



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...

The screenshot displays the SubEthaEdit application window, titled "Examples/AppKit/CircleView/CircleView.m". The main area is a code editor showing Objective-C code for a circle animation. The code includes comments and method implementations for starting, stopping, toggling, and performing the animation. A right-hand panel titled "Access Control:" is open, showing a list of users and their permissions. The "Locked" status is indicated at the top of the panel. Below the status, there is a list of users: Martin Pittenauer (245:0), Martin Ott (225:0 (44)), and Dominik Wagner (86:16 (1293)). A context menu is open over the list, showing options: "Follow", "Open chat" (highlighted), "Send email", "Promote to read/write", "Set to read only", and "Kick". The bottom of the window shows the status bar with "Objective-C", "Spaces (4)", "(LF)", "Western (Mac OS Roman)", and "70w".

```
245:0 M -performAnimation:
217 // place (for example, while a slider is being dragged).
218 [[NSRunLoop currentRunLoop] addTimer:timer forMode:
NSModalPanelRunLoopMode];
219 [[NSRunLoop currentRunLoop] addTimer:timer forMode:
NSEventTrackingRunLoopMode];
220
221 lastTime = [NSDate timeIntervalSinceReferenceDate];
222 }
223
224 - (IBAction)stopAnimation:(id)sender {
225     [timer invalidate];
226     [timer release];
227     timer = nil;
228 }
229
230 - (IBAction)toggleAnimation:(id)sender {
231     if (timer != nil) {
232         [self stopAnimation:sender];
233     } else {
234         [self startAnimation:sender];
235     }
236 }
237
238 - (void)performAnimation:(NSTimer *)aTimer {
239     // We determine how much time has elapsed since the last animation,
240     // and we advance the angle accordingly.
241     NSTimeInterval thisTime = [NSDate timeIntervalSinceReferenceDate];
242     [self setStartingAngle:startingAngle + angularVelocity * (thisTime -
lastTime)];
243     lastTime = thisTime;
244 }
245
246 @end
247
```

Access Control:

Locked

Read/Write

Martin Pittenauer 245:0

Martin Ott 225:0 (44)

Dominik Wagner 86:16 (1293)

Follow

Open chat

Send email

Promote to read/write

Set to read only

Kick

Objective-C Spaces (4) (LF) Western (Mac OS Roman) 70w



So is NetNewsWire...

NetNewsWire (283 unread)

Refresh Next Unread Mark All As Read Open in Browser Show Info Subscribe Unsubscribe Sites Drawer

	Python headlines	Source	Date	Creator
(New... (283)	PyCon Sprints Day 3	Ted Leung on the air	11:52 AM	Ted Leung
▶ Poker	PyCon sprints, day 2	Daily Python-URL! (fr...	(03:45 PM)	
▶ Comments	"Python Cookbook, 2nd ...	Daily Python-URL! (fr...	(03:45 PM)	
▶ Feedster	Generator methods	Daily Python-URL! (fr...	(03:45 PM)	
▶ Python	Pascal's triangle	Daily Python-URL! (fr...	(03:45 PM)	
▶ Software (25)				
▶ Comm... (33)				
▶ Code (8)				
▶ OS X (10)				
▶ NYC (52)				
▶ Other (44)				
▶ Apple (53)				
▶ Science (6)				
▶ News (40)				
▶ Techn...i (12)				

PyCon Sprints Day 3

Day 3 was the first day of the Chandler sprint. We had two people who joined us with the goal of learning to write a parcel. We've split into two groups. One group is working on a parcel for working with del.icio.us bookmarks, and the other group is working on a parcel for grabbing pictures from Flickr.

A sprint like this is a good way to get live feedback, and we've gotten a lot of feedback so far. The two groups are making good progress. We spent a bunch of time getting infrastructure setup -- a subversion repository, some wiki pages to record notes, sorting out the various builds, and doing some tutorial/walkthrough. So it wasn't until after lunch until the groups sat down to work on their projects.

So far things are going pretty well -- it looks like both groups will get basic functionality done by the end of tomorrow, and I'd like to be able to show the fruits of their labor as part of our presentation on Thursday. We have gotten a lot of feedback about things that are hard, error-prone, or redundant. It's also clear that we need to work some more on documentation. So it feels like there's good information exchange -- Bill, Mark, and Kragen

<http://www.sauria.com/blog/2005/03/22#1253> Default



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
- valueForKey: (like `__getattr__`)

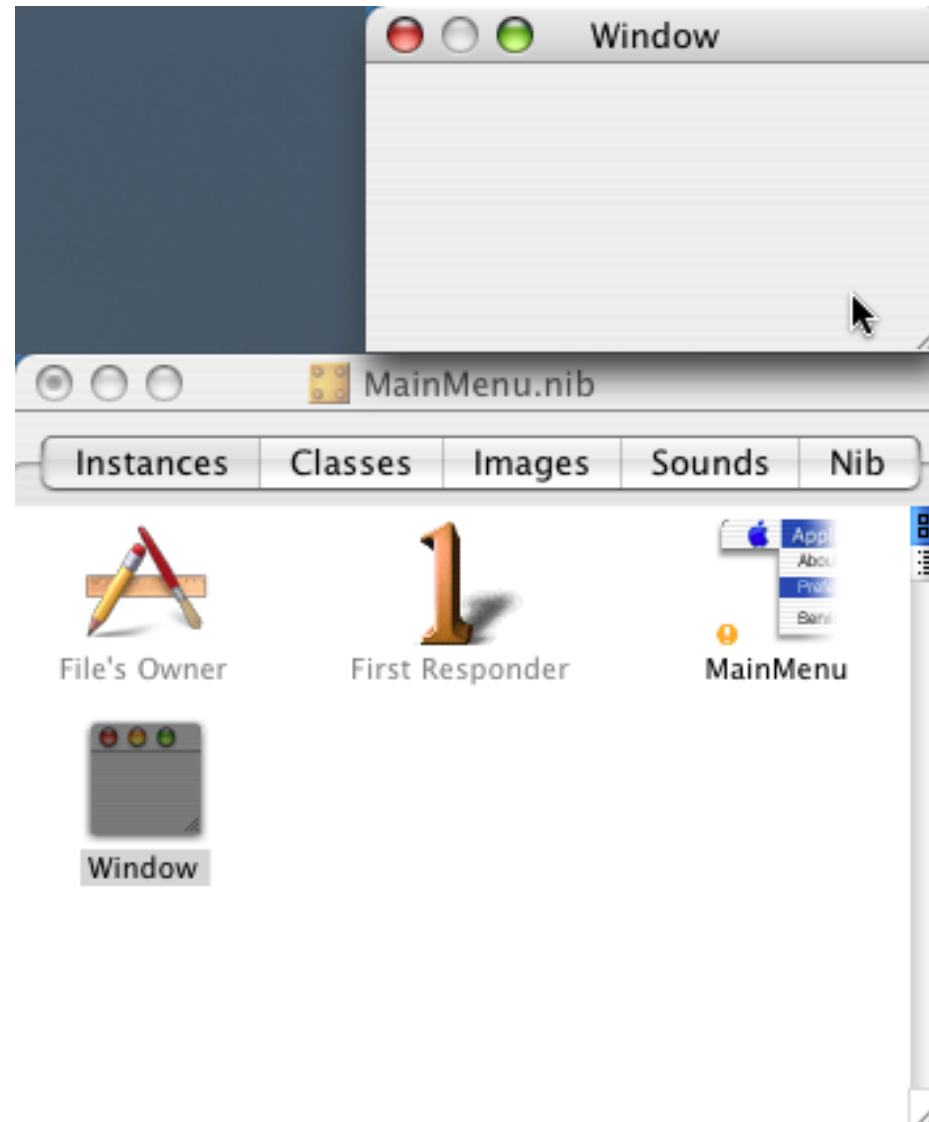


Making Money

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

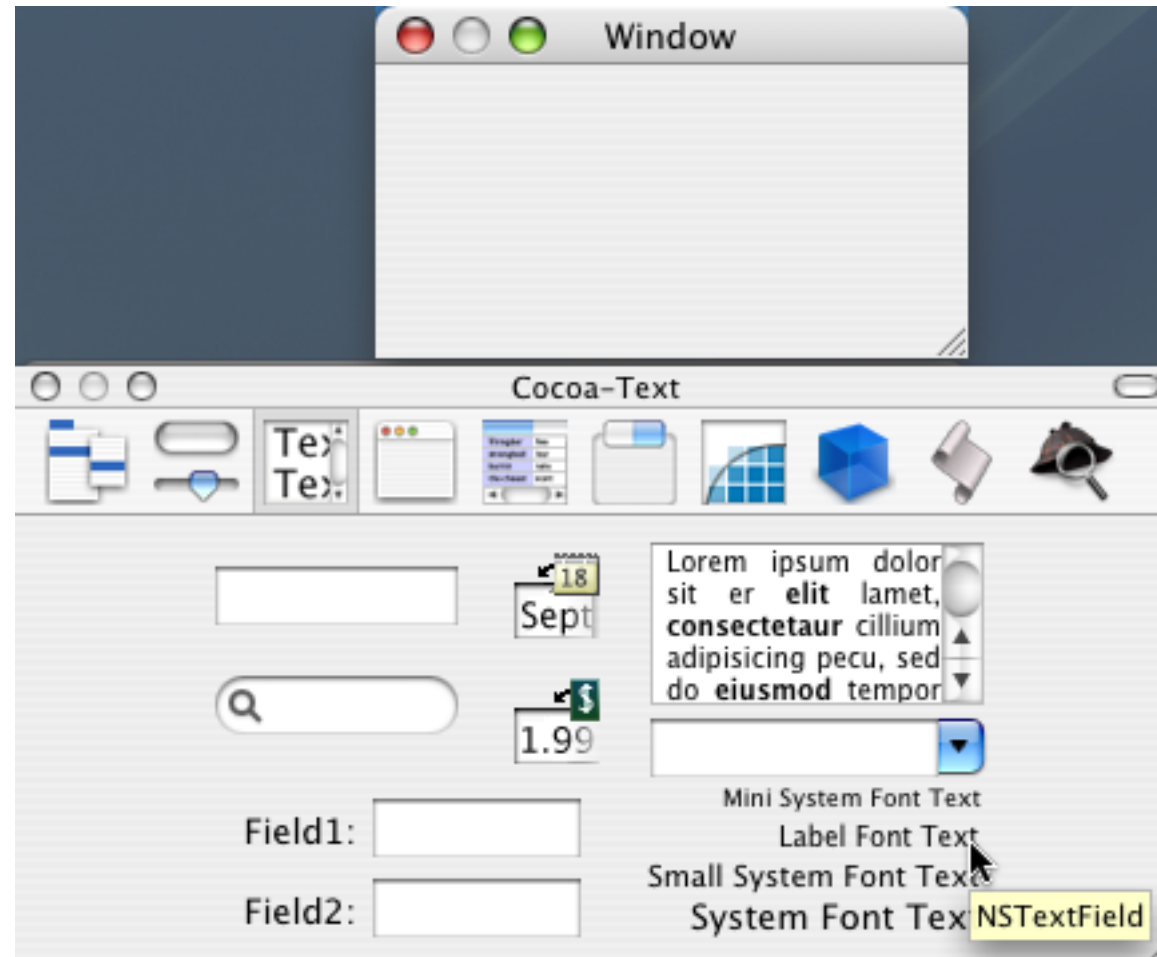


New Application in IB



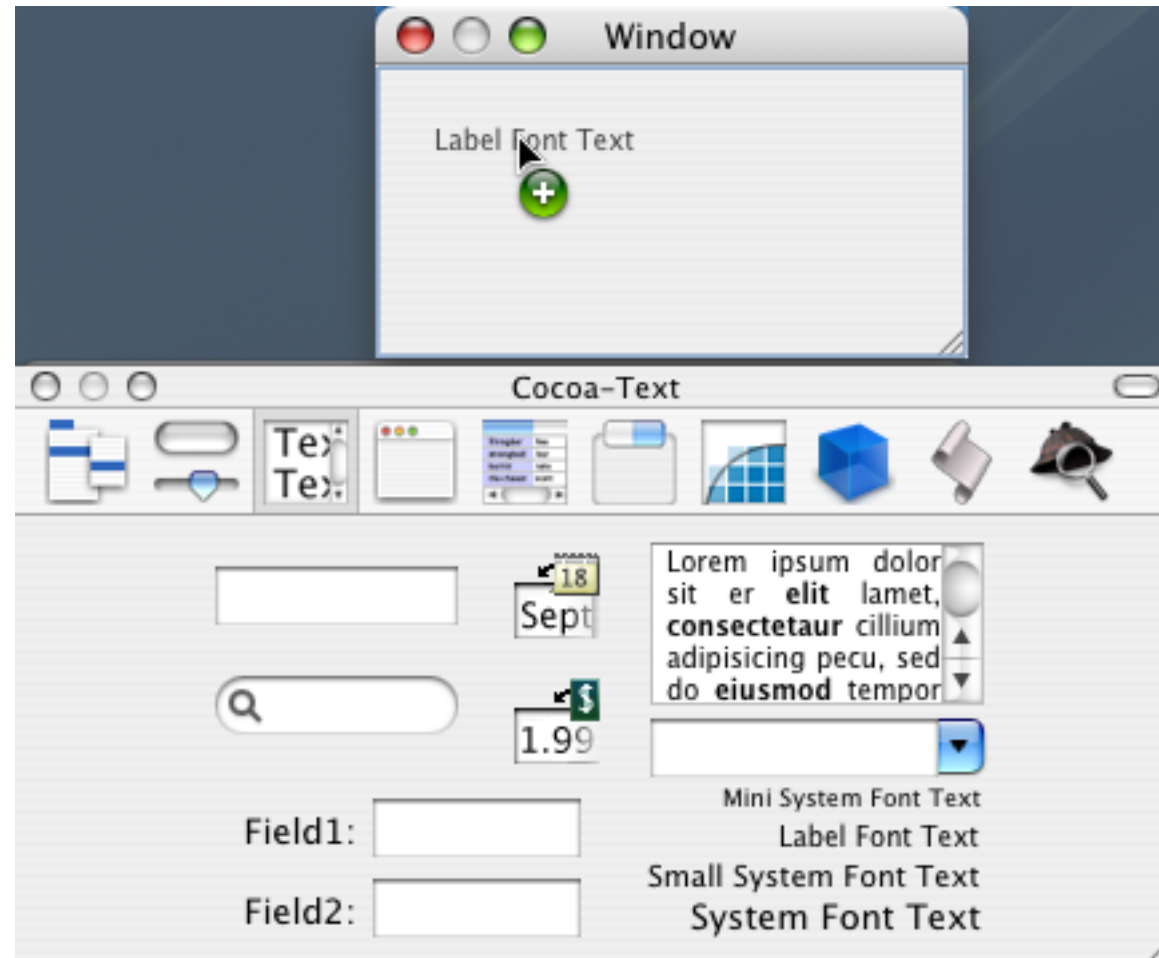


Create an NSTextField



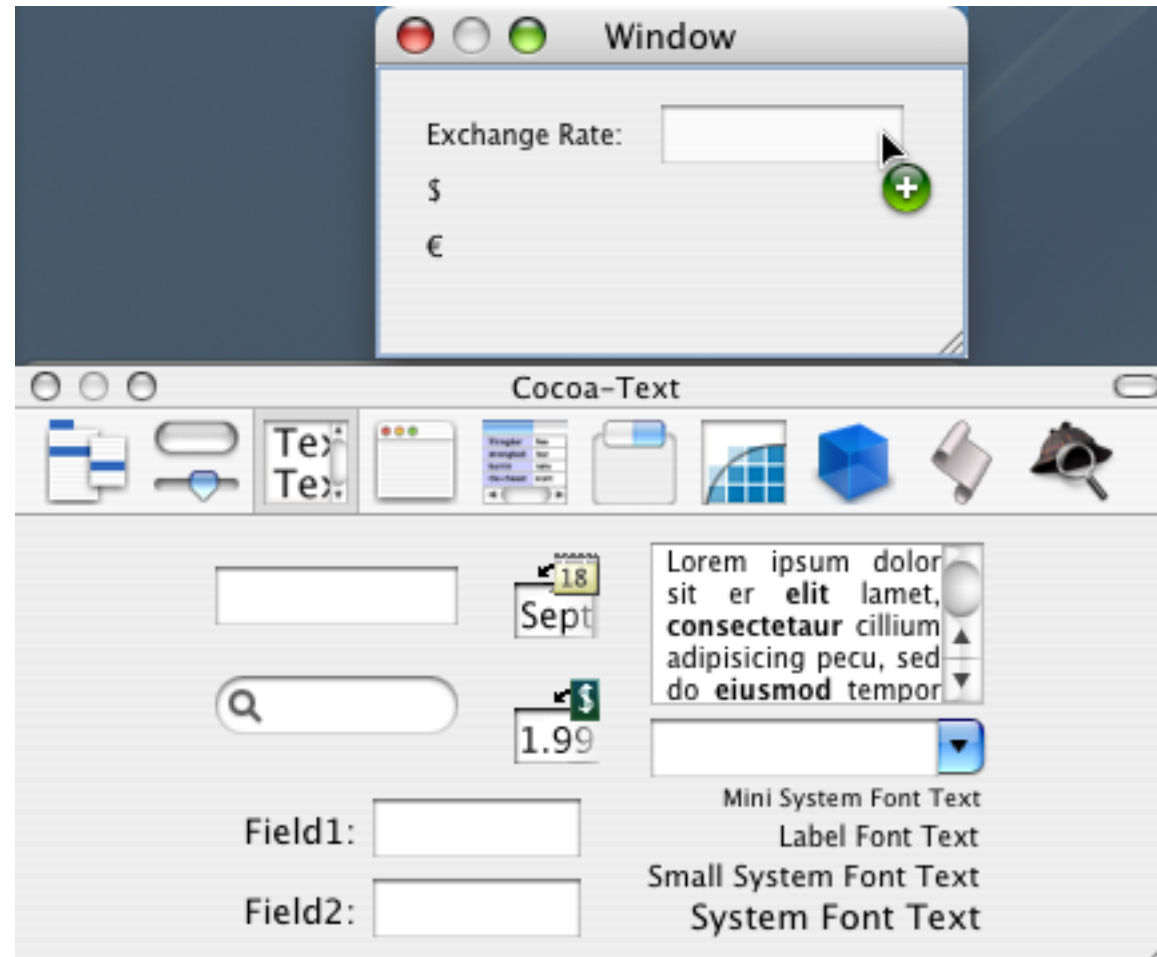


Drag to the NSWindow



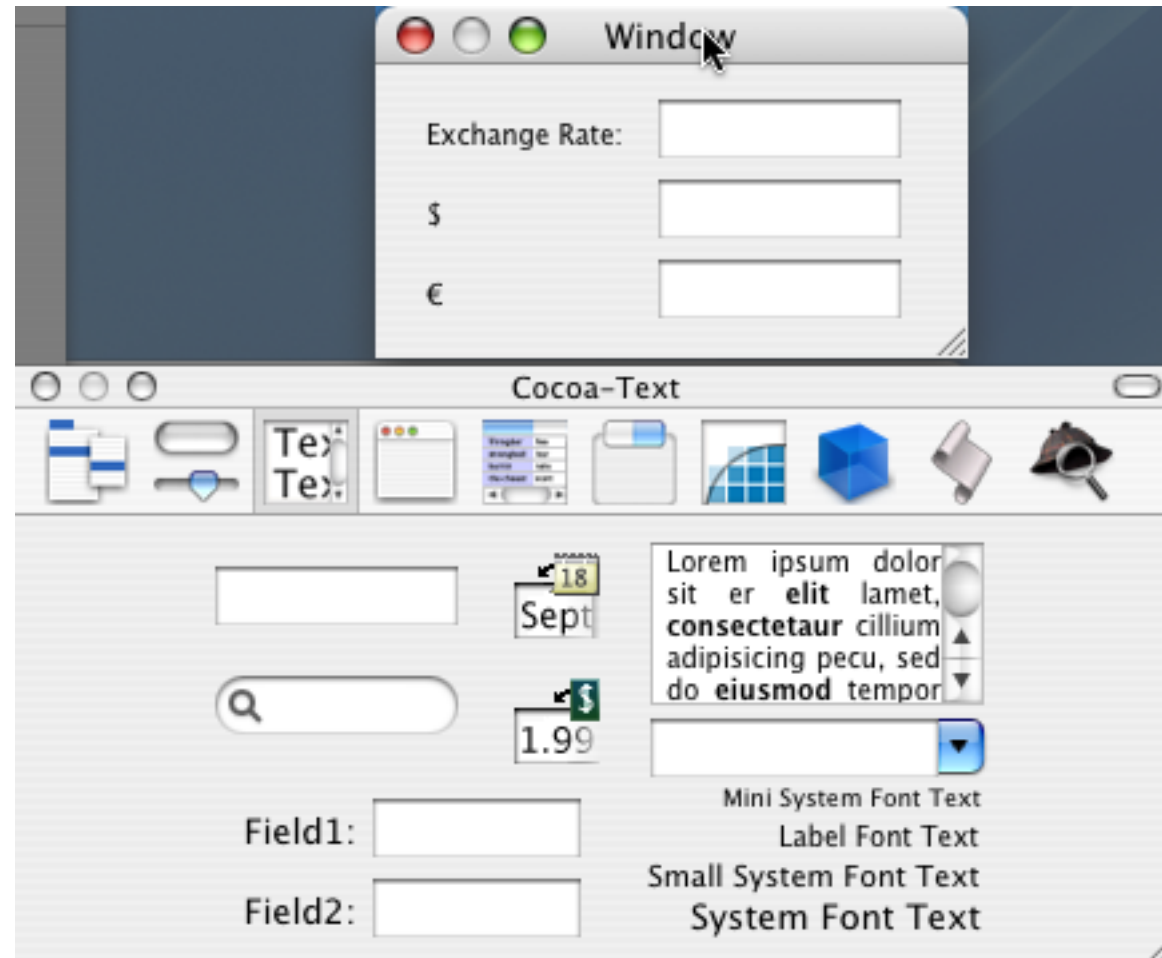


Create the input NSTextFields



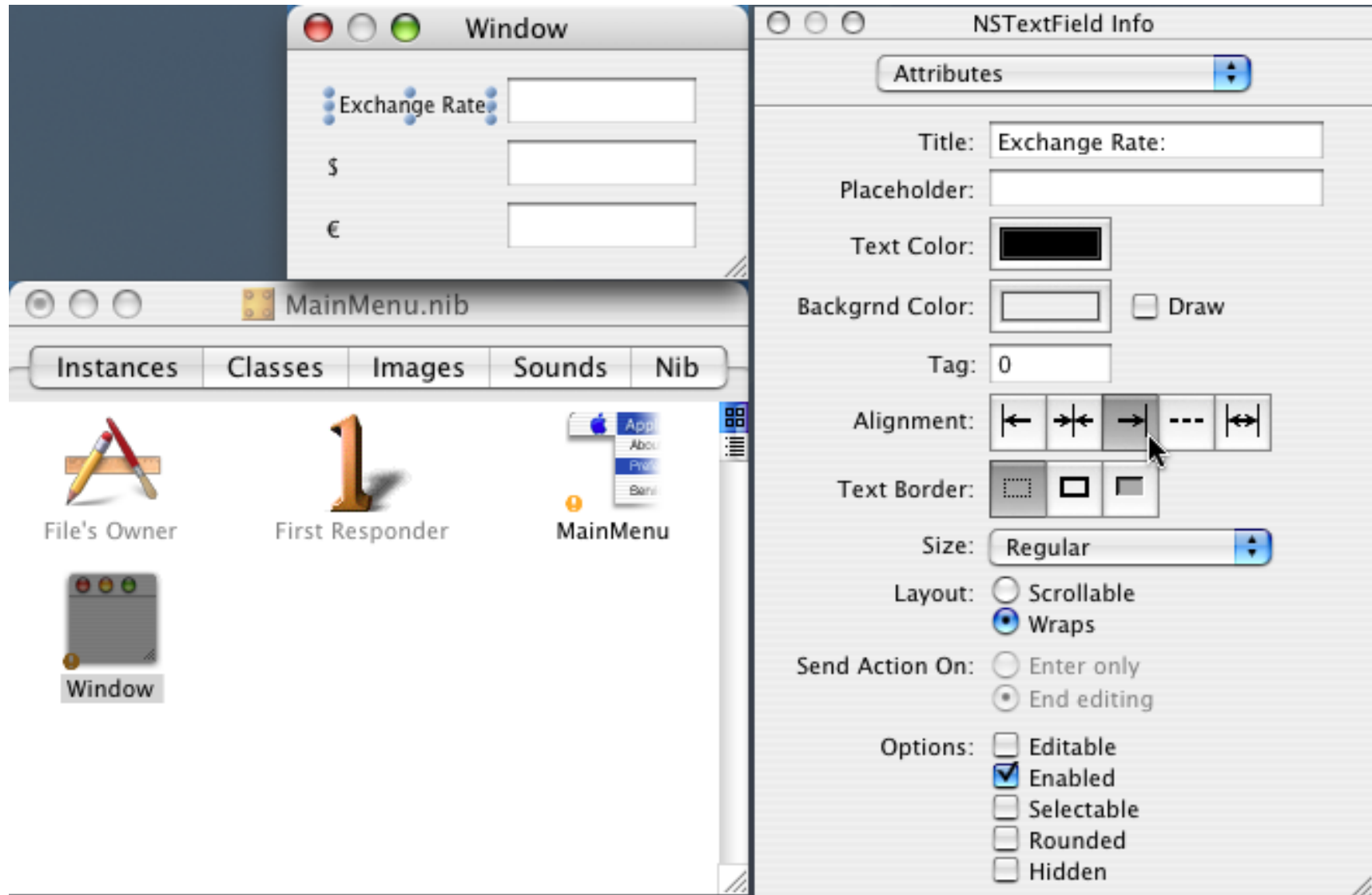


Almost finished UI Layout



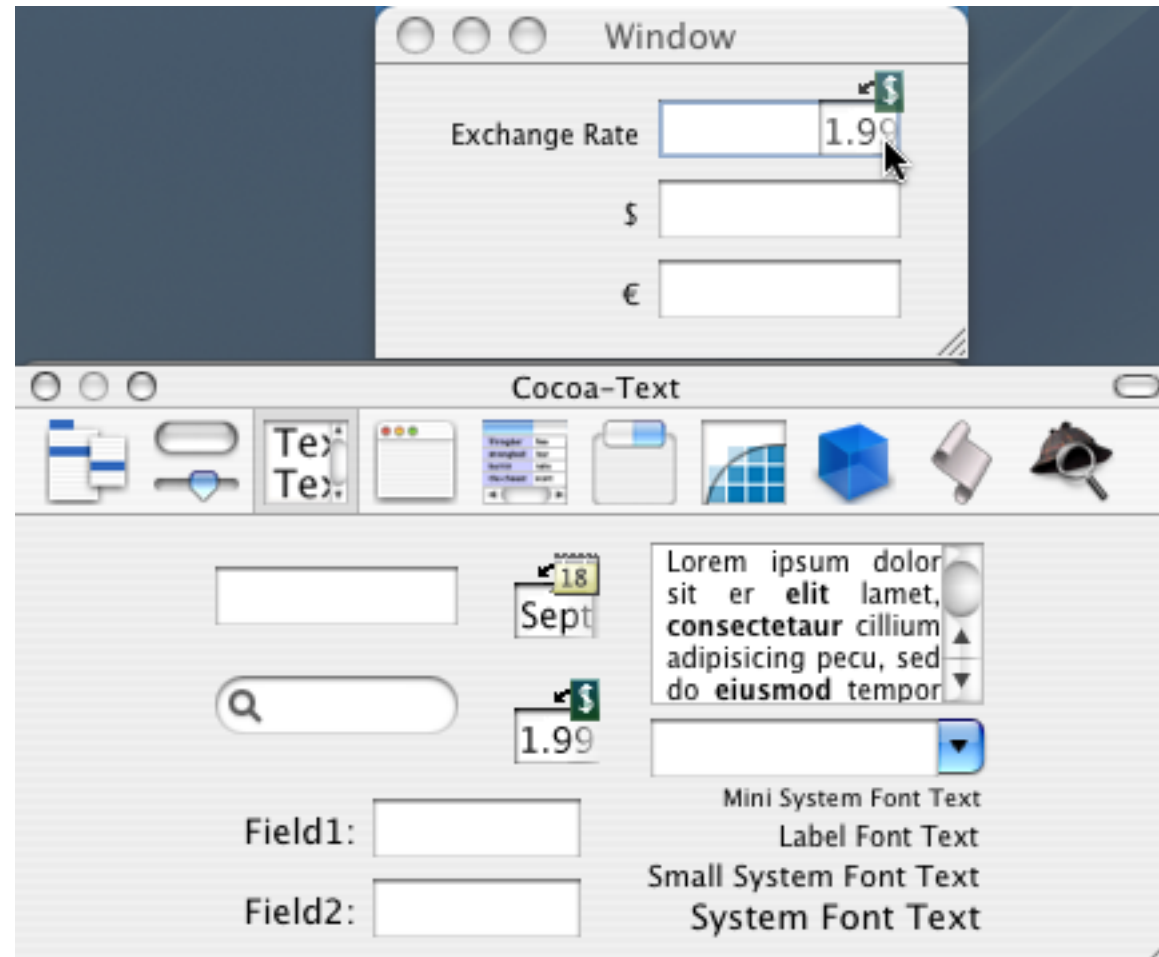


Align the labels



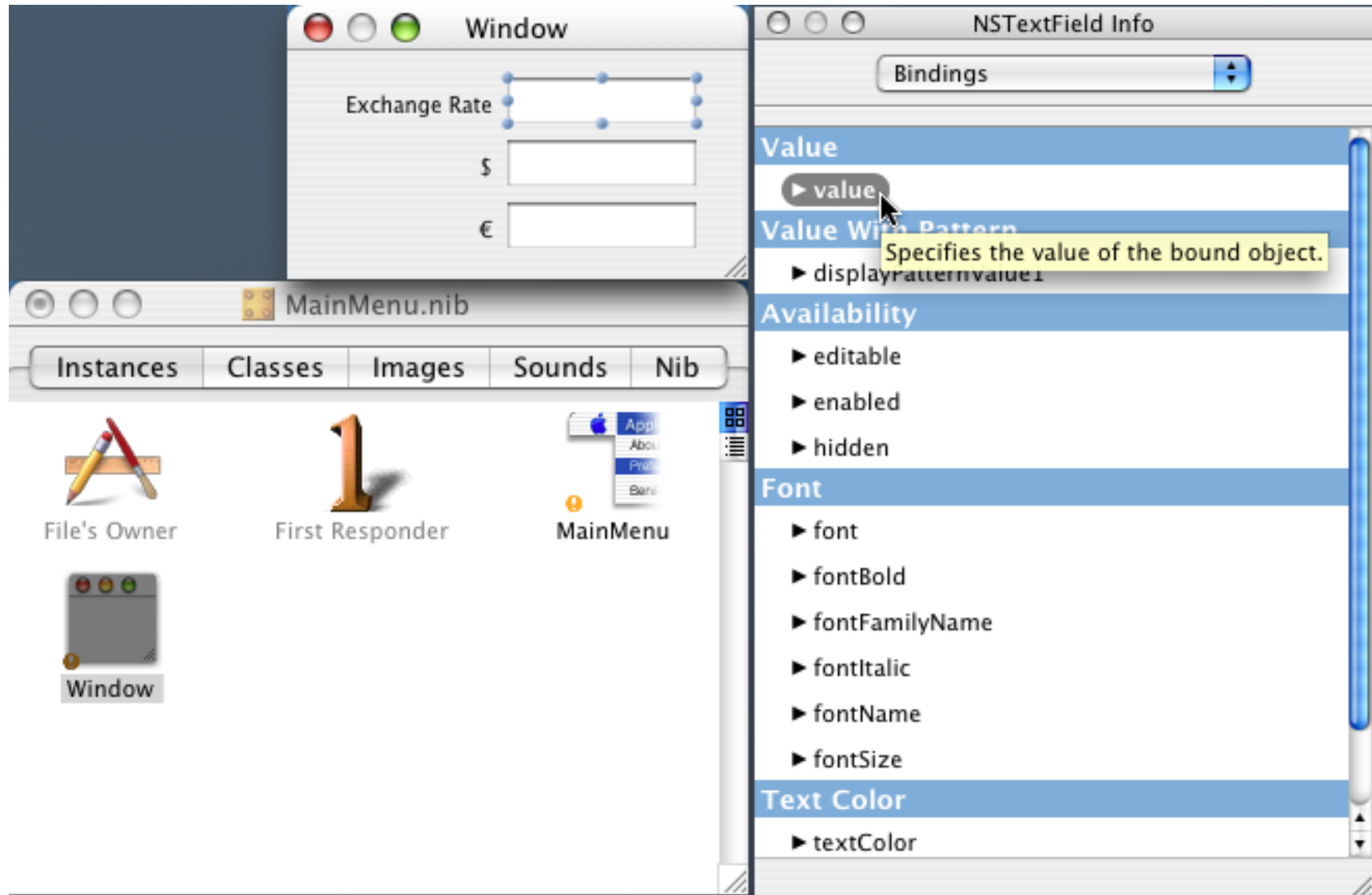


Use currency NSNumberFormatters



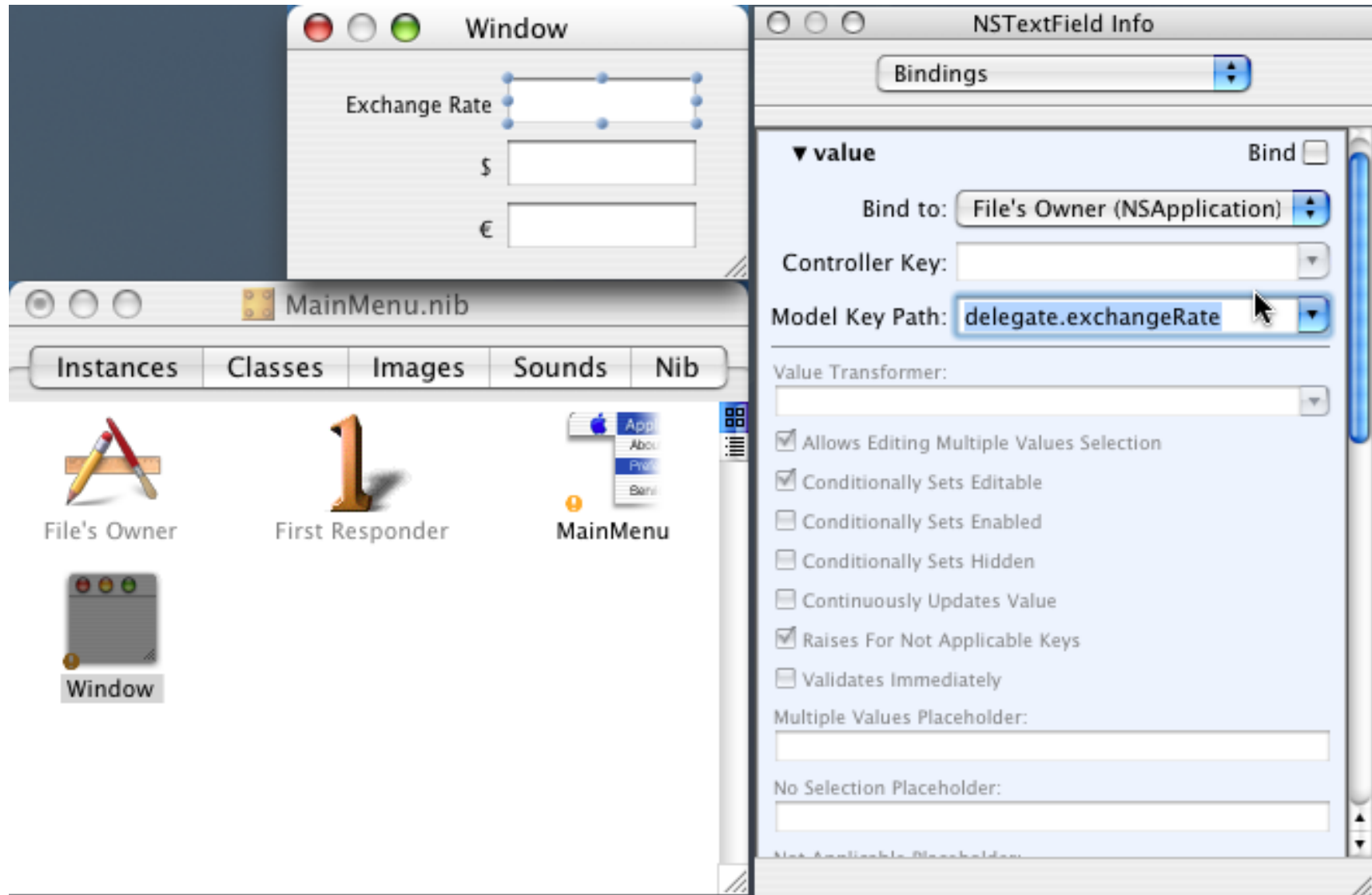


Set up the Bindings



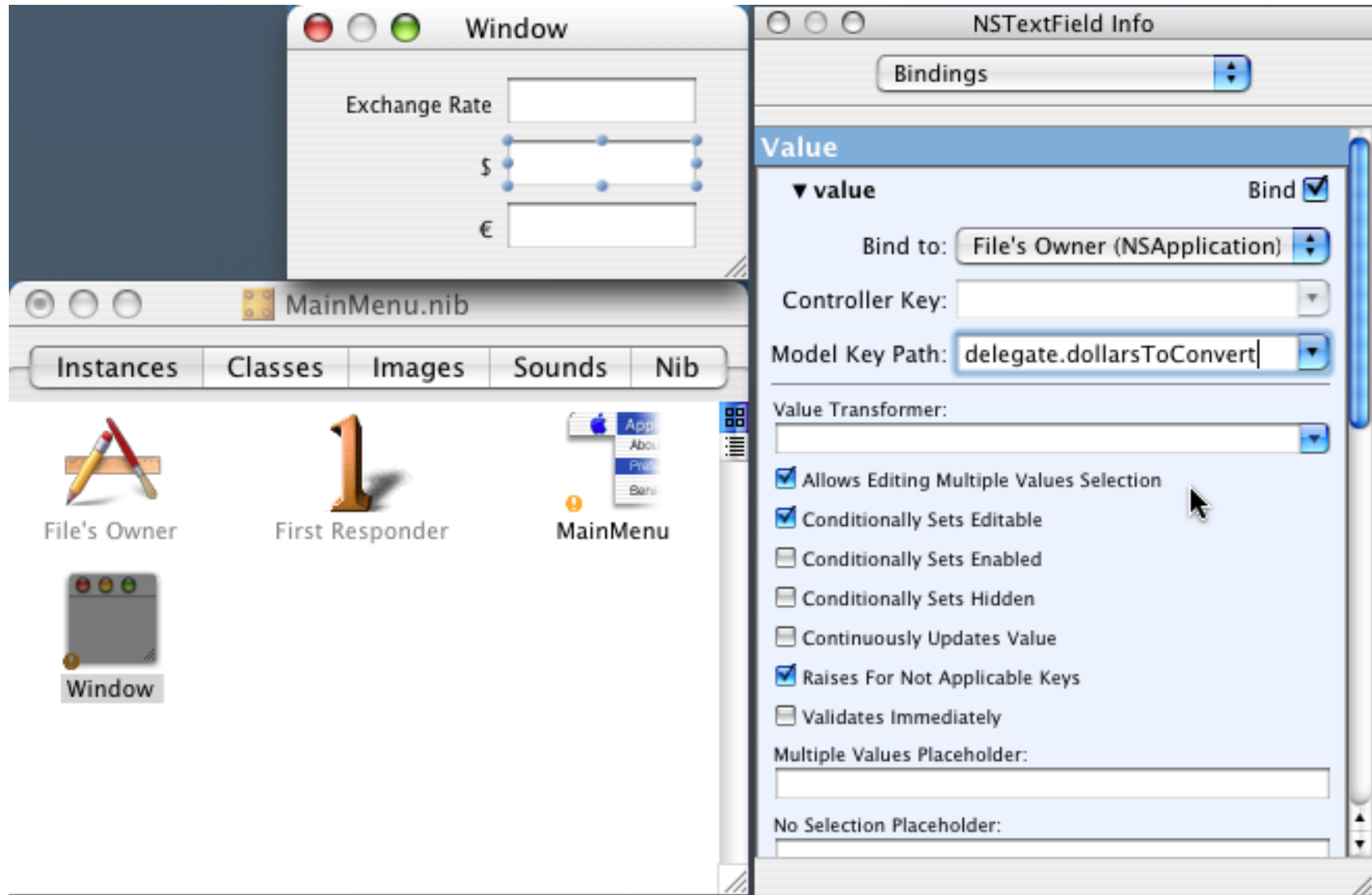


To point to your delegate



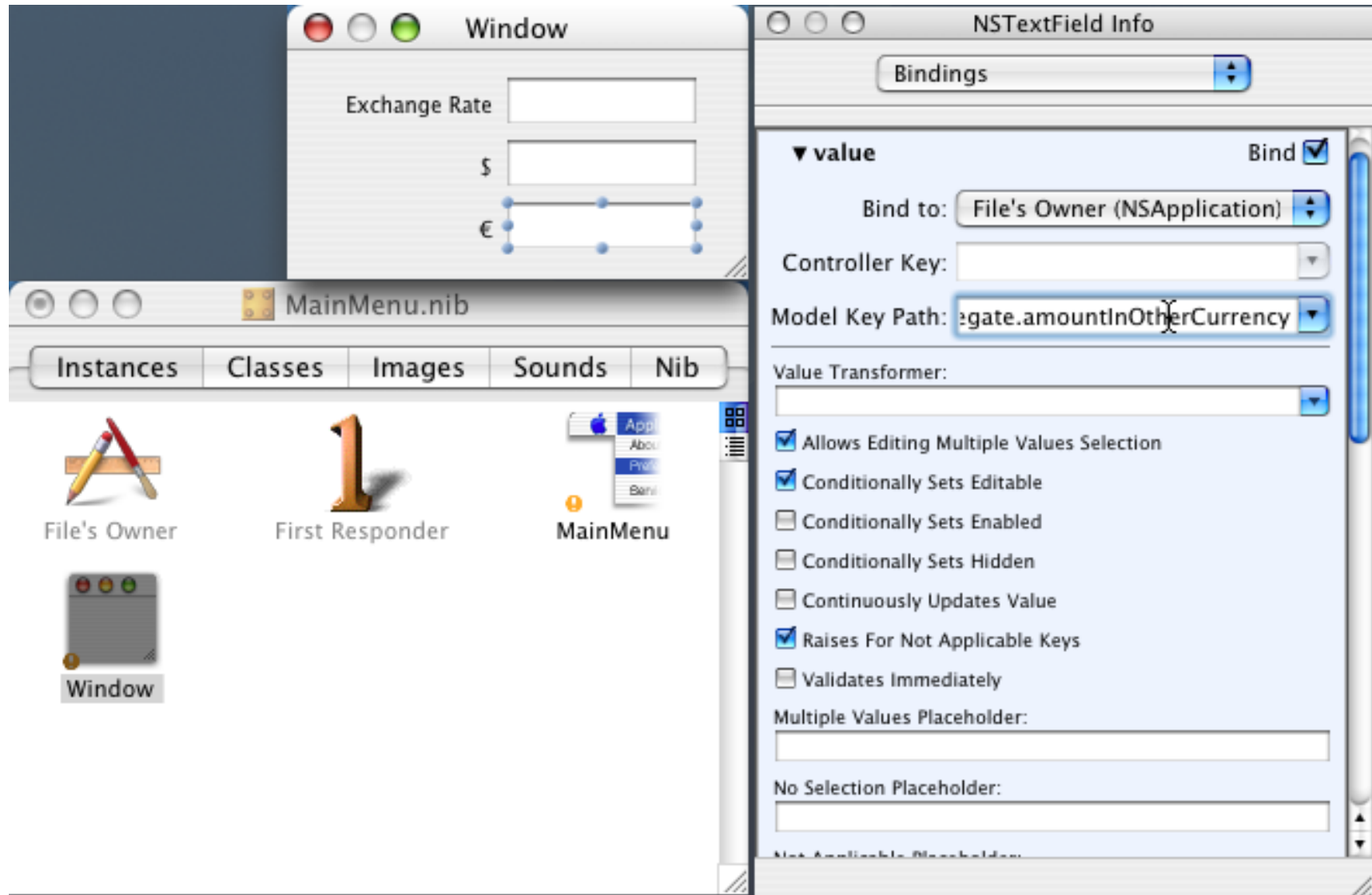


Dollars binding...



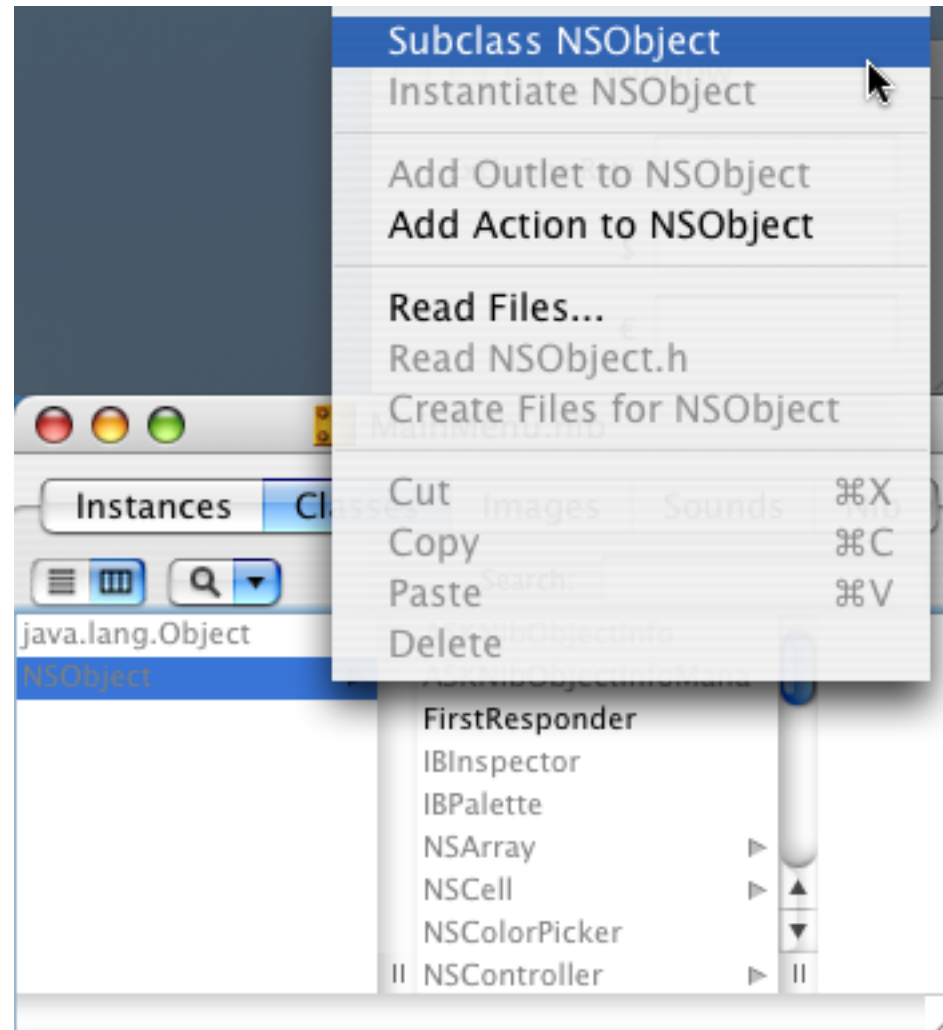


Other Currency Binding...



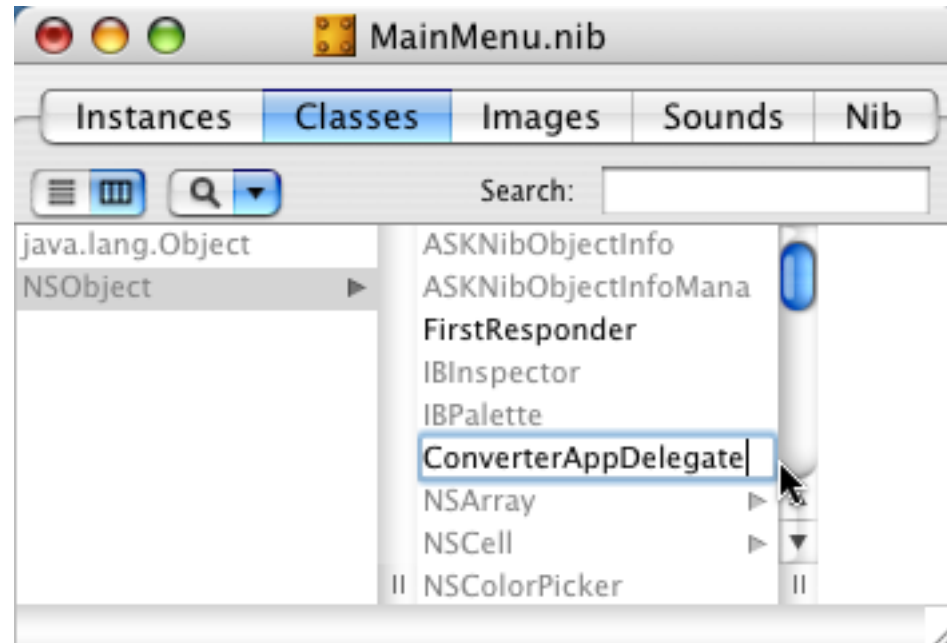


Subclass NSObject



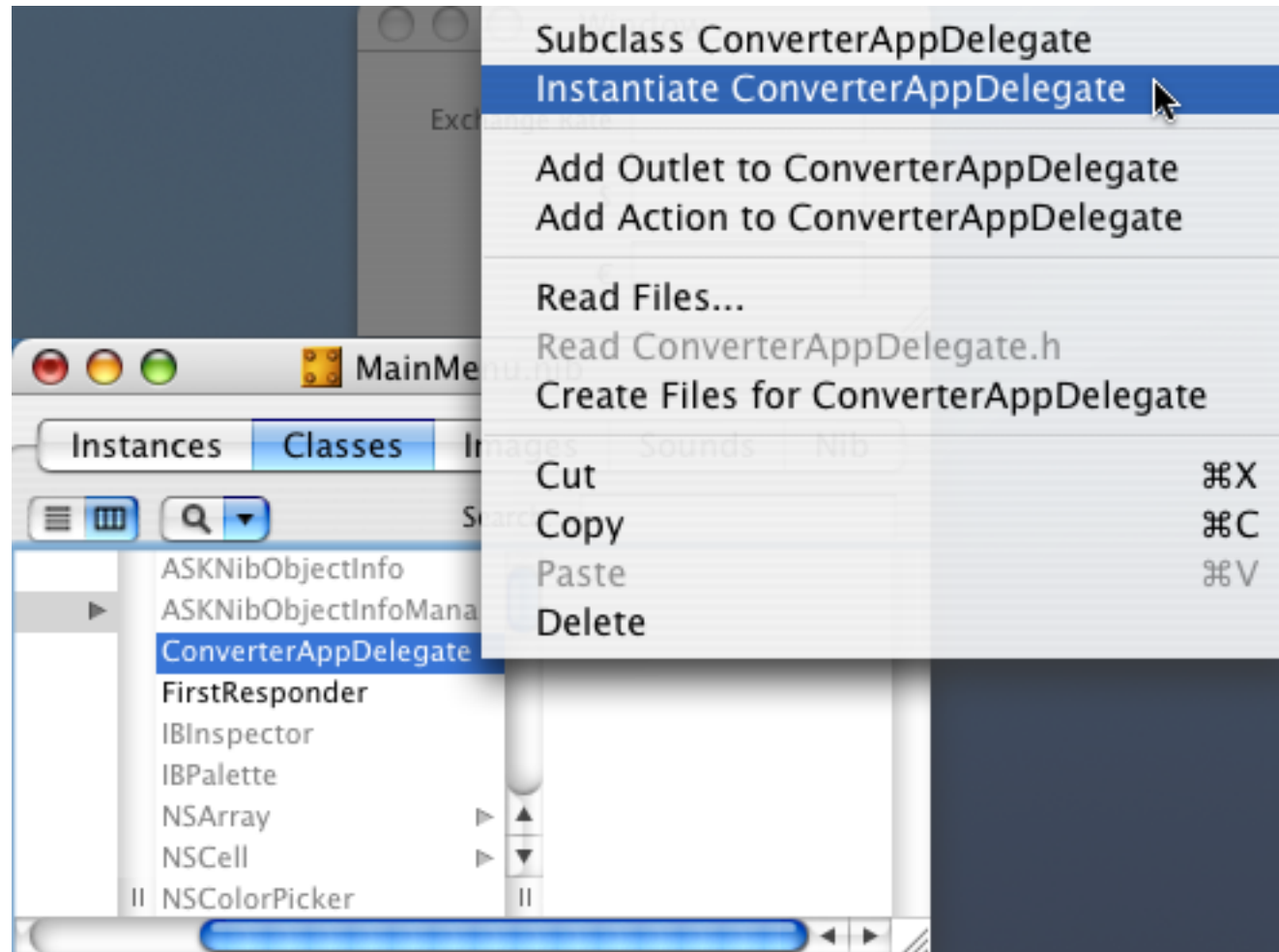


To create your delegate class



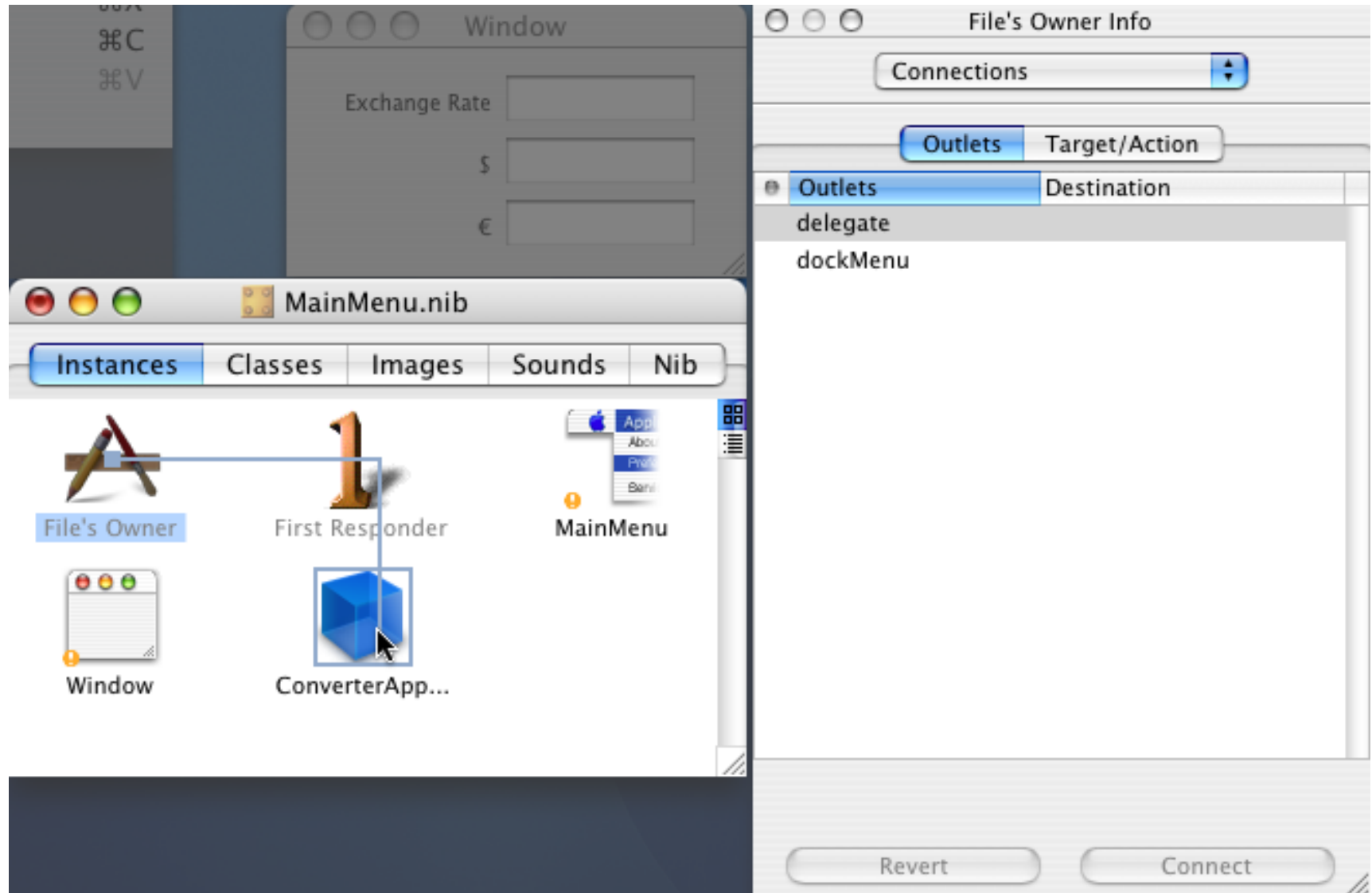


Instantiate it in your nib



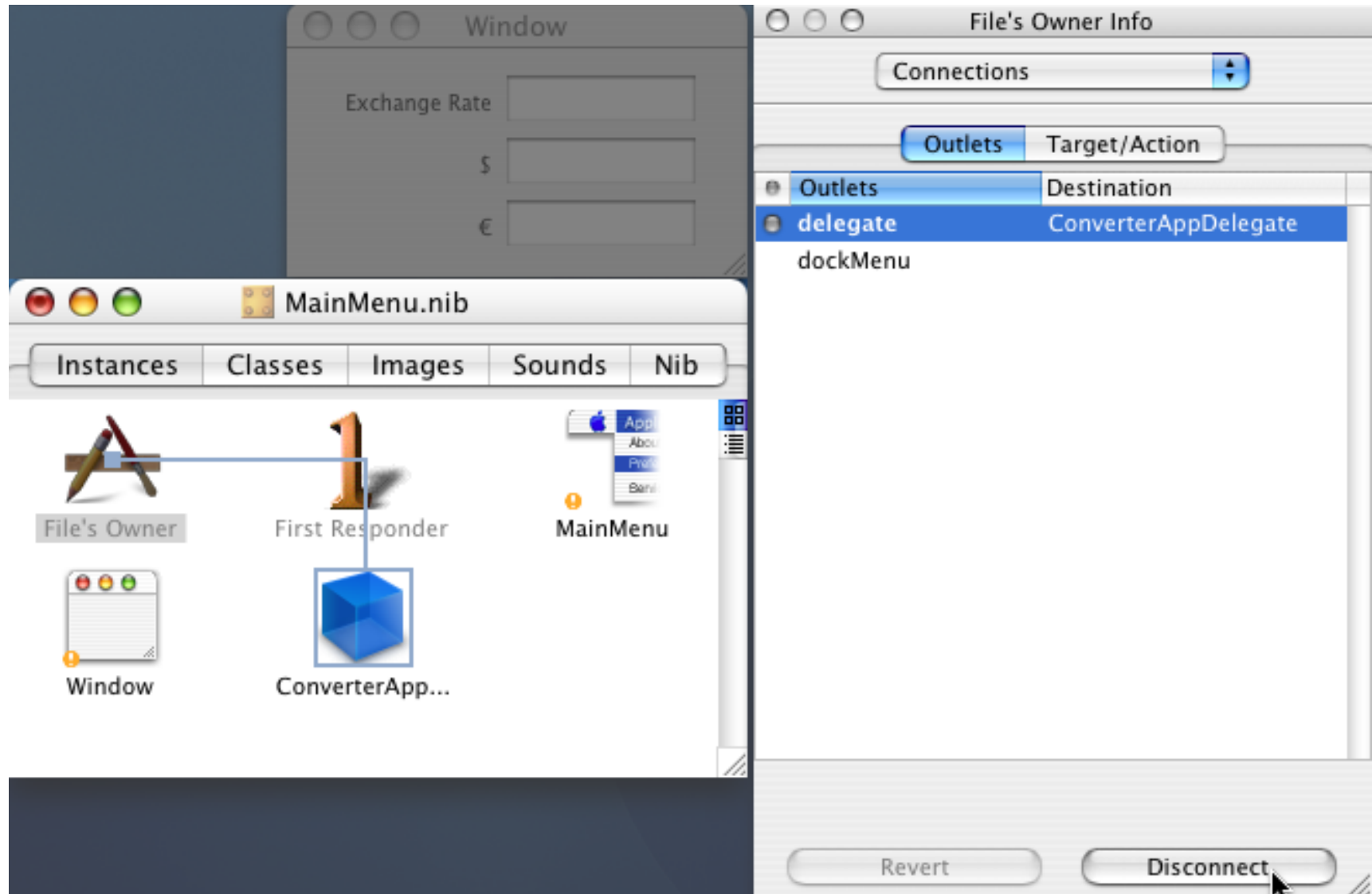


Create a connection





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

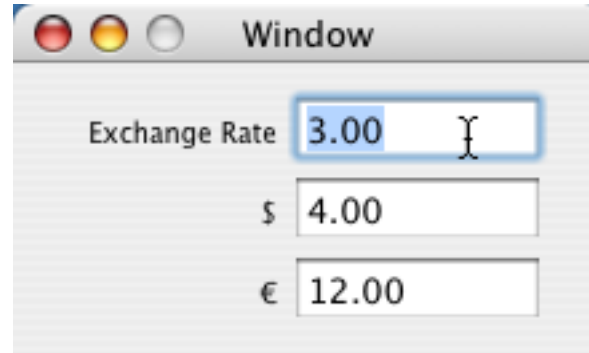
Build:

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

Run:

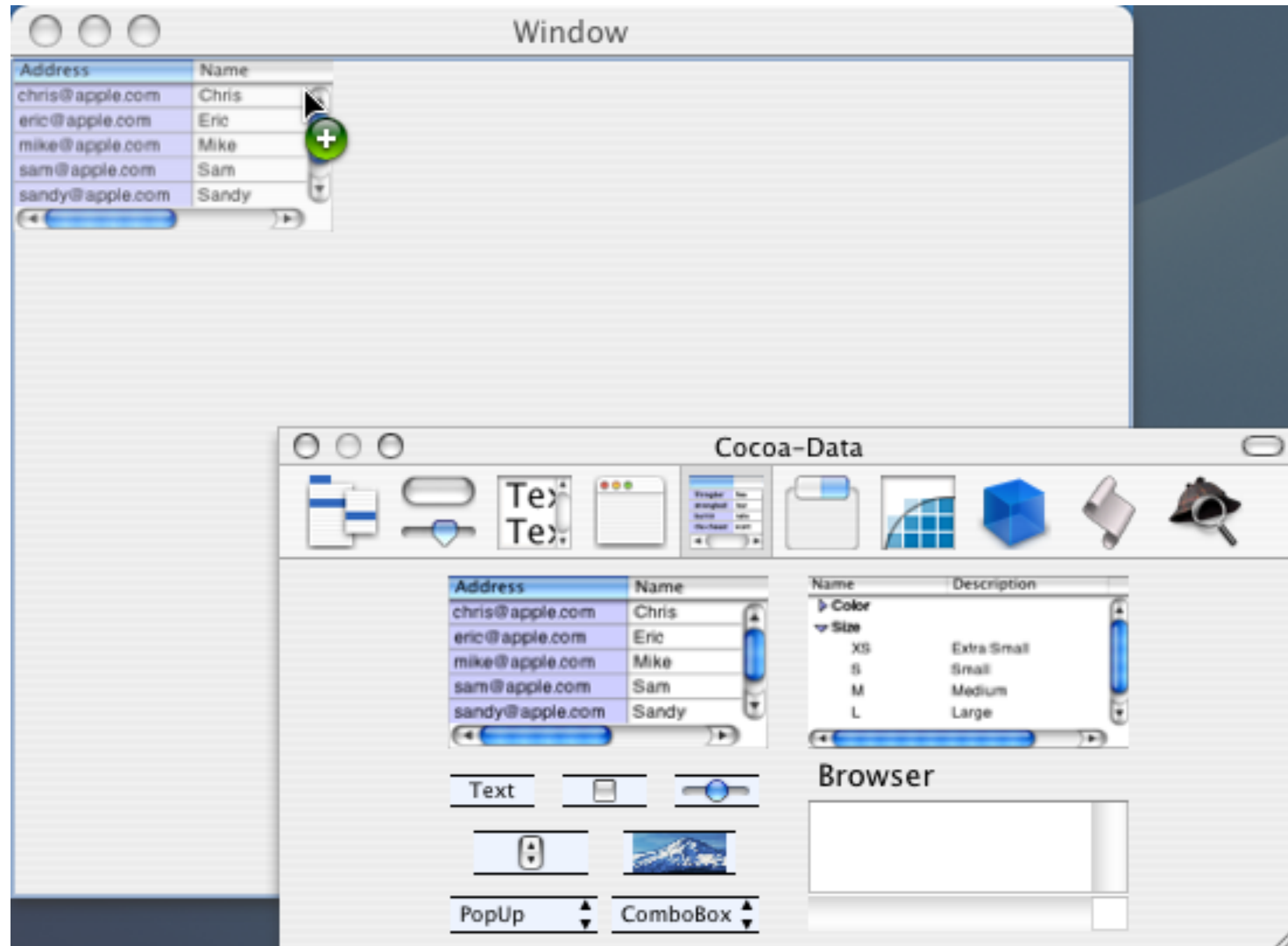
```
% open dist/Converter.app
```

Done:





New NSTableView



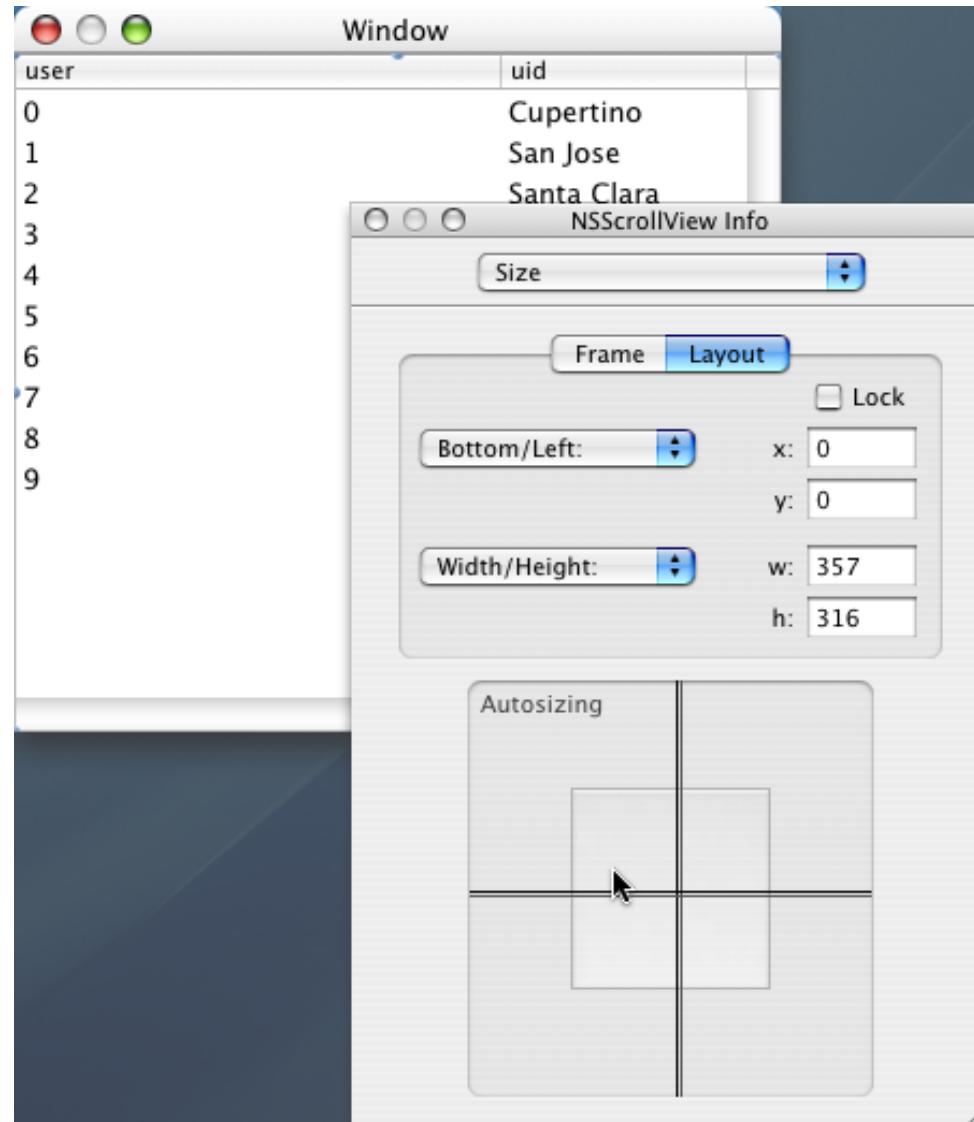


Name the columns

Window	
user	
0	Cupertino
1	San Jose
2	Santa Clara
3	San Francisco
4	Palo Alto
5	San Carlos
6	Los Gatos
7	Sunnyvale
8	Mountain View
9	Redwood City

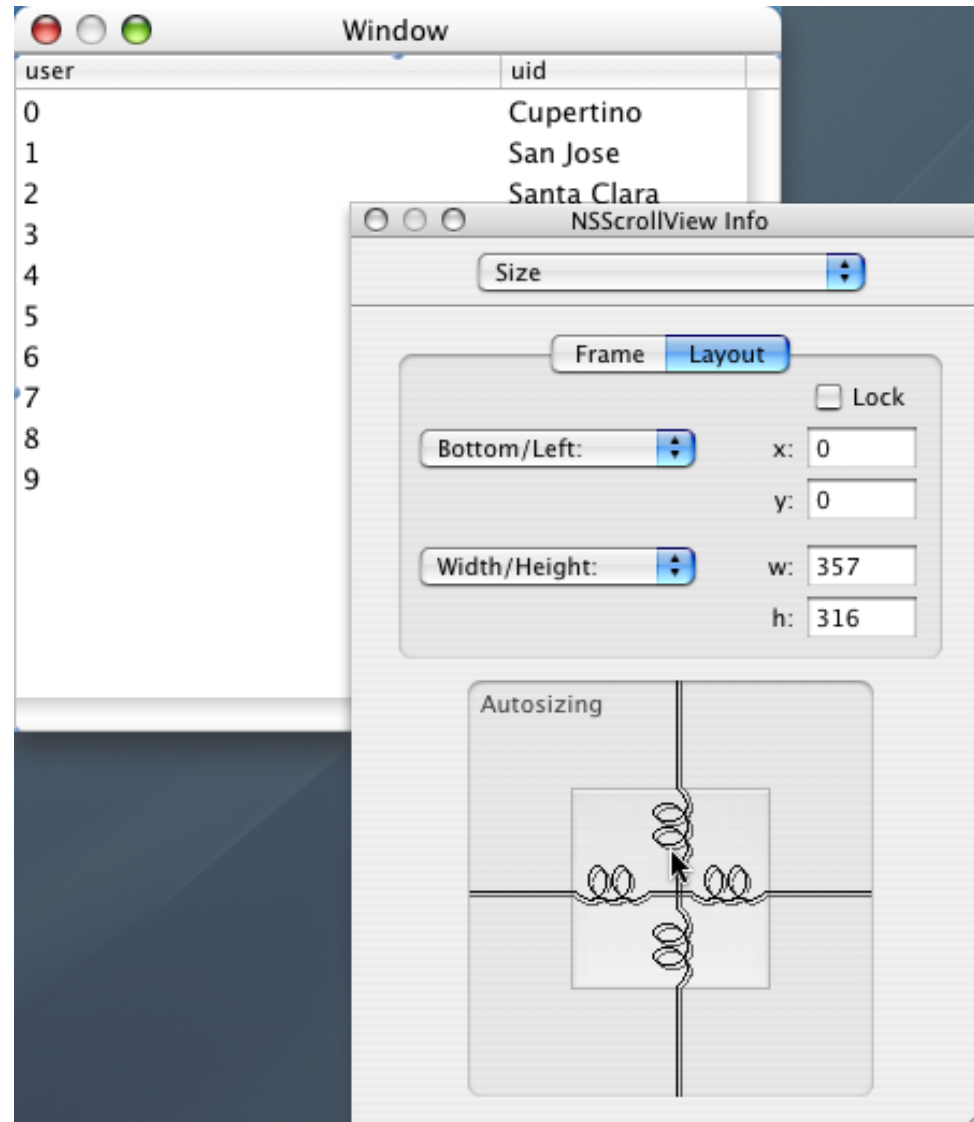


Change the resize behavior



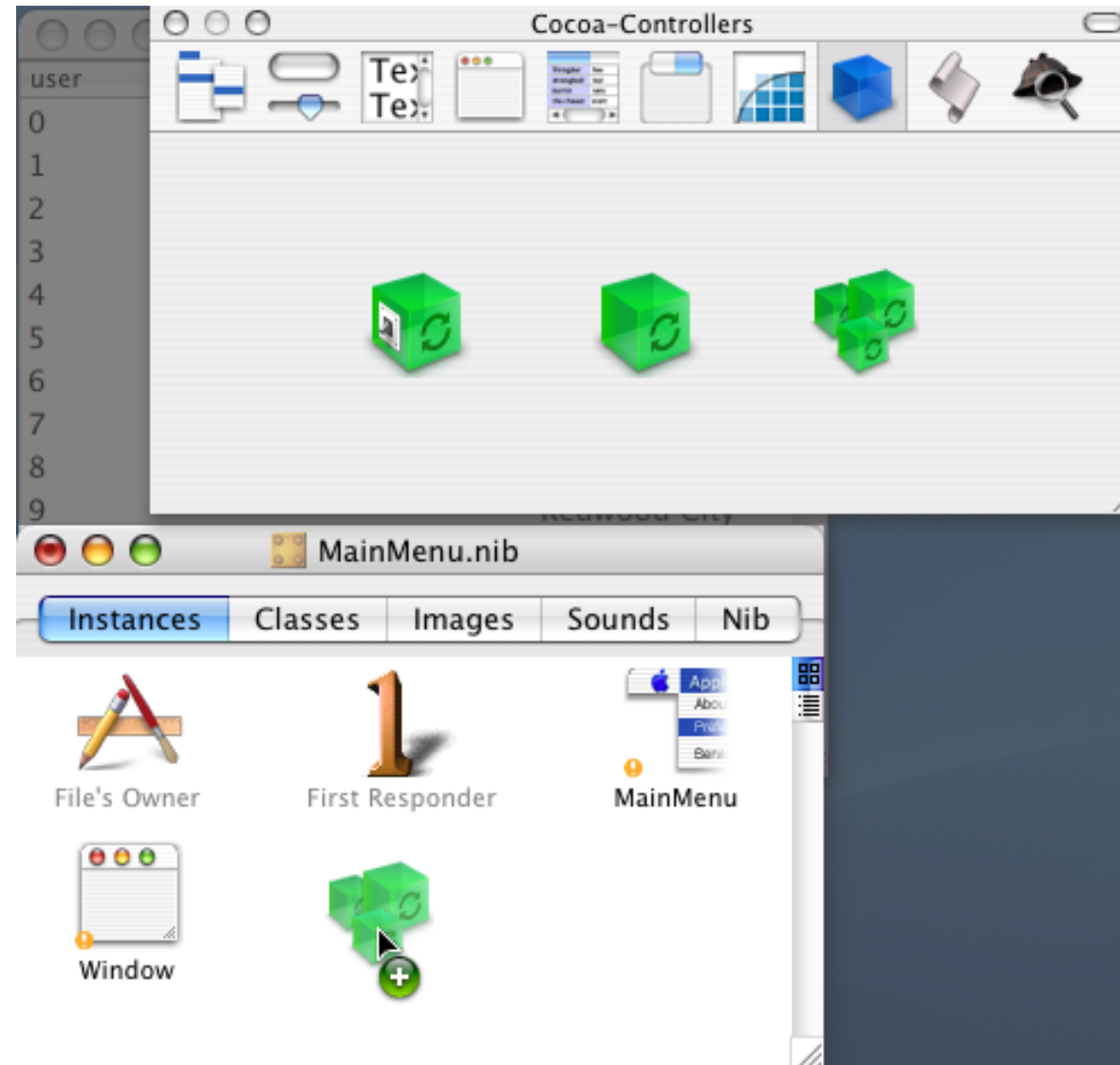


To expand with the NSWindow



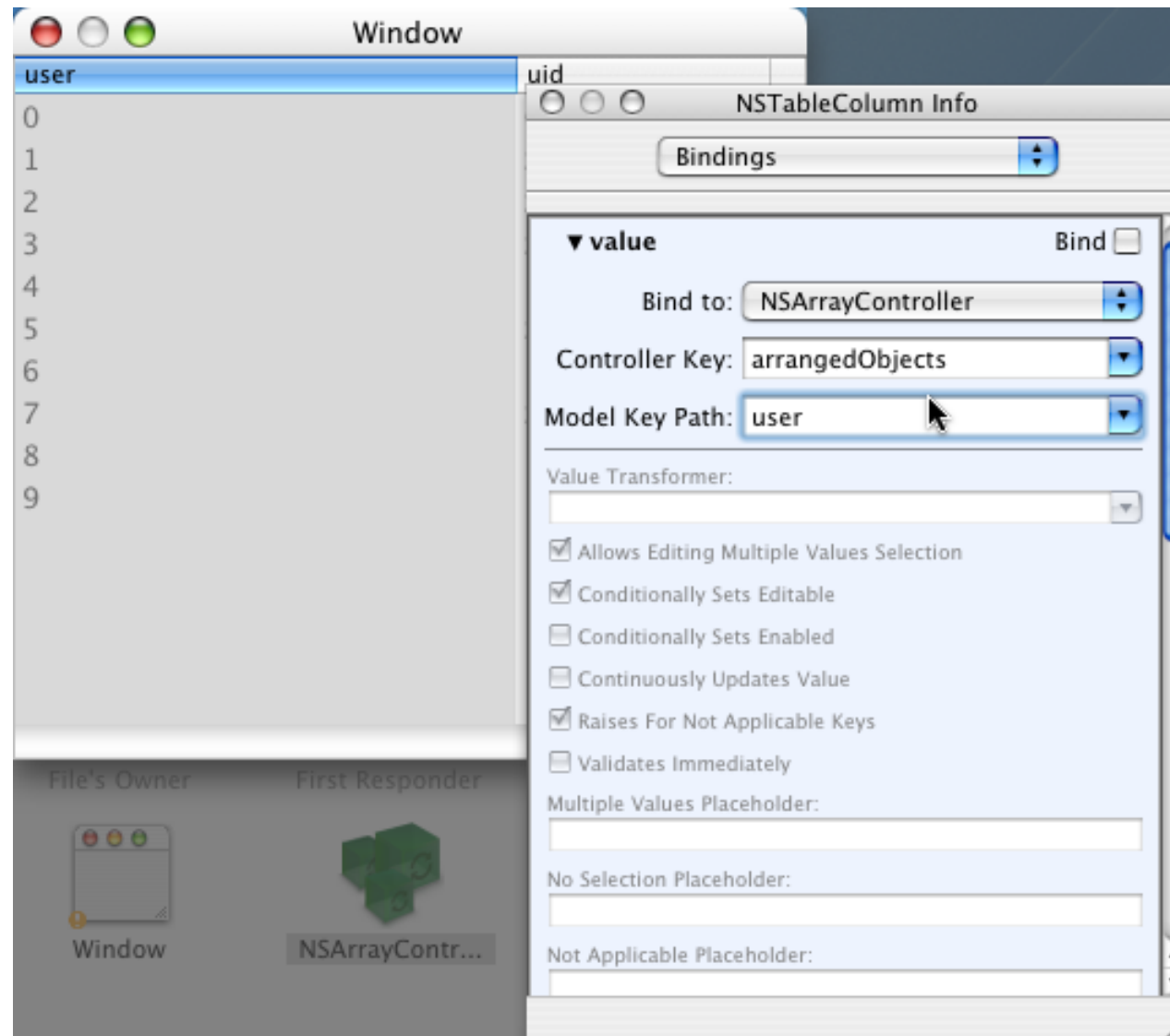


Create an NSArrayController



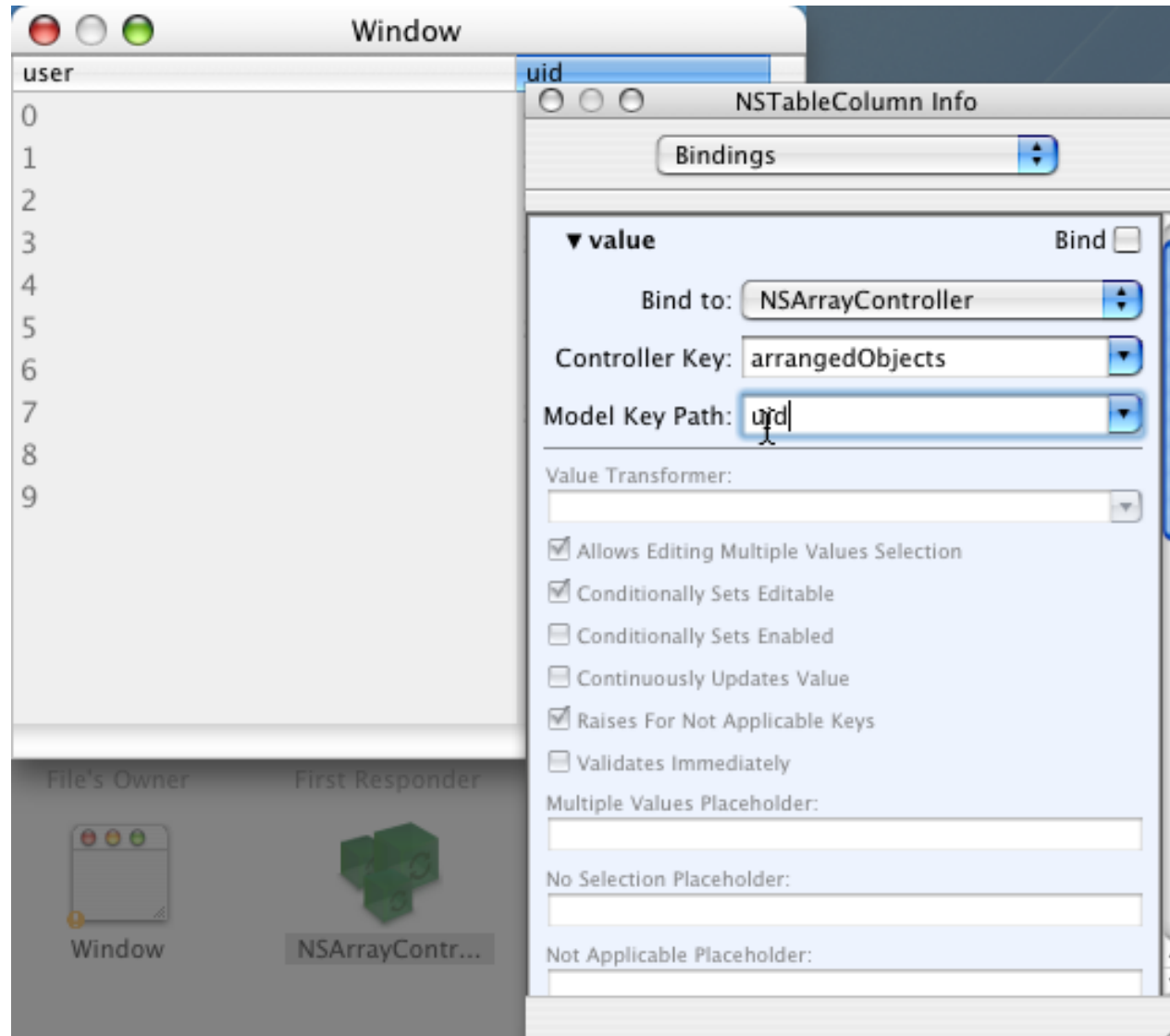


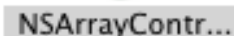
Bind the user column





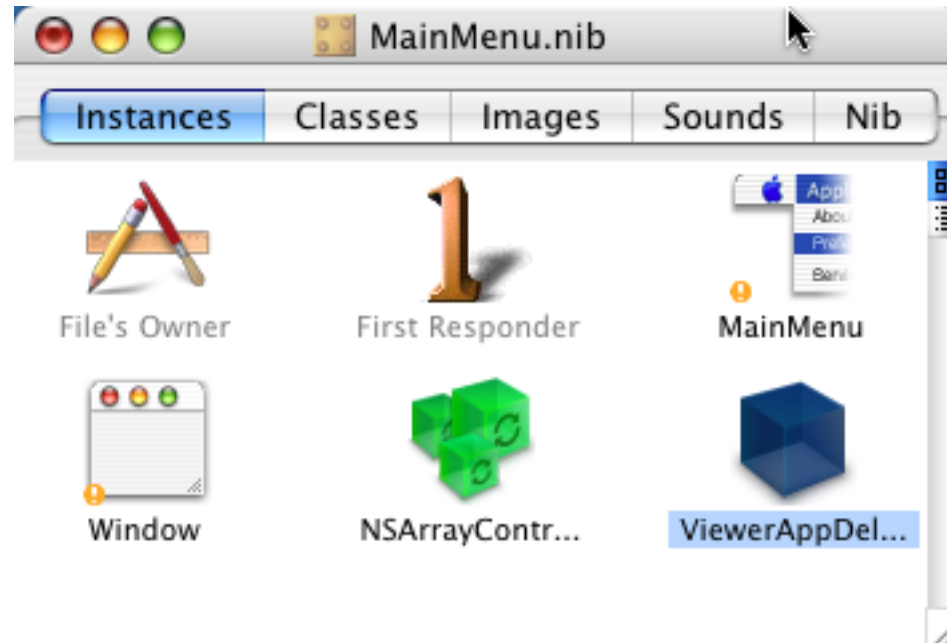
Bind the uid column







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 gecost 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

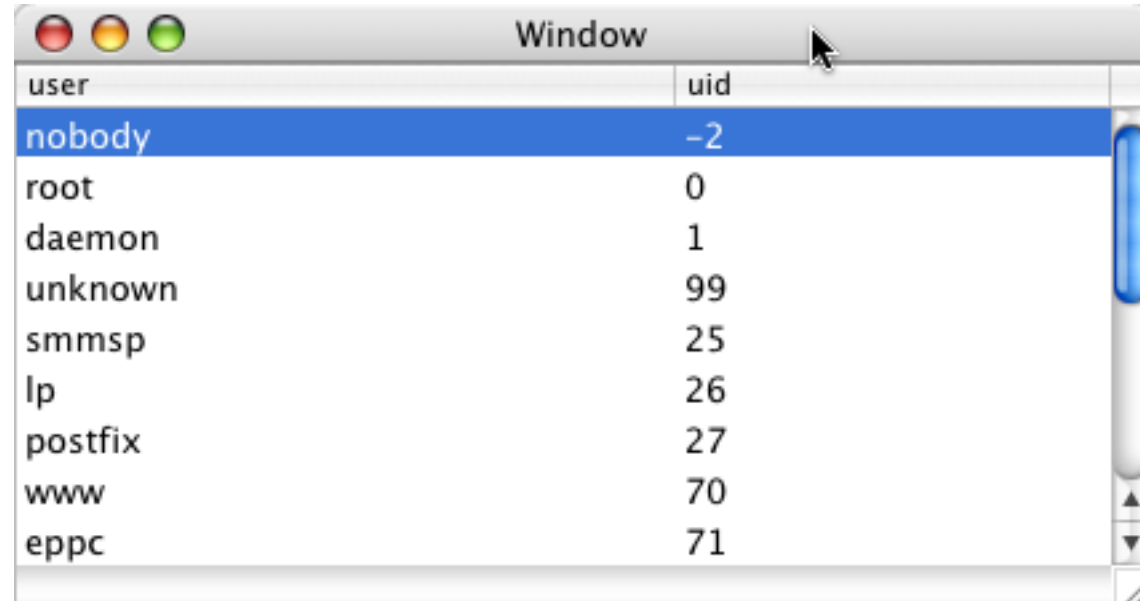
Build (redistributable!):

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

Run:

```
% open Viewer.app
```

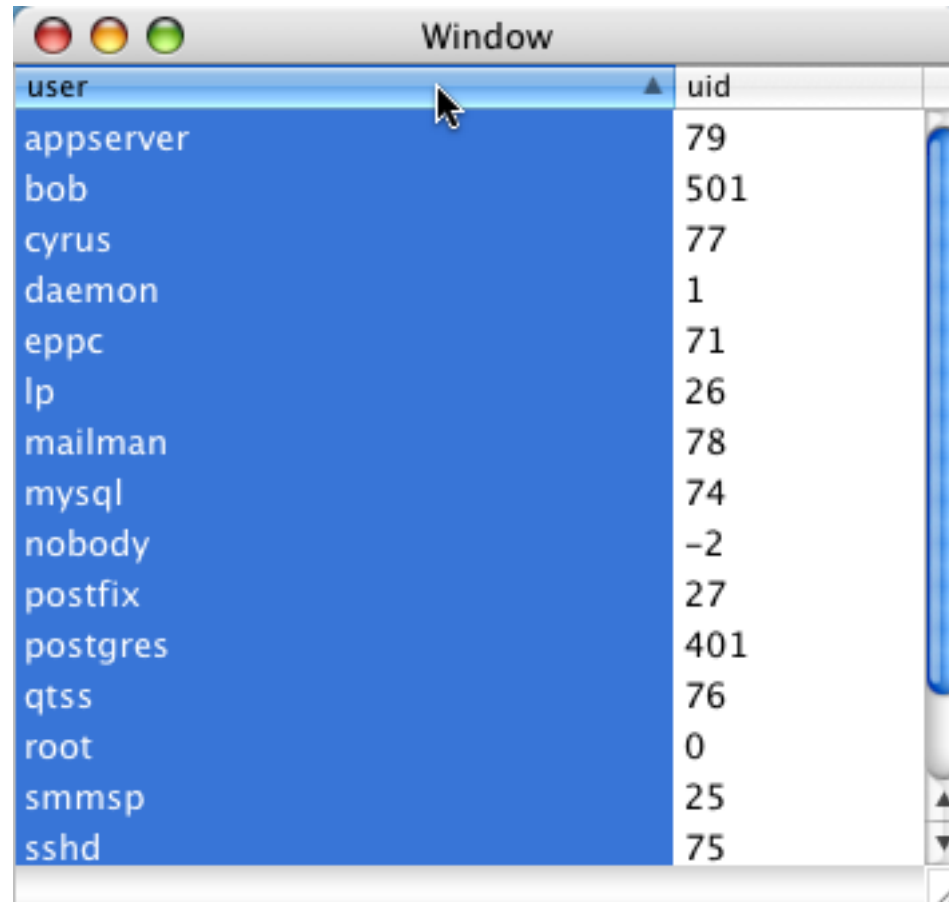
Done:



user	uid
nobody	-2
root	0
daemon	1
unknown	99
smmsp	25
lp	26
postfix	27
www	70
eppc	71



Bindings give you sorting for free!



user	uid
appserver	79
bob	501
cyrus	77
daemon	1
eppc	71
lp	26
mailman	78
mysql	74
nobody	-2
postfix	27
postgres	401
qtss	76
root	0
smmsp	25
sshd	75



Help!

Documentation:

/Developer/Python/PyObjC/Documentation

Examples:

/Developer/Python/PyObjC/Examples

Wiki:

<http://pythonmac.org/wiki>

IRC:

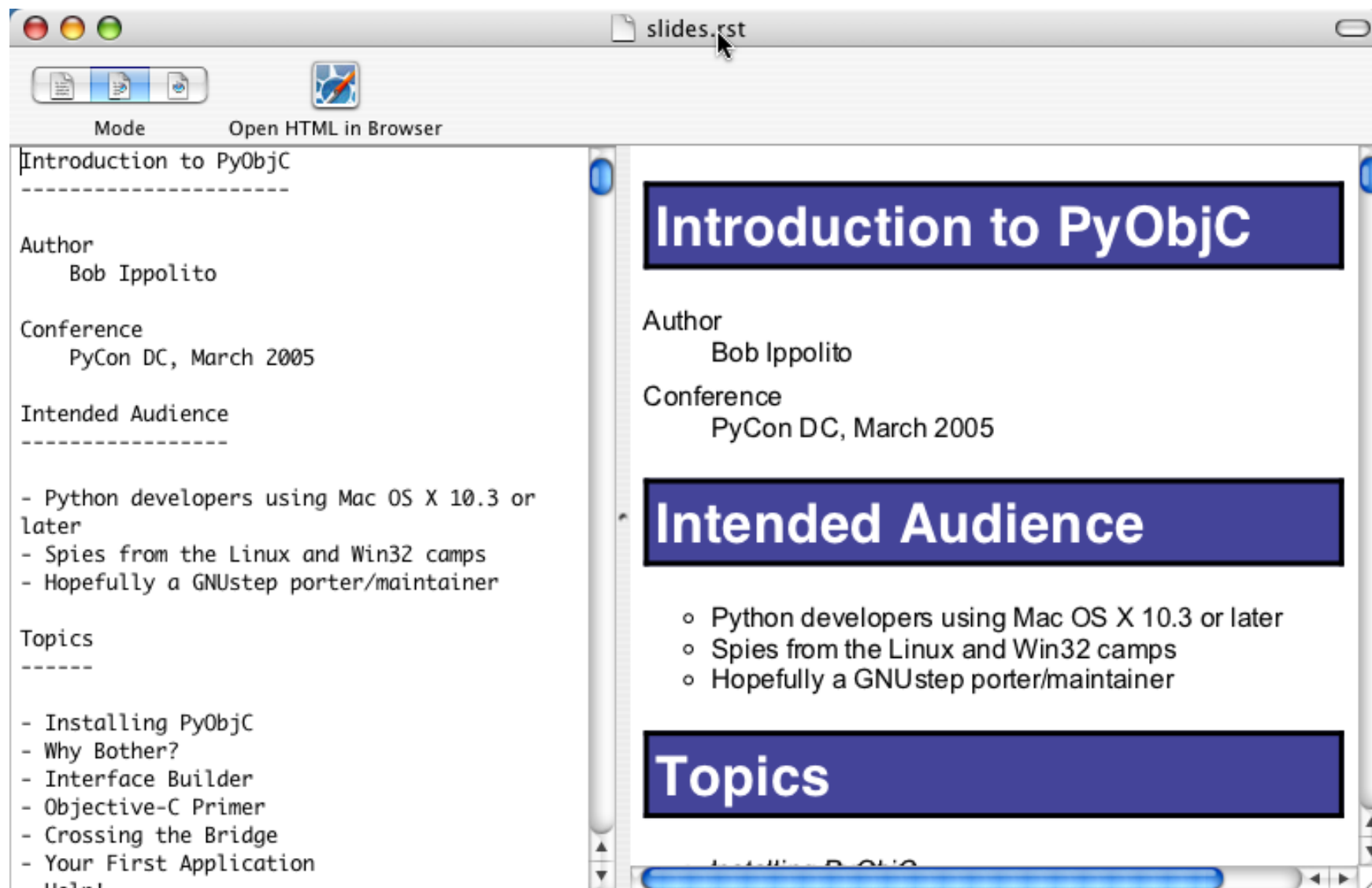
#pythonmac (on freenode)

Mailing List:

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



ReSTedit



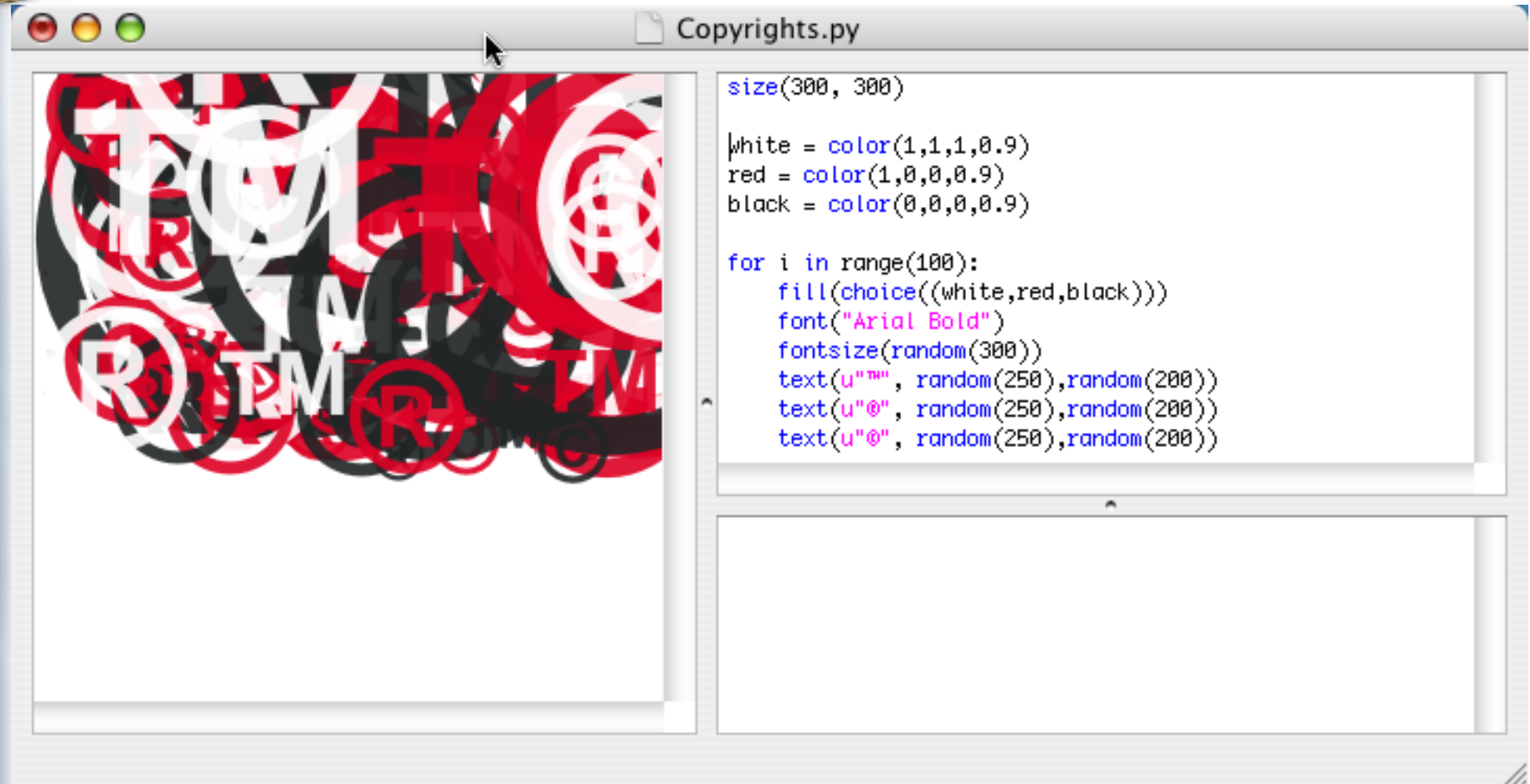


Flame

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 [0
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





Go ahead, ask.

Questions?