

PyObjC Cocoa Bindings Tutorial

Author

Bob Ippolito

Last Modified

April 4, 2004

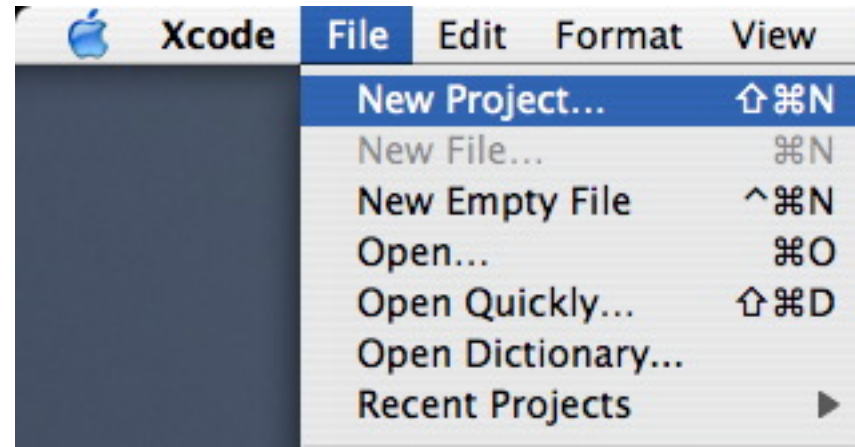
Introduction

This is a step-by-step tutorial of how to create a simple Cocoa Bindings enabled `/etc/passwd` viewer application with PyObjC.

To begin, you will need:

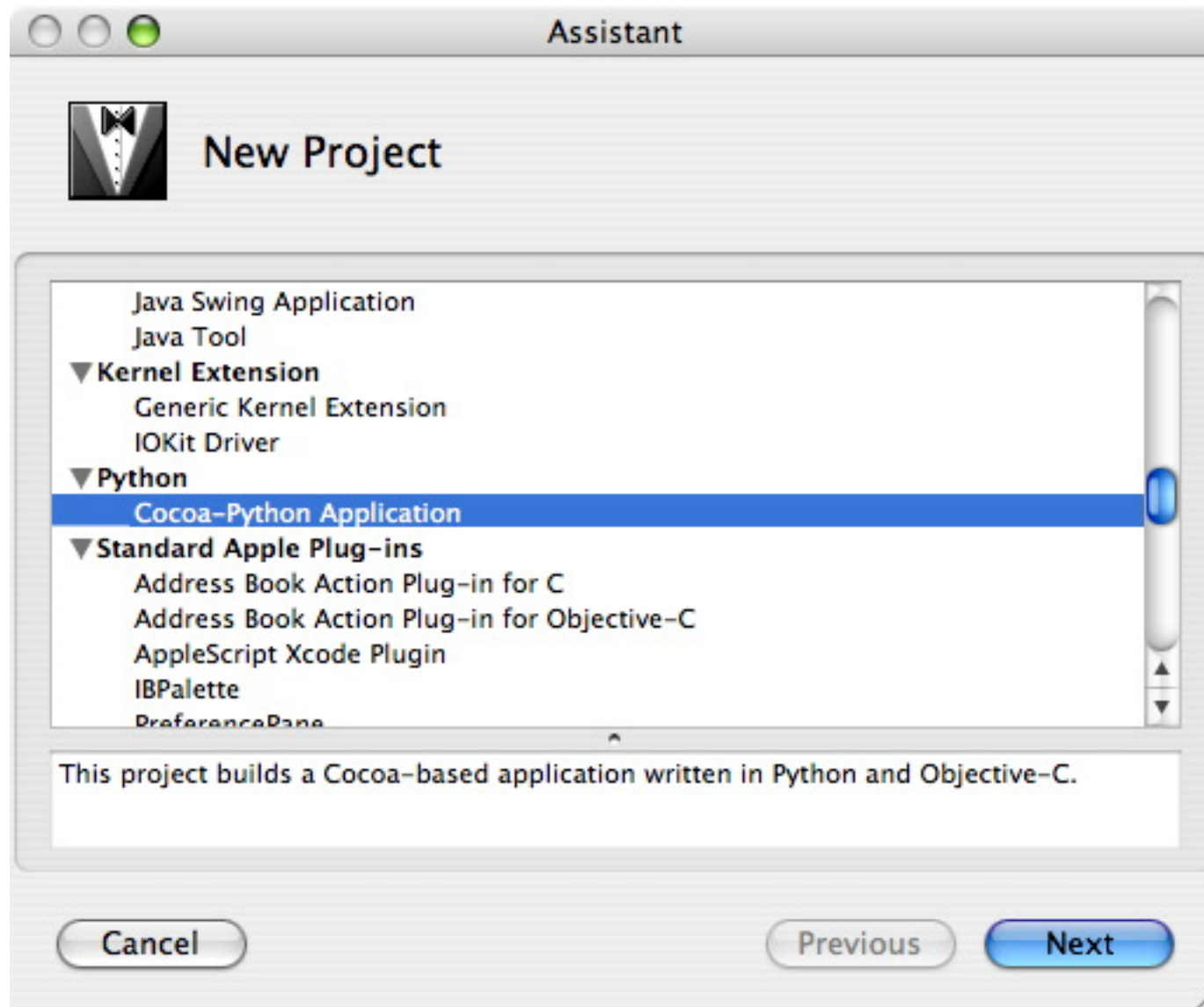
- Mac OS X 10.3 with Xcode installed
- PyObjC from CVS (or newer than 1.1b1)
- Moderate Python experience

Create the Project



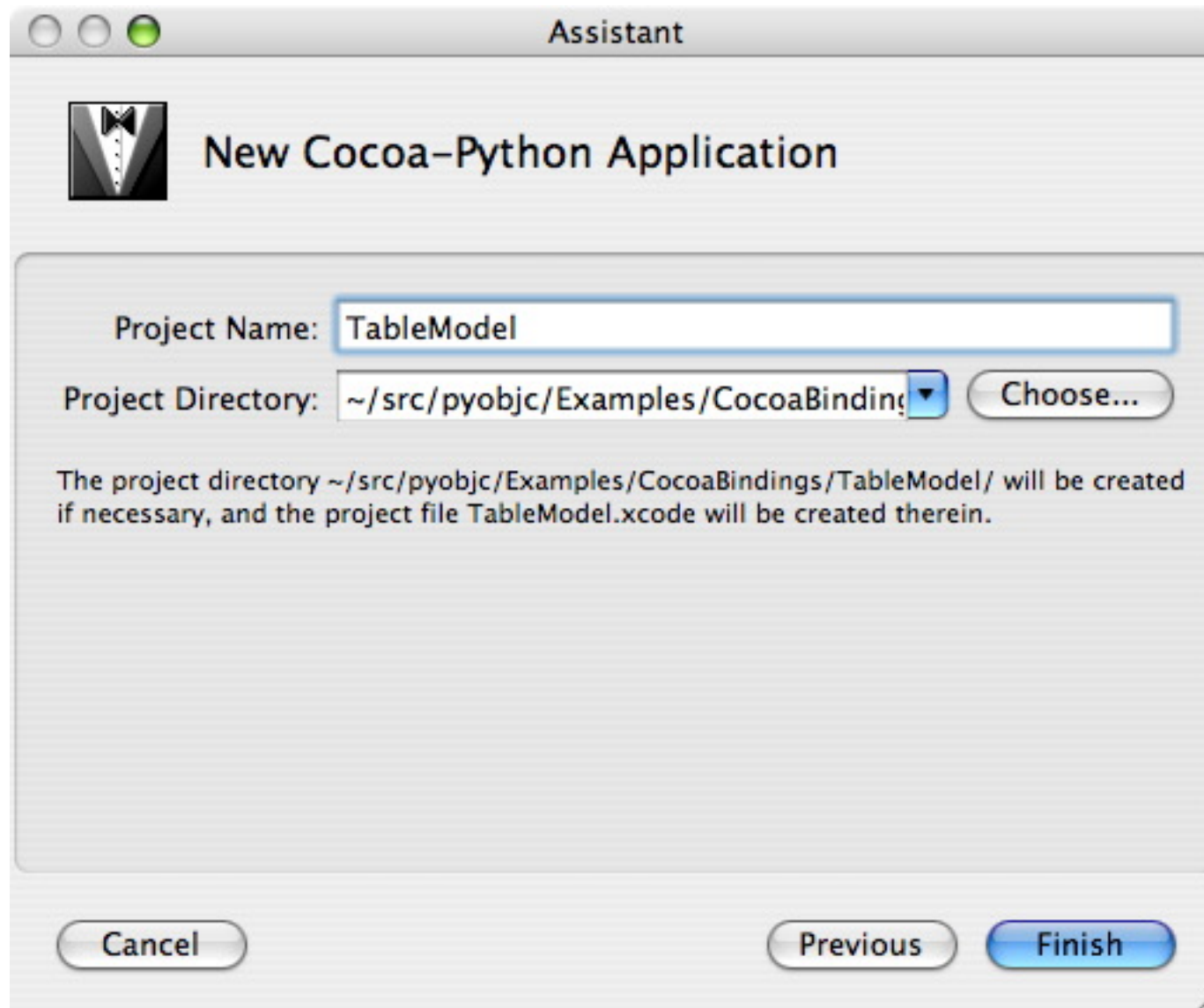
First you must open Xcode and create a New Project.

Cocoa-Python Application



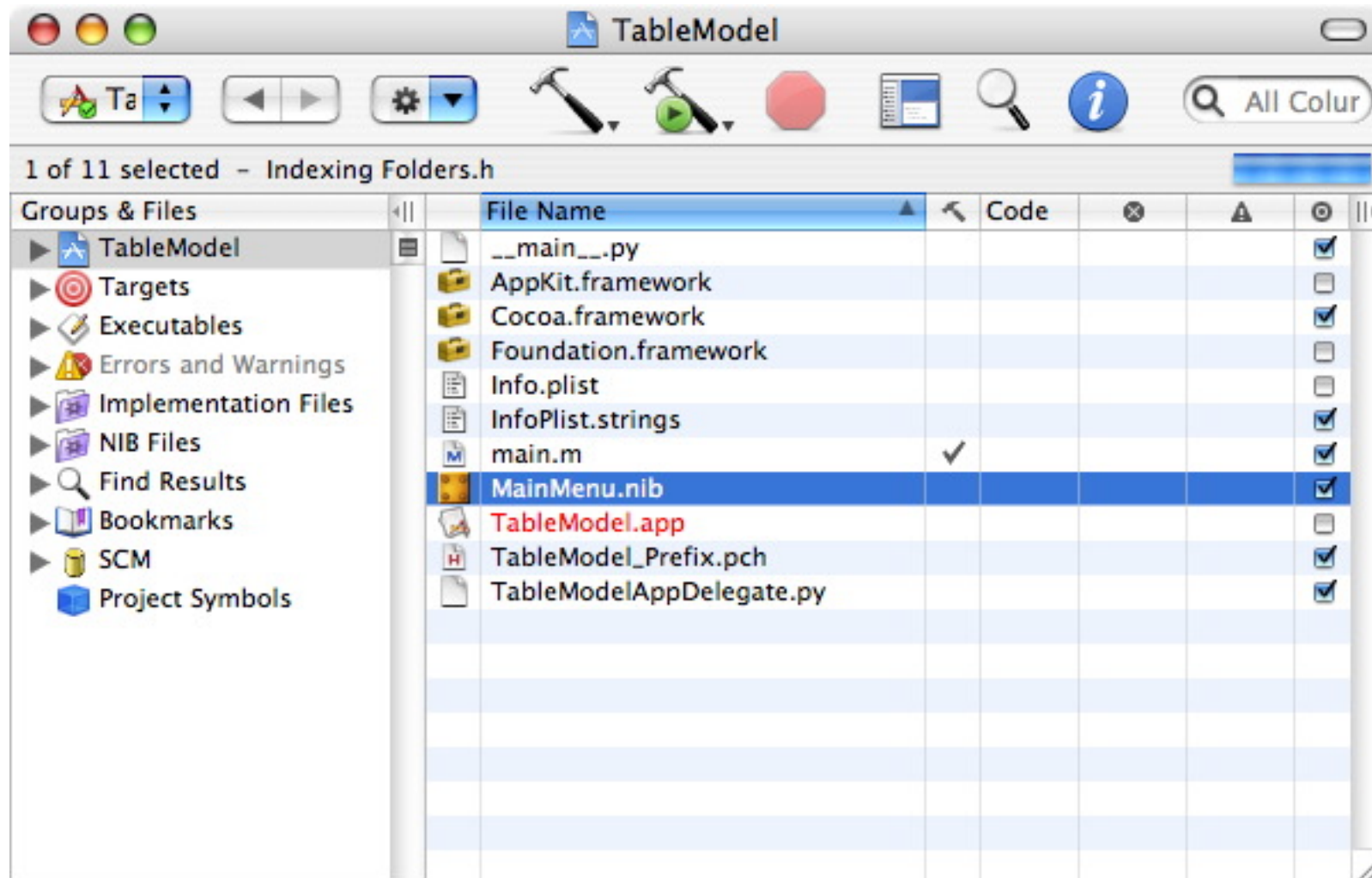
Choose the Cocoa-Python Application template from the Xcode Assistant.

Name Your Project



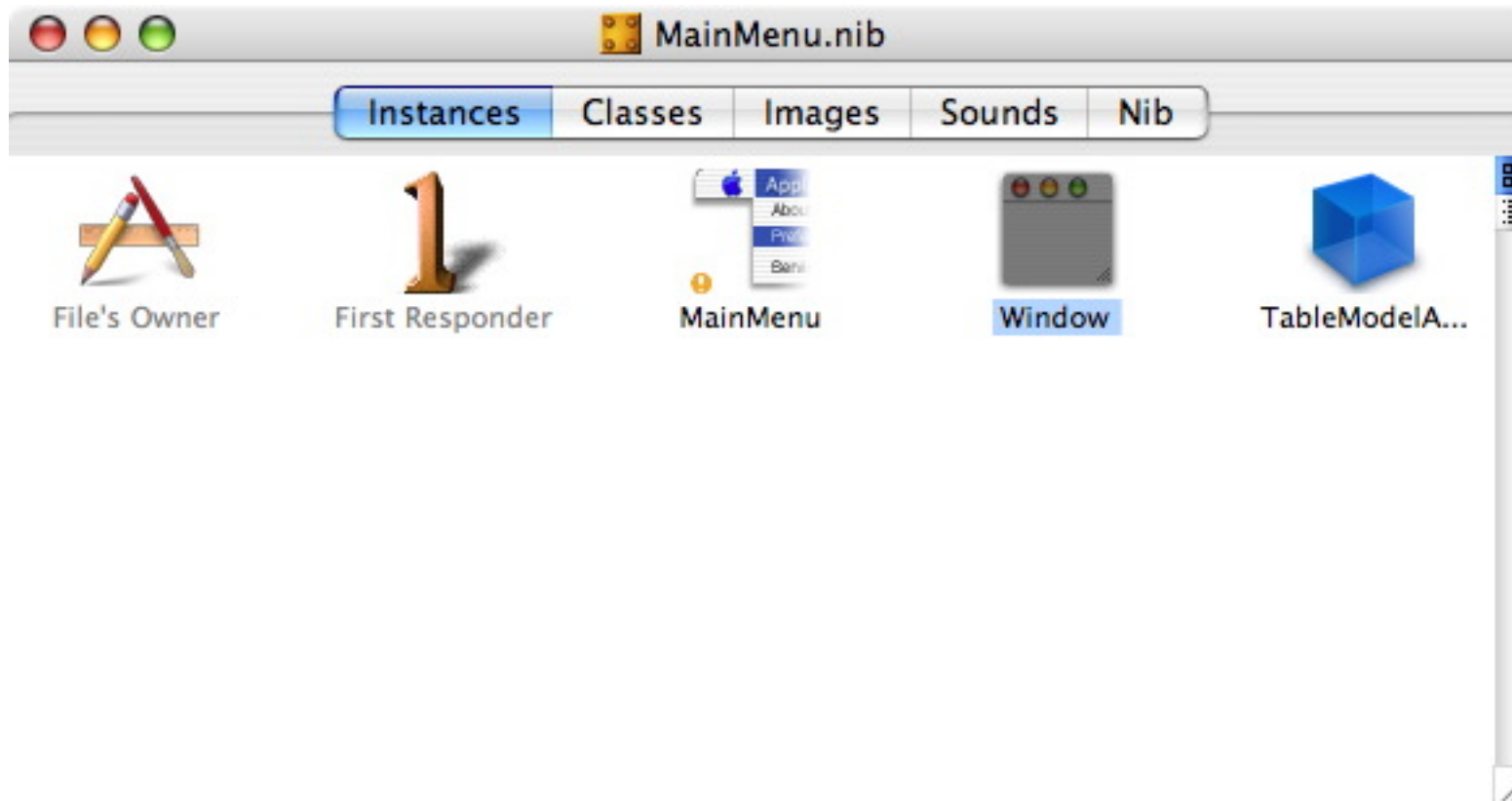
Choose a name and location for your PyObjC project.

Open the Nib File



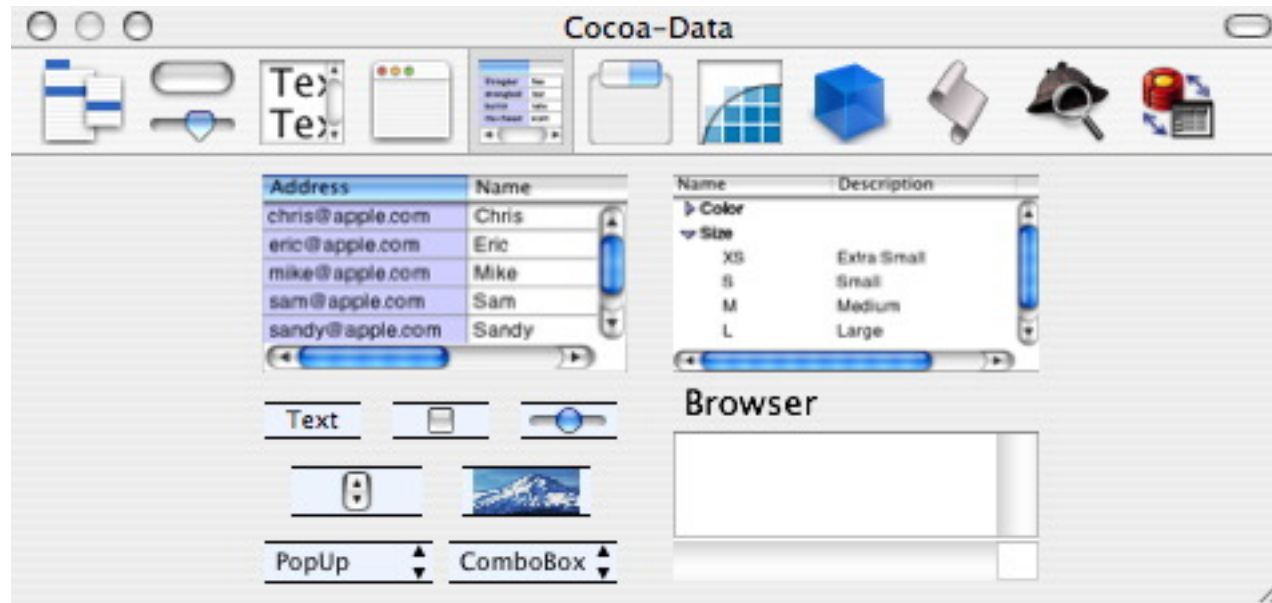
Open the MainMenu.nib file. This is the GUI of your application, and you will edit it with the Interface Builder application.

Open the Window



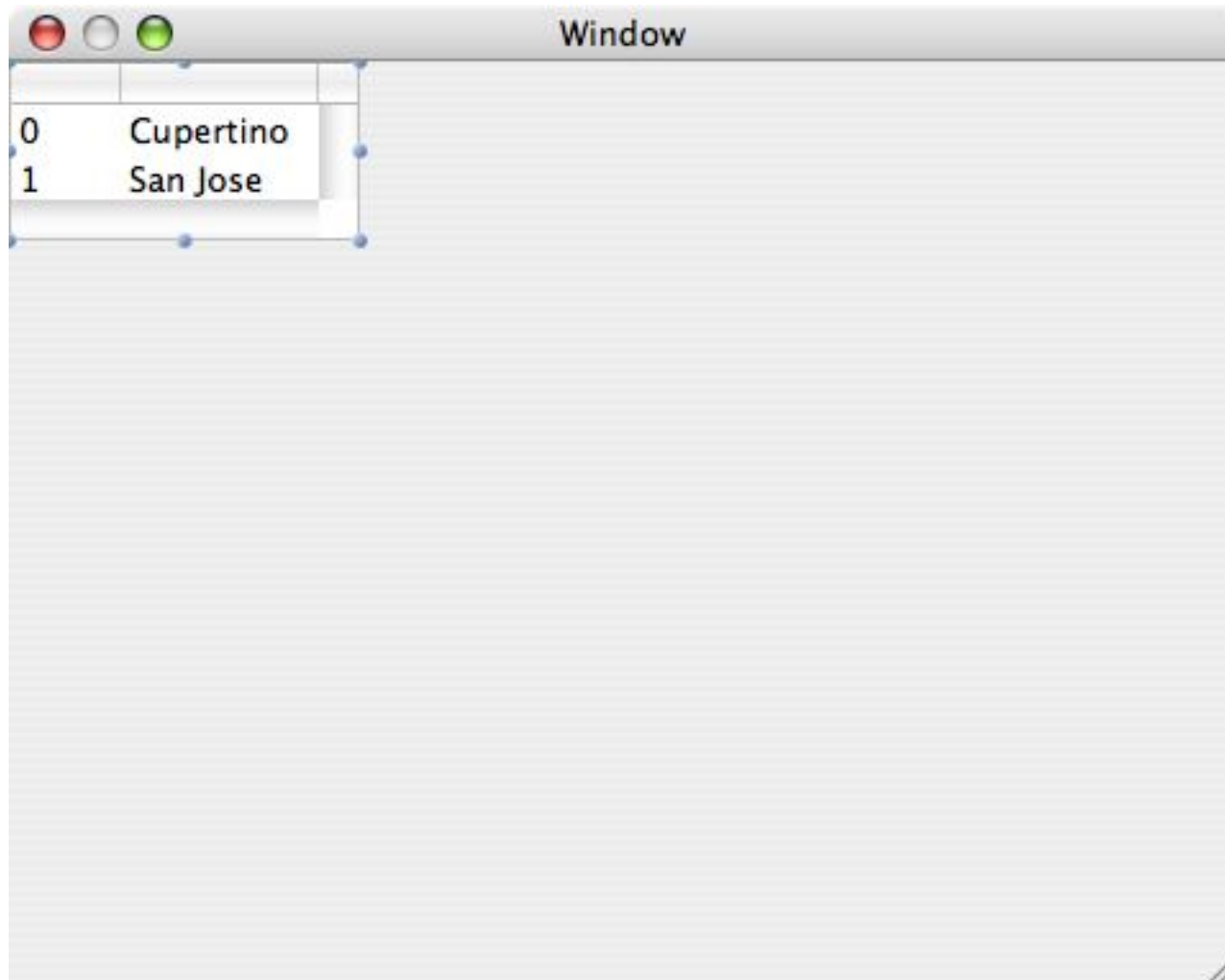
Double-click the Window object in the Instances tab of your nib document. This is the main window of your application, and will open immediately on launch.

Instantiate a NSTableView



From the Cocoa-Data section of the Palette (cmd-/ if not already open), instantiate a NSTableView by dragging it to your window. The NSTableView is the columnar data browsing widget in the upper-left hand corner of the Palette.

Adjust the Size of the NSTableView

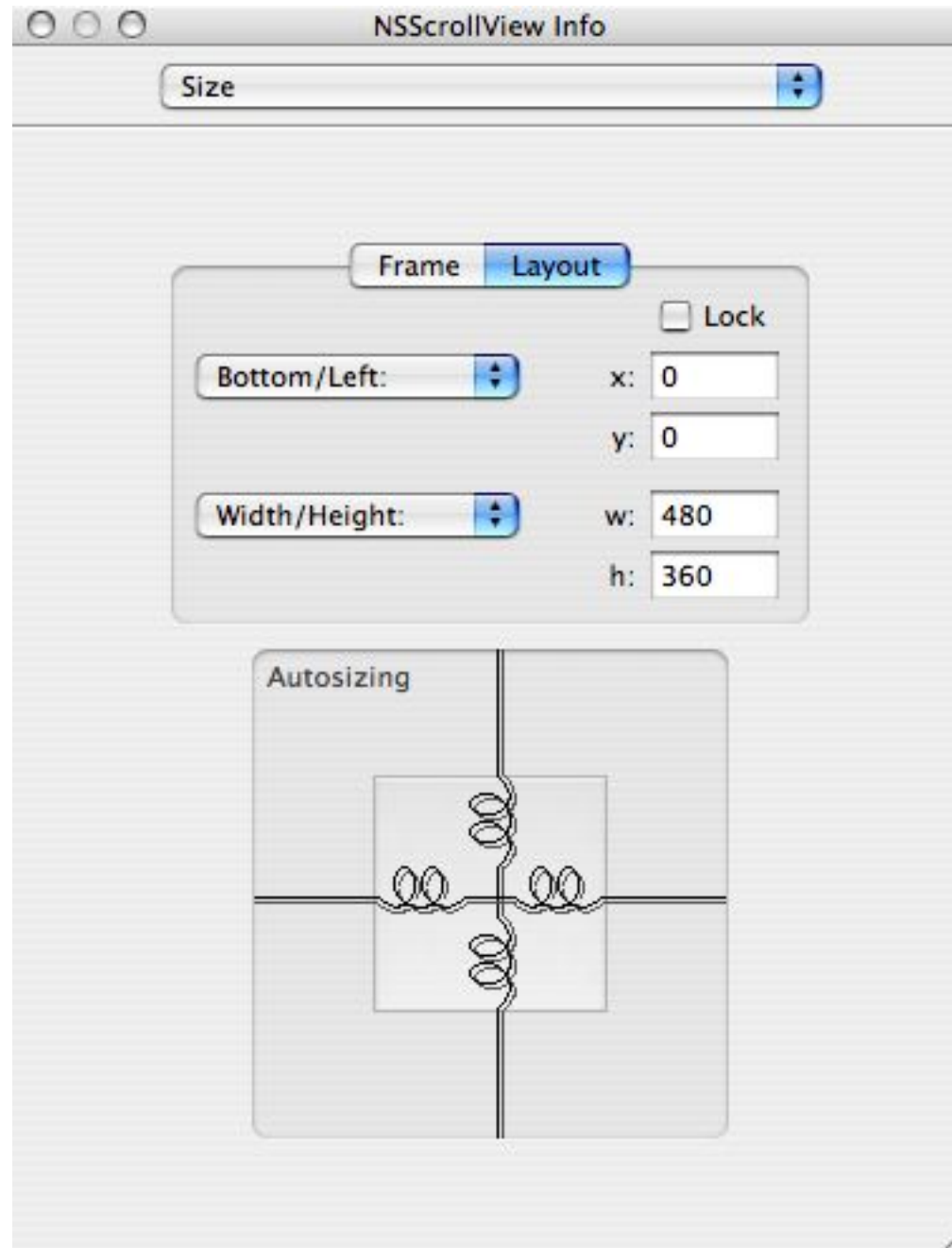


Select the newly instantiated NSTableView by clicking it from the main window. Drag the sizers so that the NSTableView instance fills up the full view of the window.

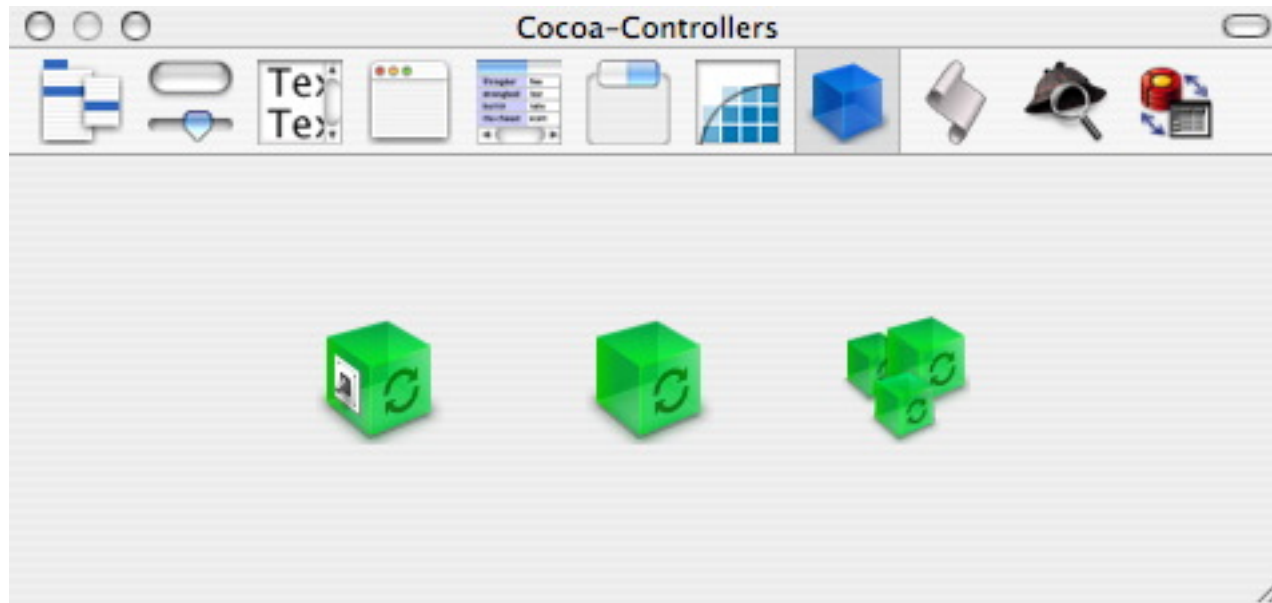
Change the Resizing Behavior

Select the NSTableView instance and open the Info panel (cmd-shift-I, or Tools > Show Info from the menu). Choose the Size section (cmd-3) and click on the horizontal and vertical bars inside the inner box. This will allow the NSTableView instance to grow and shrink automatically with the window.

Change the Resizing Behavior (Screenshot)



Create an NSArrayController

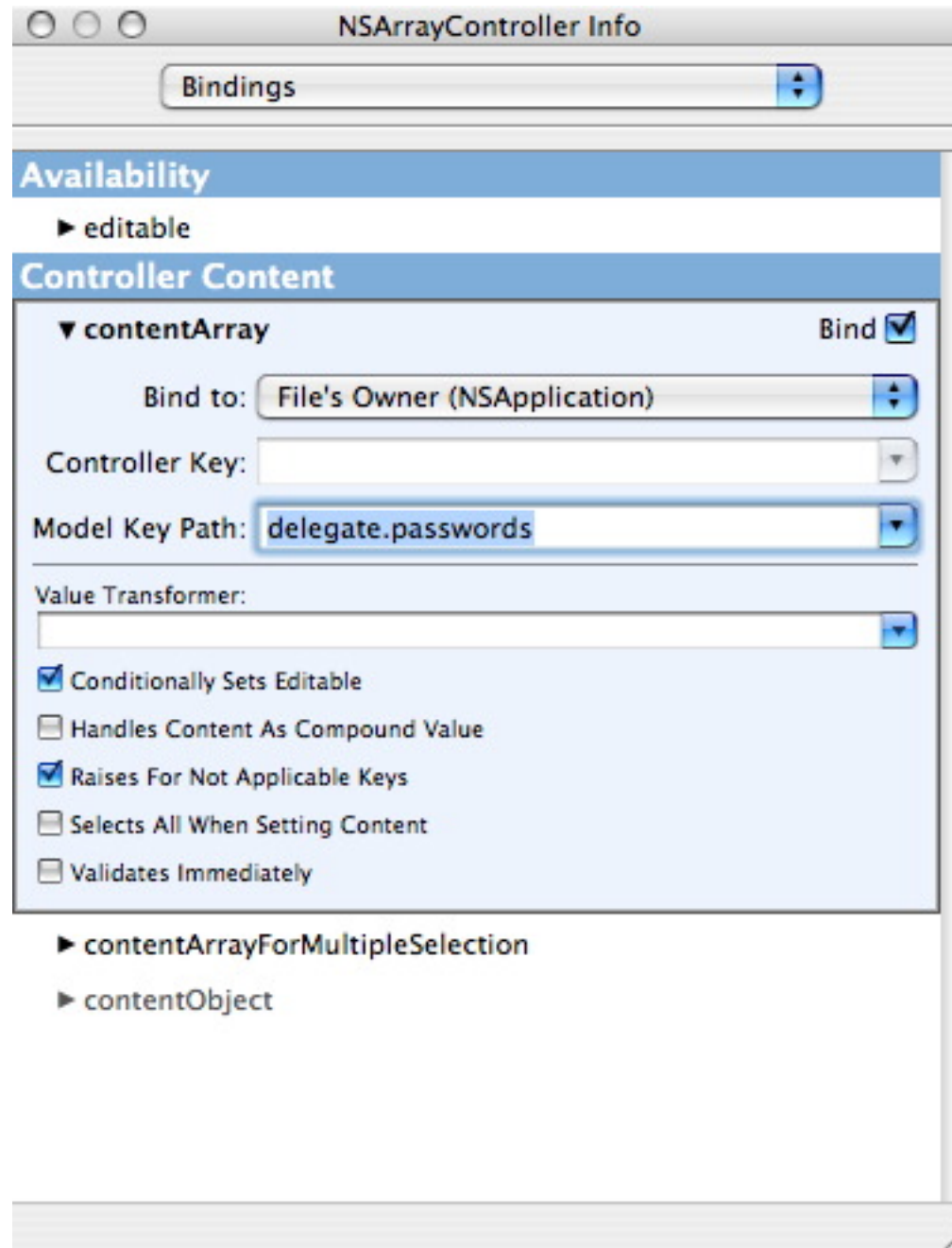


From the Cocoa-Controllers section of the Palette, drag an NSArrayController to your nib document (not the main window). The NSArrayController is the rightmost object in the palette, and is easily distinguished because it has multiple cubes.

Bind the NSArrayController

From your nib document, click on the newly instantiated NSArrayController and then open the Info panel. Choose the Bindings section (cmd-4), and click on contentArray. Bind to File's Owner (NSApplication) and use delegate.passwords as the Model Key Path.

Bind the NSArrayController (Screenshot)



Edit the Columns of Your Table



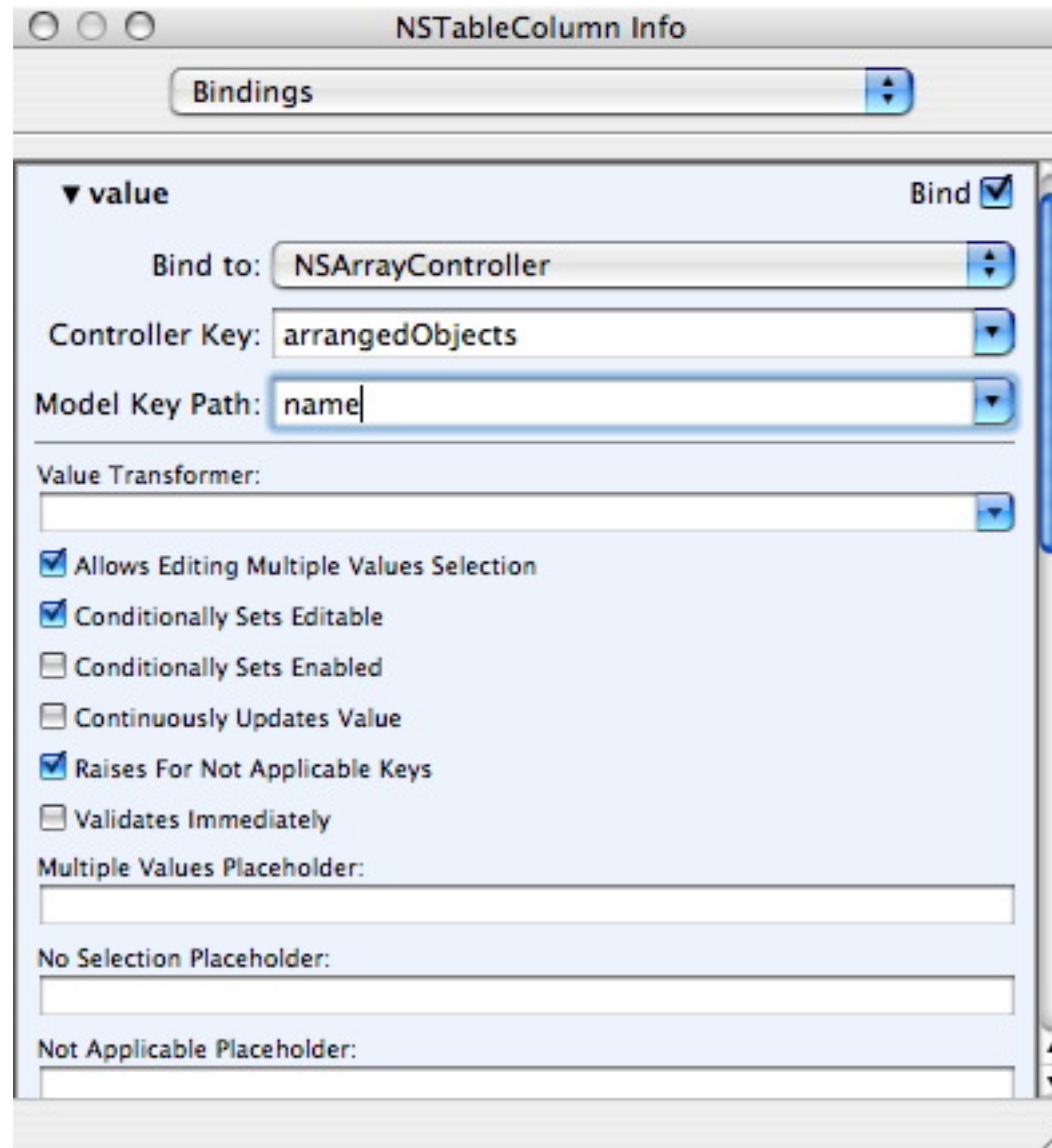
0	Cupertino
1	San Jose
2	Santa Clara
3	San Francisco
4	Palo Alto
5	San Carlos
6	Los Gatos
7	Sunnyvale
8	Mountain View
9	Redwood City

Select the NSTableView instance by double-clicking it, then click on one of the column headers to select it. Click on the header again, or go to the Info panel to change the displayed name of each column. For now, name one of the columns Short Name and the other UID.

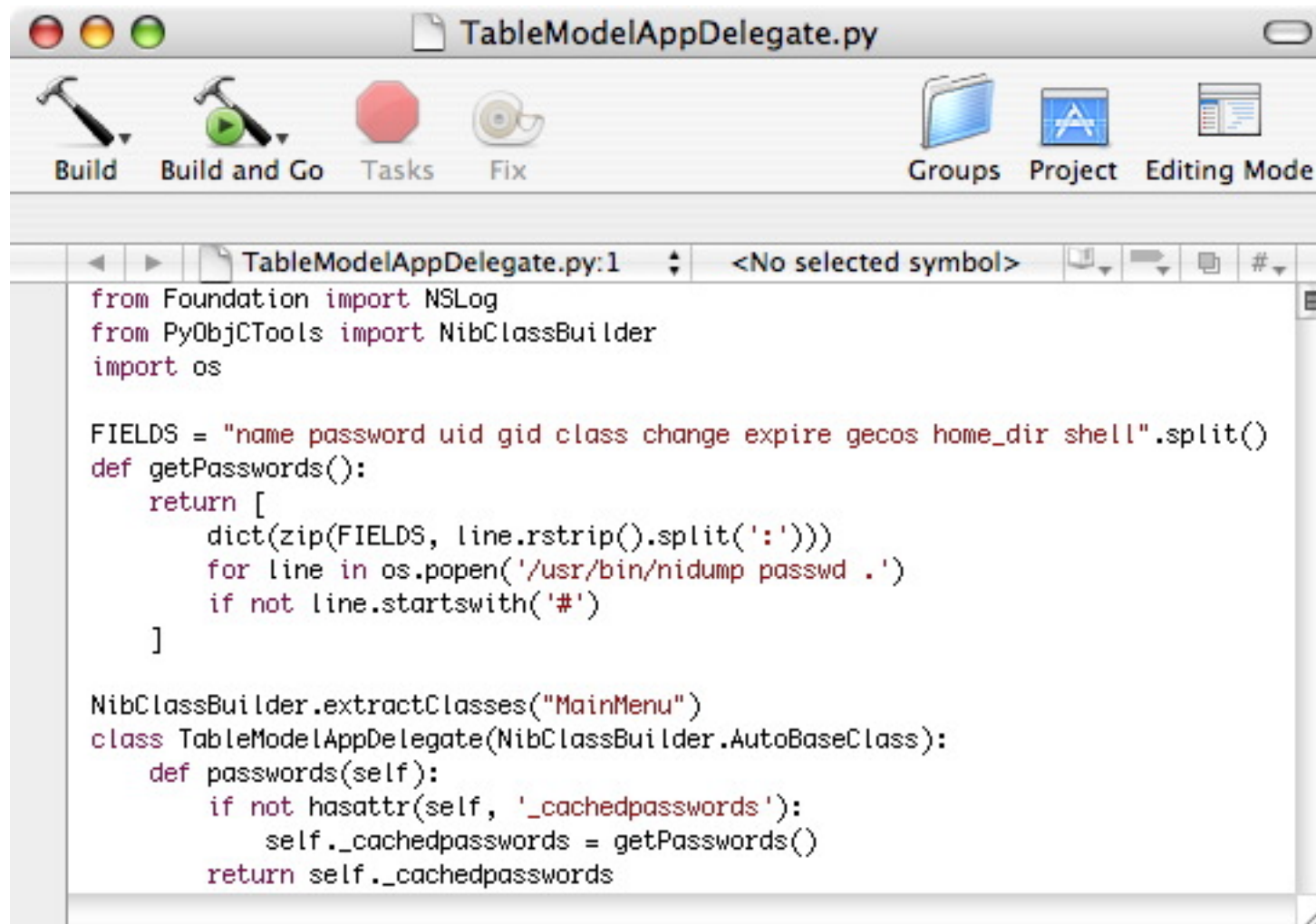
Bind Your Table Columns

Select the Short Name table column and open the Bindings section of the Info panel. Click on value and bind to the name Model Key Path of the NSArrayController. Do the same for the UID column, but use a Model Key Path of uid.

Bind Your Table Columns (Screenshot)



Write the Code



The screenshot shows the Xcode IDE with a Python file named 'TableModelAppDelegate.py' open. The code defines a class 'TableModelAppDelegate' that inherits from 'NibClassBuilder.AutoBaseClass'. It includes a method 'passwords()' that returns a list of dictionaries, each representing a password entry from the system's password file. The code uses 'os.popen()' to read the file and 'dict(zip())' to create the dictionaries.

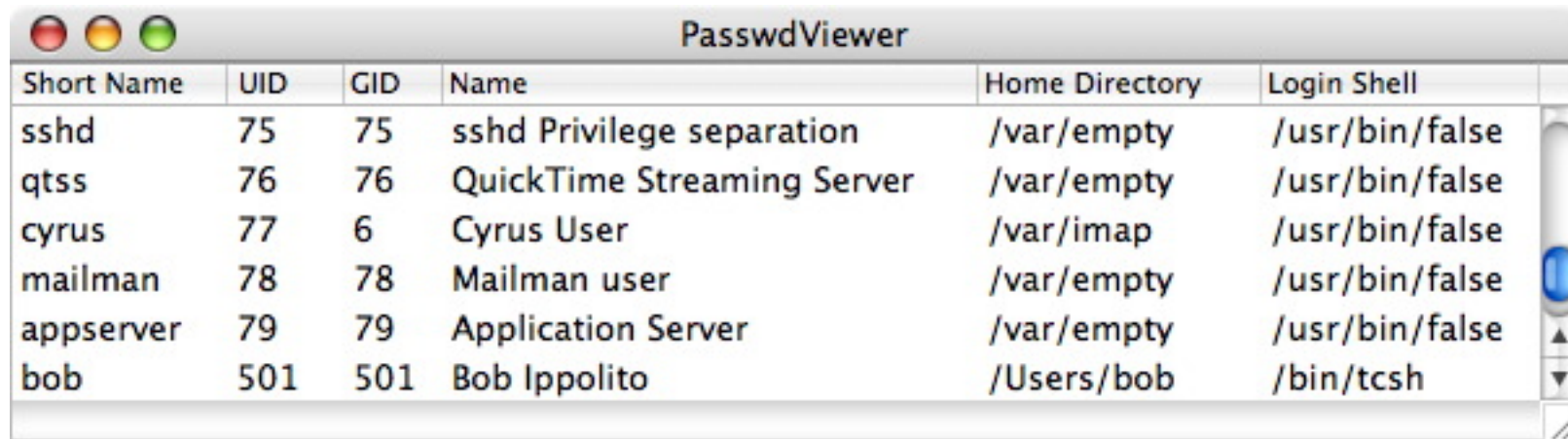
```
from Foundation import NSLog
from PyObjCTools import NibClassBuilder
import os

FIELDS = "name password uid gid class change expire gecos home_dir shell".split()
def getPasswords():
    return [
        dict(zip(FIELDS, line.rstrip().split(':')))
        for line in os.popen('/usr/bin/nidump passwd .')
        if not line.startswith('#')
    ]

NibClassBuilder.extractClasses("MainMenu")
class TableModelAppDelegate(NibClassBuilder.AutoBaseClass):
    def passwords(self):
        if not hasattr(self, '_cachedpasswords'):
            self._cachedpasswords = getPasswords()
        return self._cachedpasswords
```

Go back to Xcode and modify your application delegate class such that it has a method named `passwords` that returns a list of dicts. Each dict in the list must have the keys 'name' and 'uid', but may have other keys.

Run Your Application



The screenshot shows a macOS window titled "PasswdViewer" with a scrollable table of system users. The table has six columns: Short Name, UID, GID, Name, Home Directory, and Login Shell. The data is as follows:

Short Name	UID	GID	Name	Home Directory	Login Shell
sshd	75	75	sshd Privilege separation	/var/empty	/usr/bin/false
qtss	76	76	QuickTime Streaming Server	/var/empty	/usr/bin/false
cyrus	77	6	Cyrus User	/var/imap	/usr/bin/false
mailman	78	78	Mailman user	/var/empty	/usr/bin/false
appserver	79	79	Application Server	/var/empty	/usr/bin/false
bob	501	501	Bob Ippolito	/Users/bob	/bin/tcsh

You now have a complete minimal application that uses Cocoa Bindings. From Xcode, you can run it using cmd-R or by clicking the "Build and Go" button in the toolbar. For more Interface Builder practice, you should add more columns to the NSTableView and bind them to the other keys in your dictionaries.