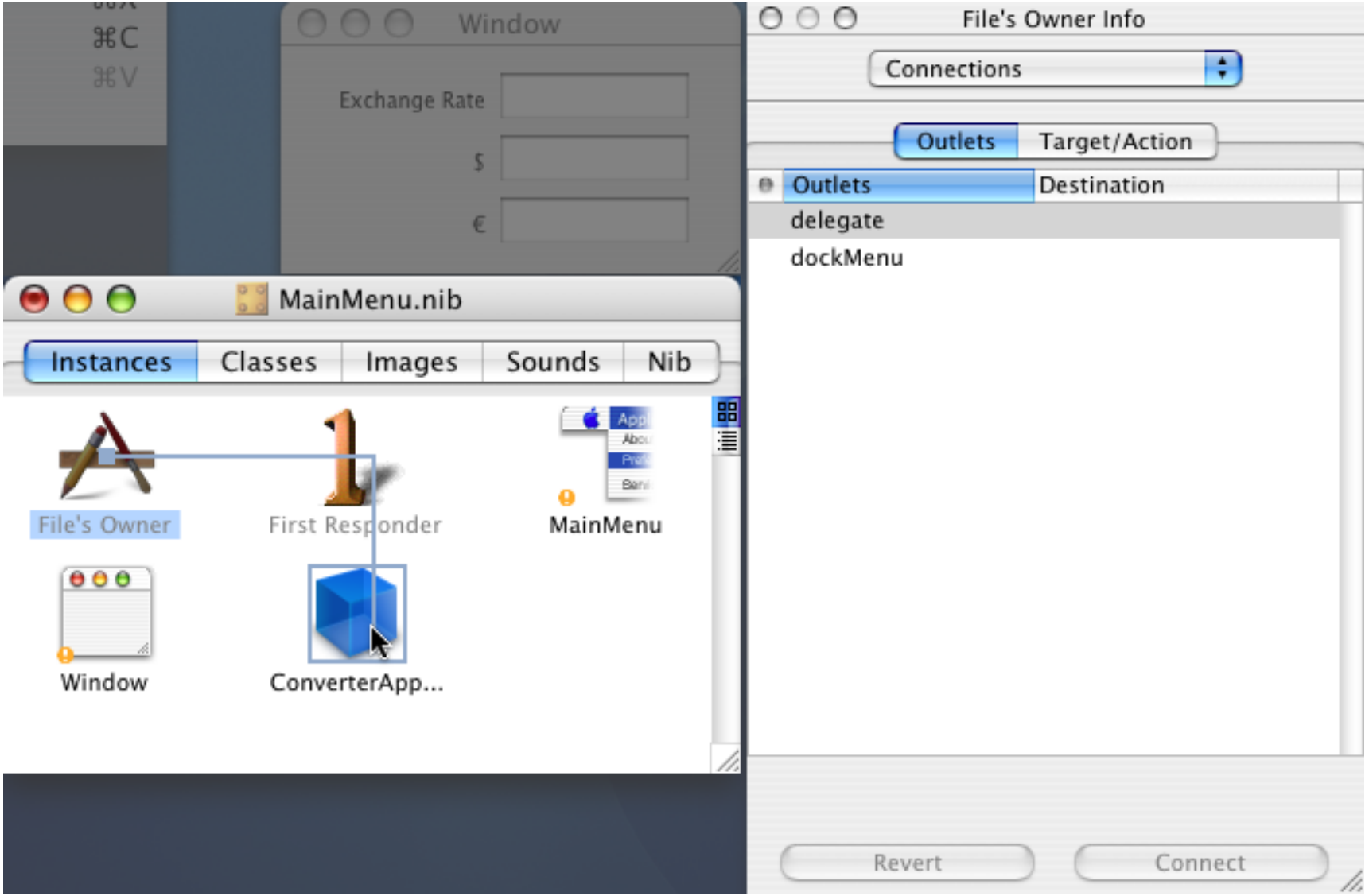
To create your delegate class



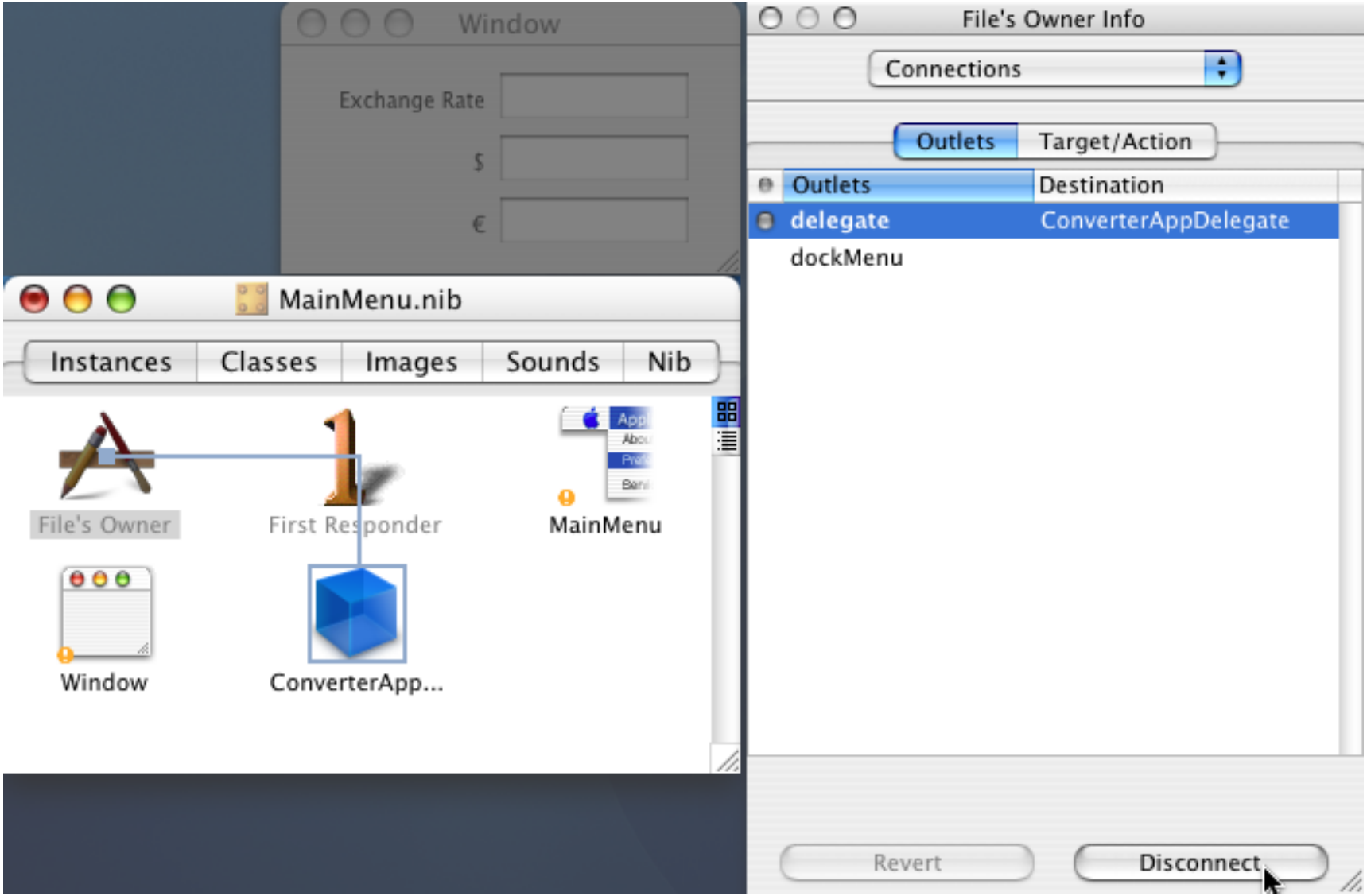
Instantiate it in your nib



Create a connection



To the NSApplication



ConverterAppDelegate.py Class

```
from Foundation import *
from AppKit import *
import objc

class ConverterAppDelegate(NSObject):
    def init(self):
        self = super(ConverterAppDelegate, self).init()
        self.exchangeRate = 3
        self.dollarsToConvert = 4
        return self

    def amountInOtherCurrency(self):
        return self.dollarsToConvert * self.exchangeRate

    def setAmountInOtherCurrency_(self, amt):
        self.dollarsToConvert = amt / self.exchangeRate

# shamelessly preventing line wrapping
cls = ConverterAppDelegate
cls.setKeys_triggerChangeNotificationsForDependentKey_(
    [u'dollarsToConvert', u'exchangeRate'],
    u'amountInOtherCurrency',
)
```

Converter.py script

```
from PyObjCTools import AppHelper
import ConverterAppDelegate
if __name__ == '__main__':
    AppHelper.runEventLoop()
```

Converter setup.py script

```
from distutils.core import setup
import py2app
setup(
    app = ['Converter.py'],
    data_files = ['MainMenu.nib'],
)
```

Build and Run

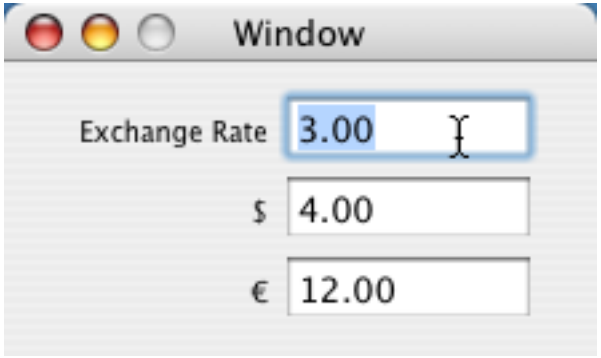
Build:

```
% python setup.py py2app --alias
```

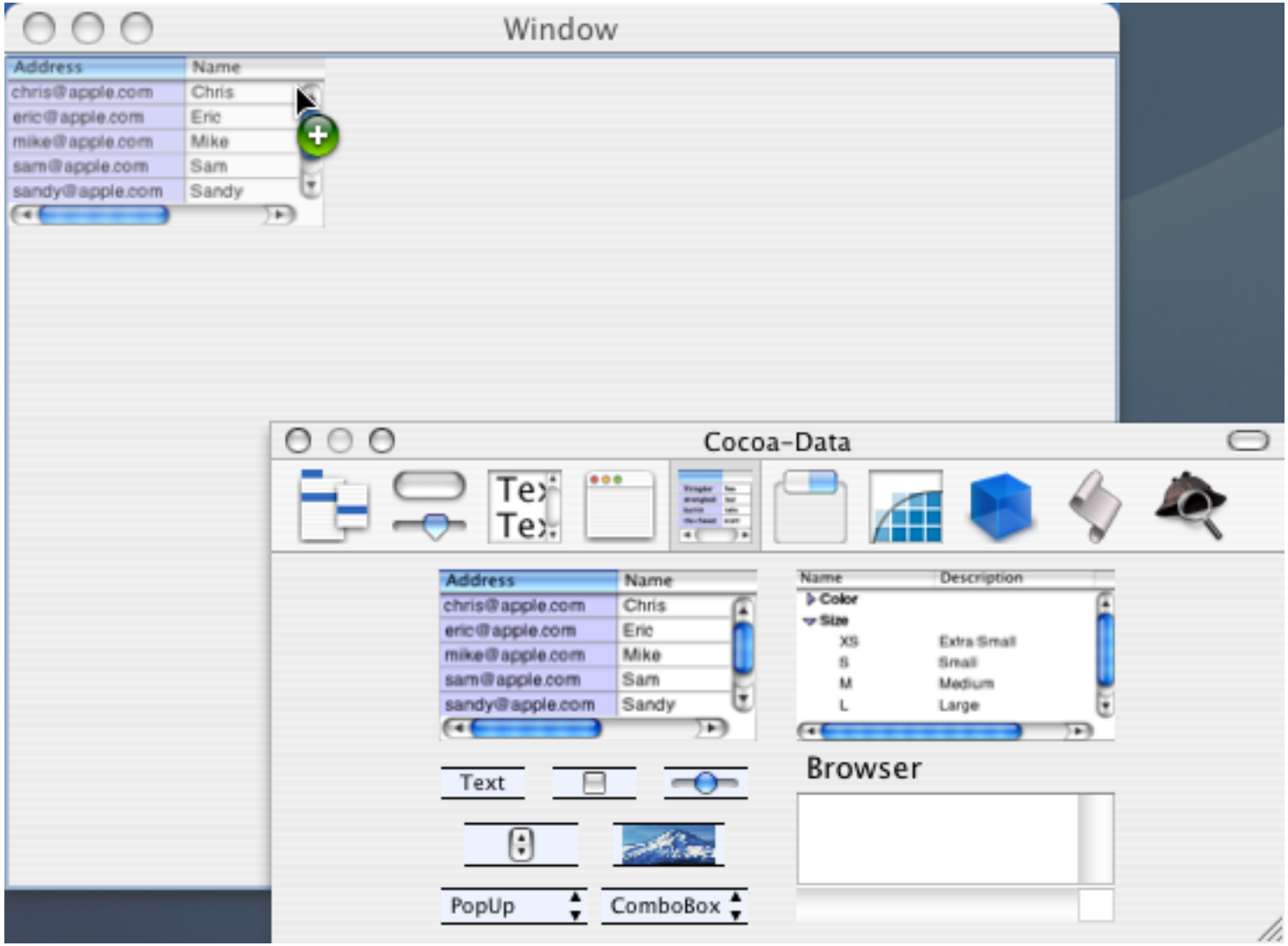
Run:

```
% open dist/Converter.app
```

Done:

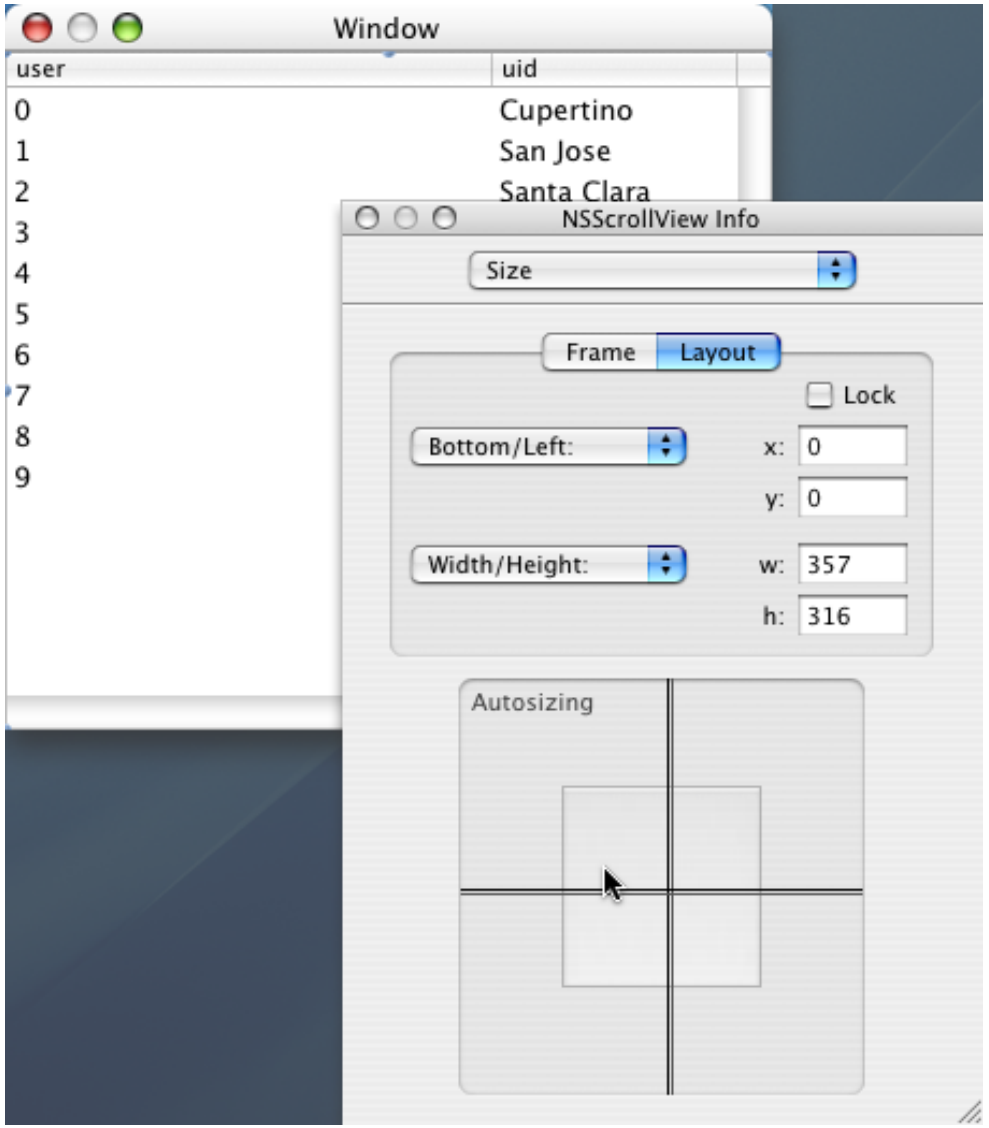


New NSTableView

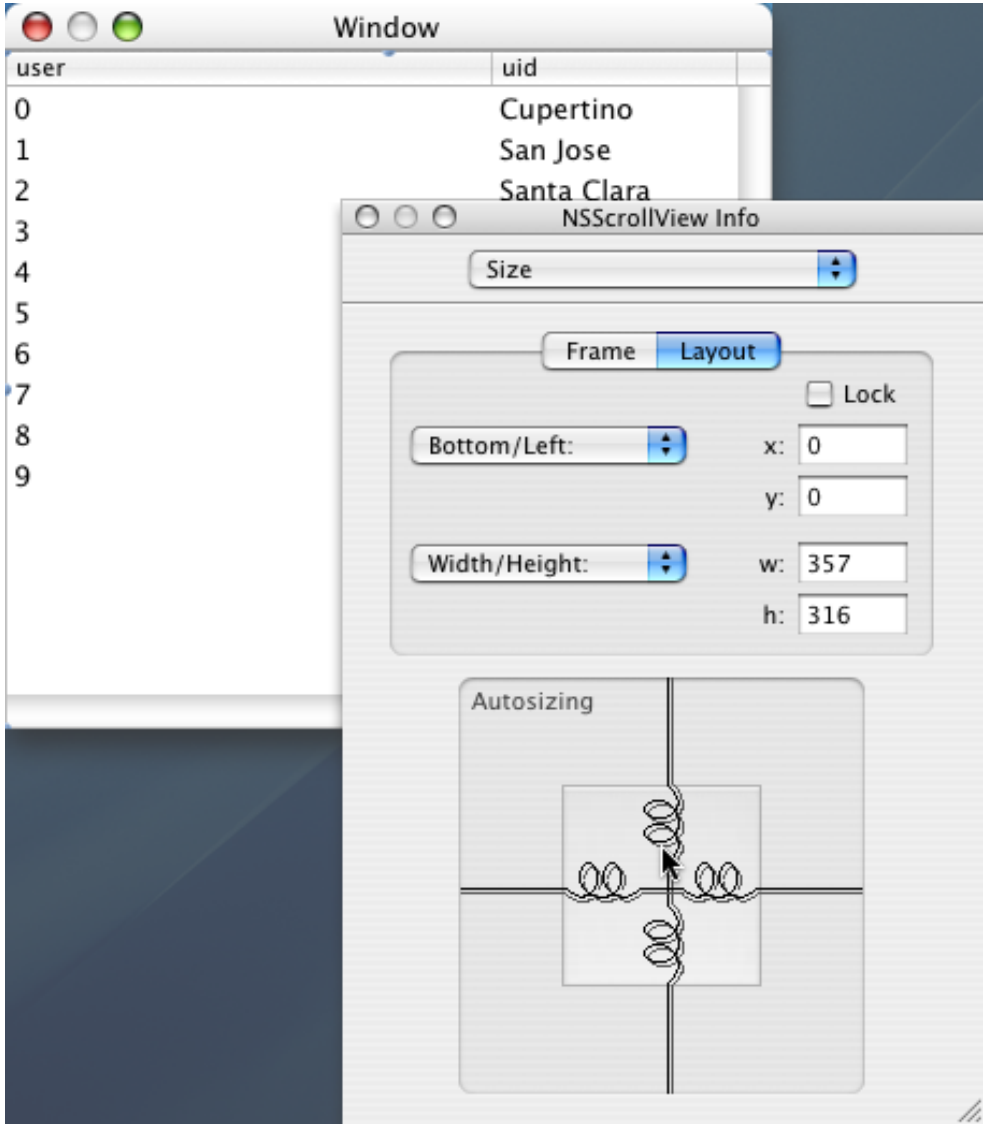


Name the columns

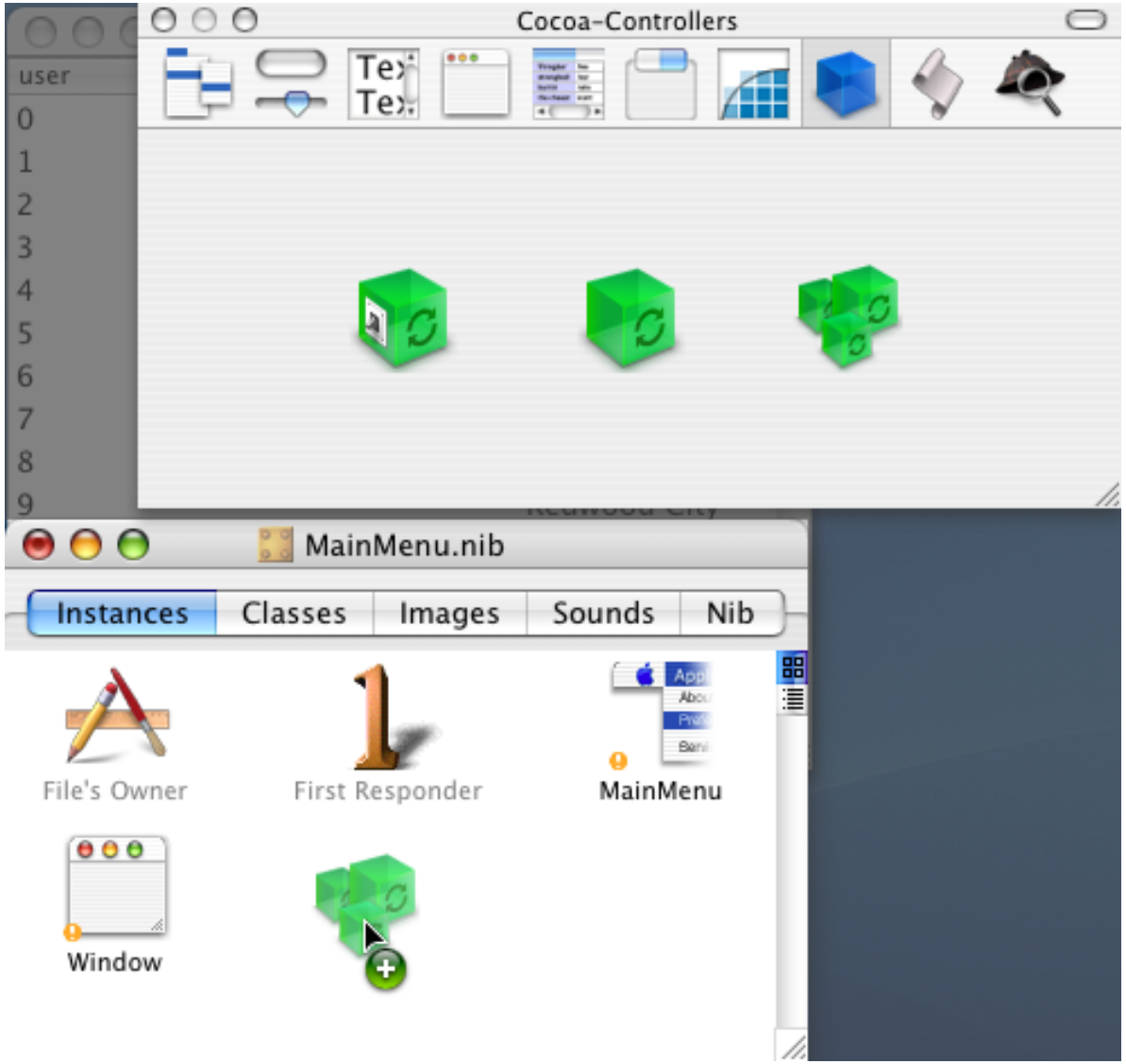
Change the resize behavior



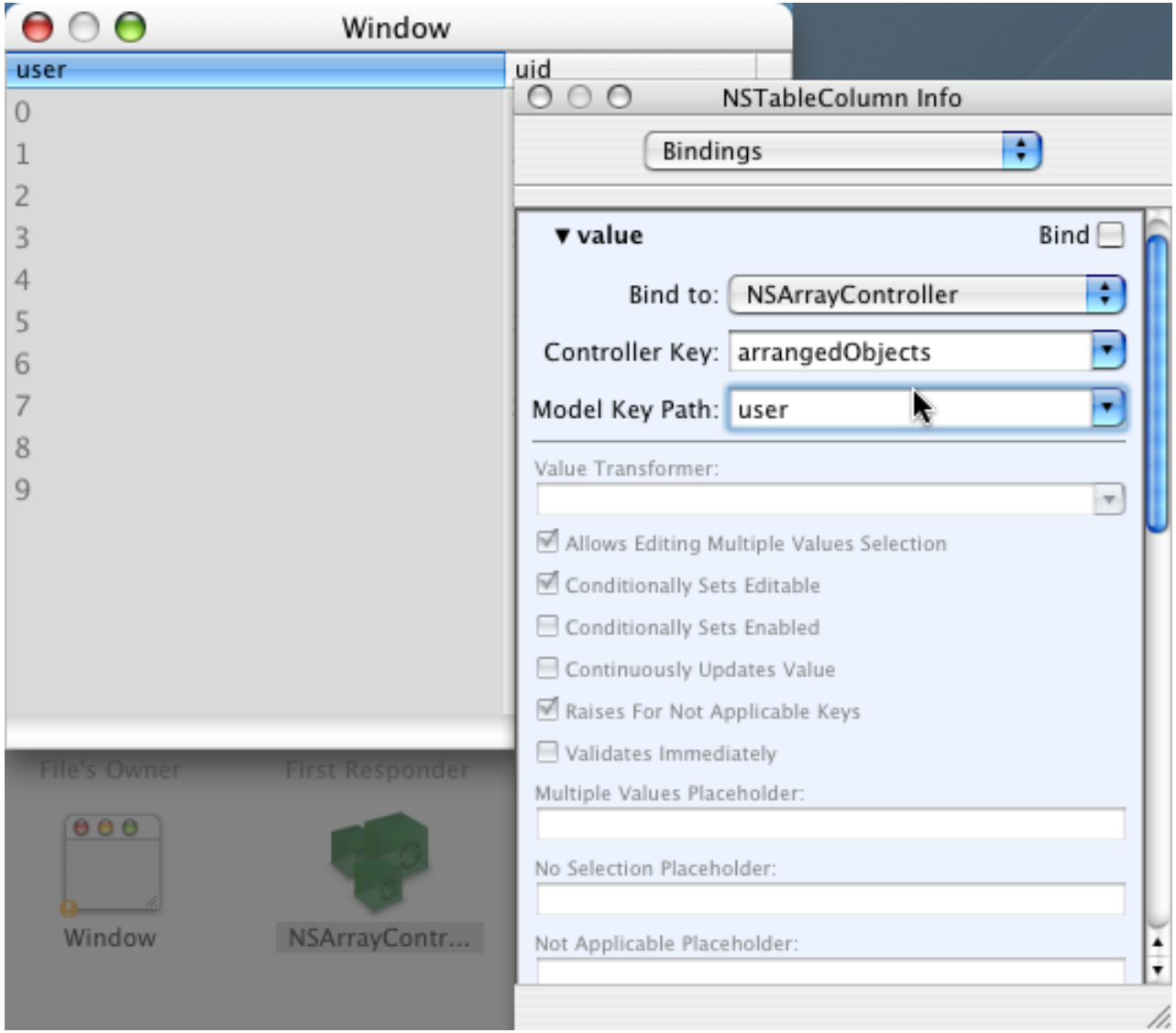
To expand with the NSWindow



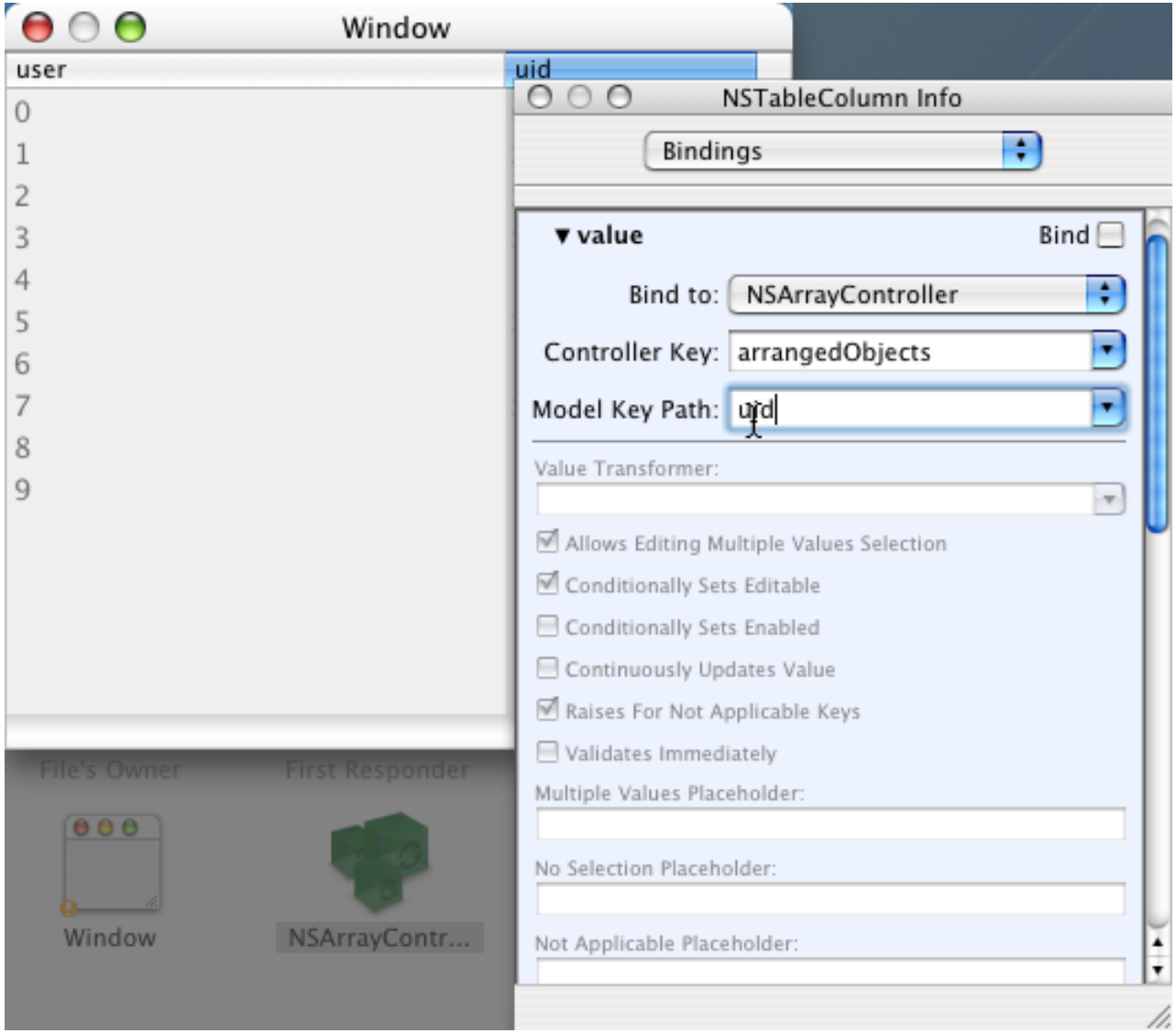
Create an NSArrayController



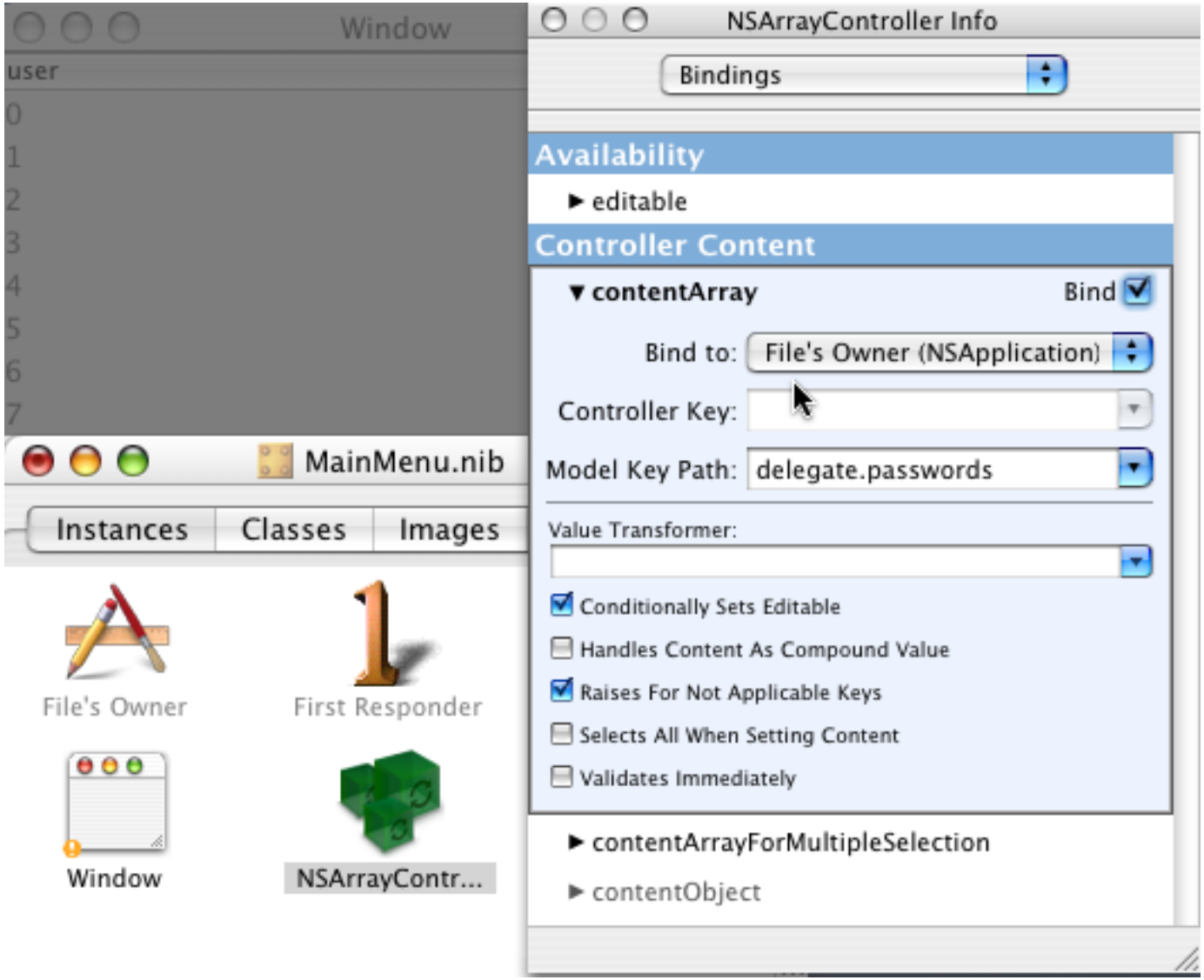
Bind the user column



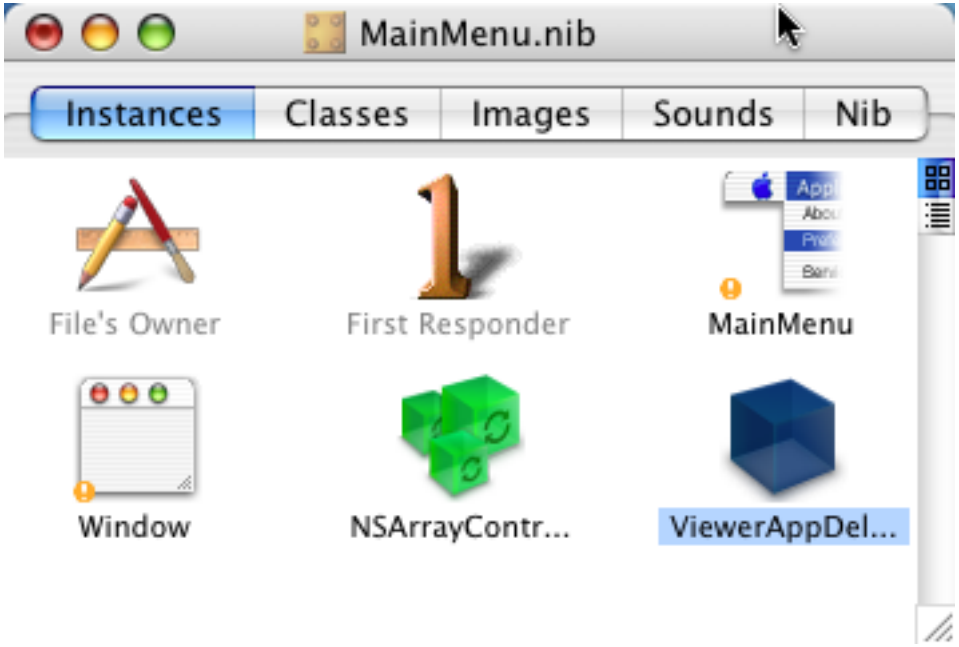
Bind the uid column



Bind the NSArrayController



Create the ViewerAppDelegate



Like the previous application:

- Subclass NSObject
- Instantiate the subclass
- Connect it to the NSApplication's delegate outlet

Viewer.py

```
from PyObjCTools import AppHelper
from Foundation import *
from AppKit import *
import os

# another shameless anti-line-wrapping hack
FIELDS = """
user password uid gid class change
expire gecos home_dir shell
""".split()

class ViewerAppDelegate(NSObject):
    def init(self):
        self = super(ViewerAppDelegate, self).init()
        self.passwords = [
            dict(zip(FIELDS, line.rstrip().split(':')))
            for line in os.popen('/usr/bin/nidump passwd .')
            if line and not line.startswith('#')
        ]
        return self

if __name__ == '__main__':
    AppHelper.runEventLoop()
```

Build and Run Viewer

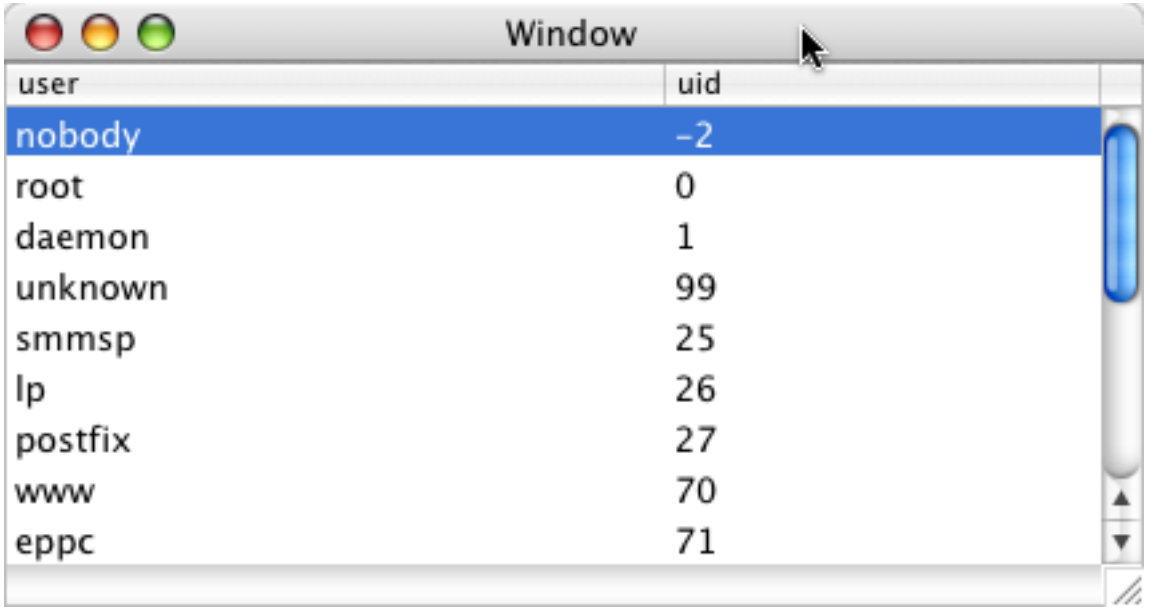
Build (redistributable!):

```
% py2applet Viewer.py MainMenu.nib
```

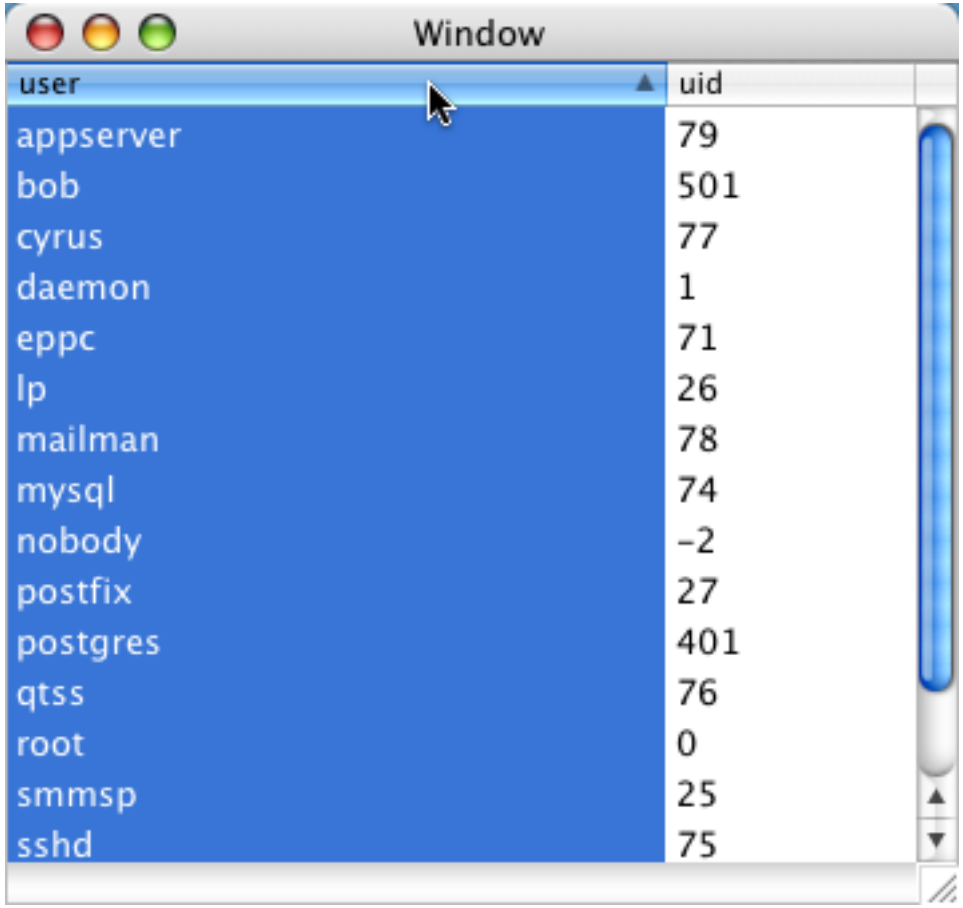
Run:

```
% open Viewer.app
```

Done:



Bindings give you sorting for free!



Help!

Documentation:

/Developer/Python/PyObjC/Documentation

Examples:

/Developer/Python/PyObjC/Examples

Wiki:

http://pythonmac.org/wiki

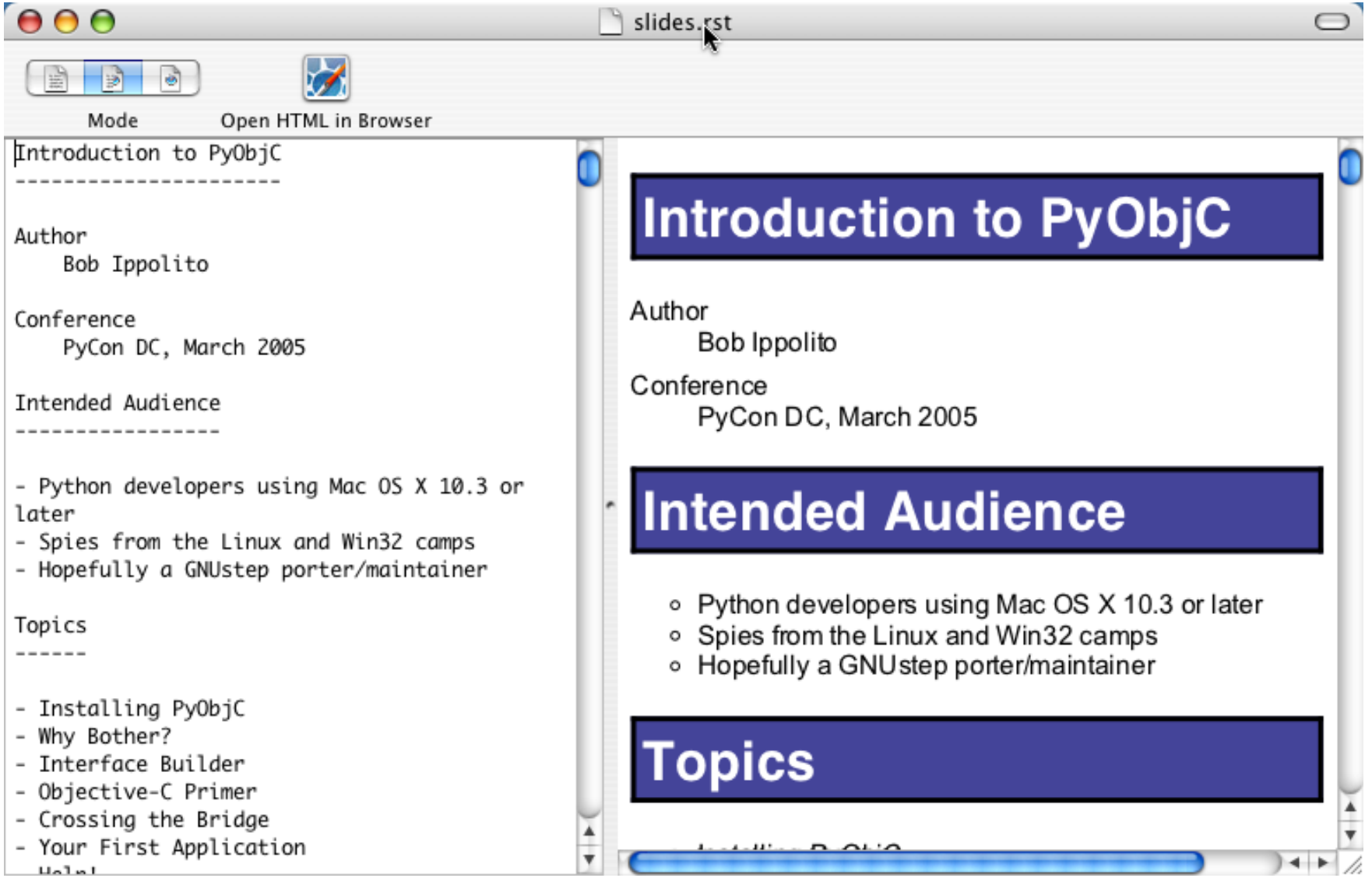
IRC:

#pythonmac (on freenode)

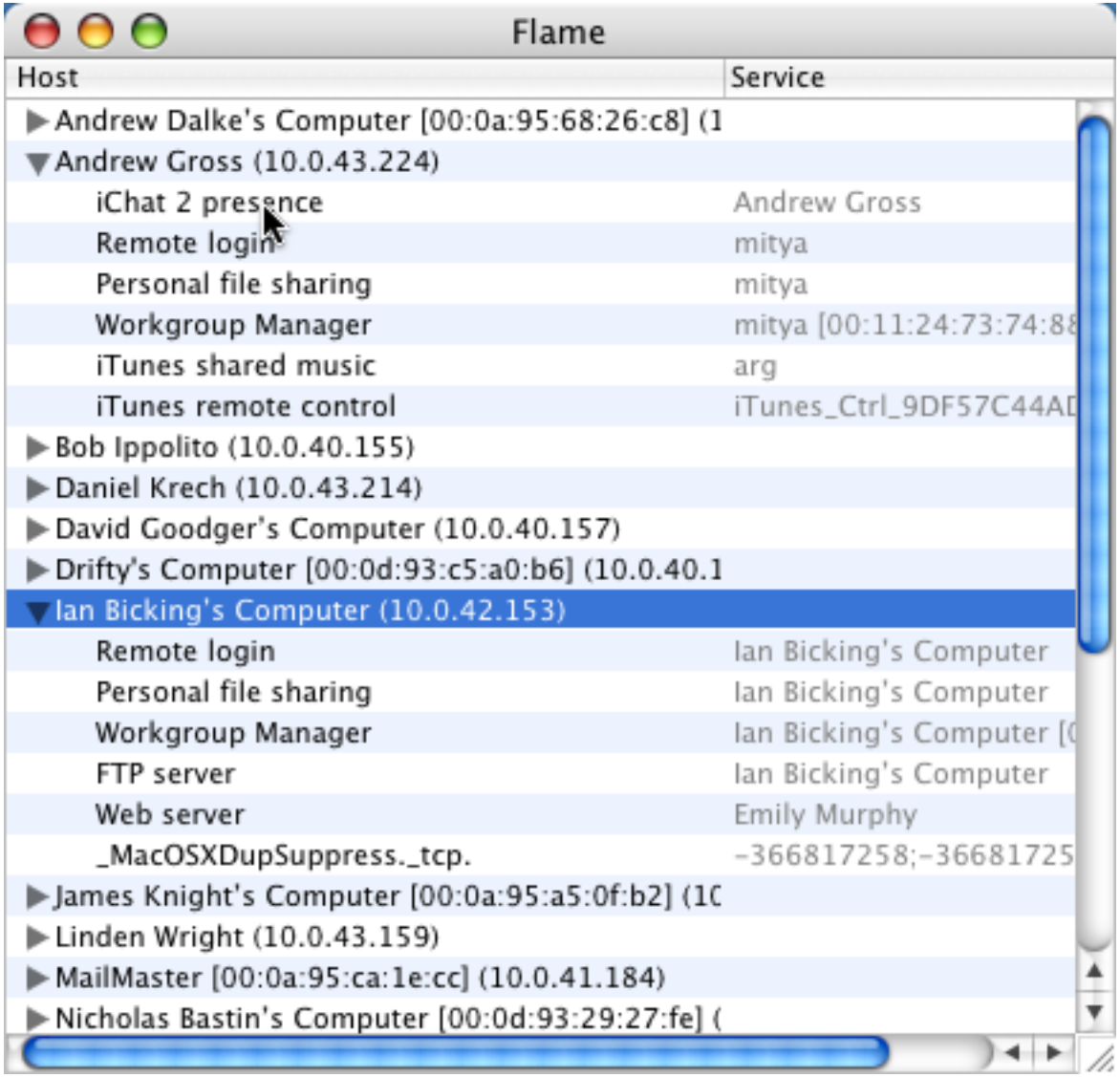
Mailing List:

- pythonmac-sig@python.org
- pyobjc-dev@lists.sourceforge.net

ReSTedit


slides.rst
Mode
Open HTML in Browser
Introduction to PyObjC
Author
Bob Ippolito
Conference
PyCon DC, March 2005
Intended Audience
- Python developers using Mac OS X 10.3 or later
- Spies from the Linux and Win32 camps
- Hopefully a GNUstep porter/maintainer
Topics
- Installing PyObjC
- Why Bother?
- Interface Builder
- Objective-C Primer
- Crossing the Bridge
- Your First Application
Introduction to PyObjC
Author
Bob Ippolito
Conference
PyCon DC, March 2005
Intended Audience
Python developers using Mac OS X 10.3 or later
Spies from the Linux and Win32 camps
Hopefully a GNUstep porter/maintainer
Topics

Flame


Flame
Host
Service
Andrew Dalke's Computer [00:0a:95:68:26:c8] (1
Andrew Gross (10.0.43.224)
iChat 2 presence
Andrew Gross
Remote login
mitya
Personal file sharing
mitya
Workgroup Manager
mitya [00:11:24:73:74:88
iTunes shared music
arg
iTunes remote control
iTunes_Ctrl_9DF57C44AD
Bob Ippolito (10.0.40.155)
Daniel Krech (10.0.43.214)
David Goodger's Computer (10.0.40.157)
Drifty's Computer [00:0d:93:c5:a0:b6] (10.0.40.1
Ian Bicking's Computer (10.0.42.153)
Remote login
Ian Bicking's Computer
Personal file sharing
Ian Bicking's Computer
Workgroup Manager
Ian Bicking's Computer [(
FTP server
Ian Bicking's Computer
Web server
Emily Murphy
_MacOSXDupSuppress._tcp.
-366817258;-36681725
James Knight's Computer [00:0a:95:a5:0f:b2] (1C
Linden Wright (10.0.43.159)
MailMaster [00:0a:95:ca:1e:cc] (10.0.41.184)
Nicholas Bastin's Computer [00:0d:93:29:27:fe] (

NodeBox

Copyrights.py

```
size(300, 300)

white = color(1,1,1,0.9)
red = color(1,0,0,0.9)
black = color(0,0,0,0.9)

for i in range(100):
    fill(choice((white,red,black)))
    font("Arial Bold")
    fontsize(random(300))
    text(u"™", random(250),random(200))
    text(u"®", random(250),random(200))
    text(u"©", random(250),random(200))
```

Questions?

Go ahead, ask.