Guide to DVD Chapter 29 Examples: *Andrew Beck*

Alternate Controllers: Connecting to HID Devices in Cocoa – HidFinder

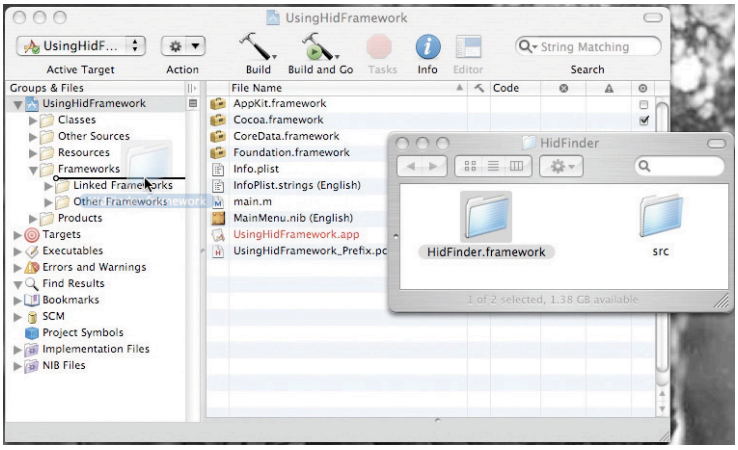
What is HidFinder?

HidFinder is a framework that easily integrates with Interface Builder to allow you to connect to USB or Bluetooth HID devices. This means most things from joysticks to mice to Arduino boards can be read by this program. It also allows you to load and save settings.

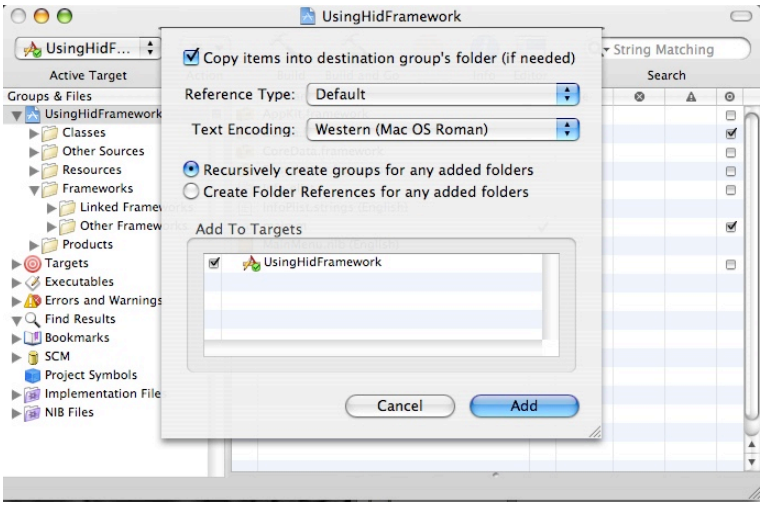
This is a quick and easy way to integrate alternate controllers into your application – all you need is one line of code and a few simple entries in Interface Builder.

Using *HidFinder.framework* in an Existing Program

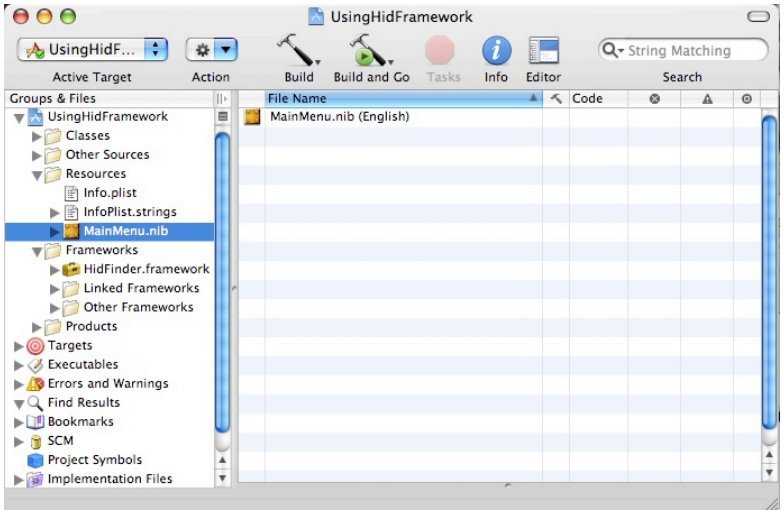
1. Drag the folder "HidFinder.framework" to your project, in the folder titled "Frameworks."



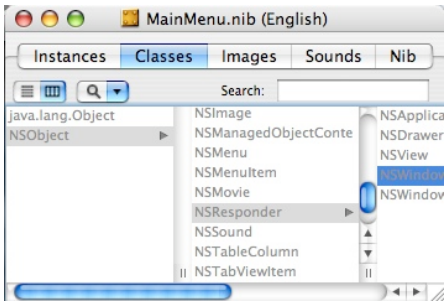
2. Select "Copy Files if needed" and press OK.



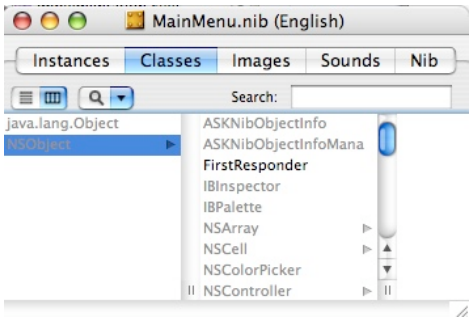
3. Open Interface Builder by clicking on “MainMenu.nib” located in “Resources” folder.



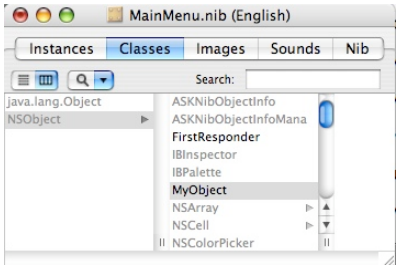
4. In the MainMenu.nib main window, select the “Classes” tab.



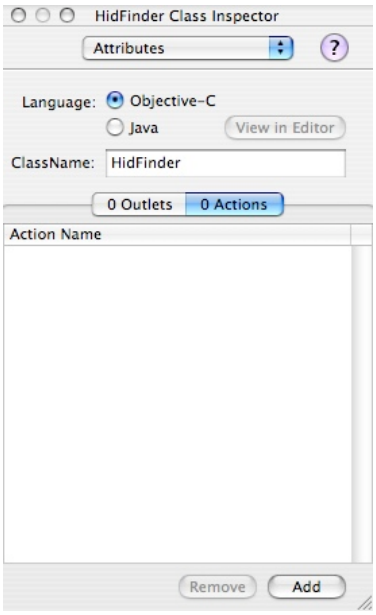
5. Select “NSObject” and press enter.



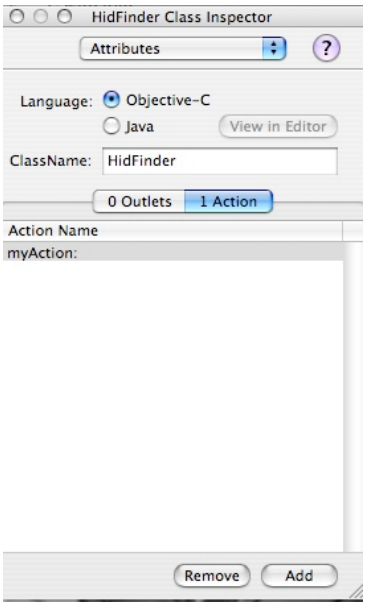
6. This will create a new class “MyObject”. Rename this “HidFinder”



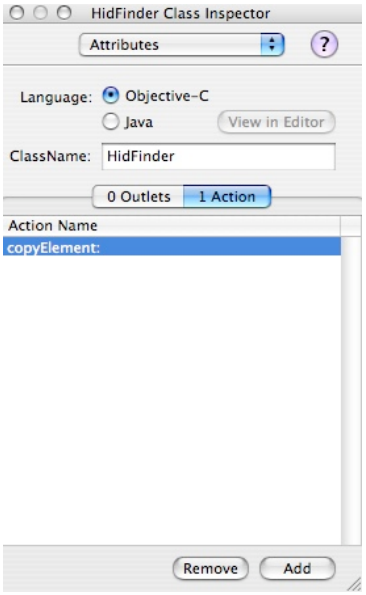
7. Press Command-1 or press Shift-Command-I and select “Attributes” to bring up the inspector window.



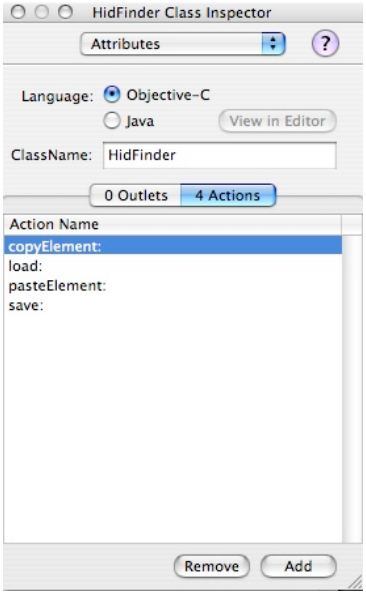
8. Under “actions”, click “add.” This will create a new action called “myAction:”



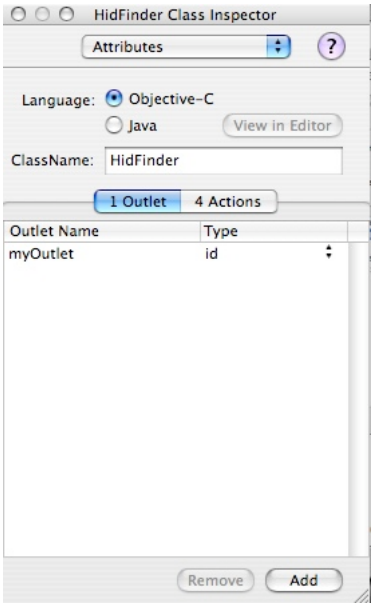
9. Rename this “copyElement:”.



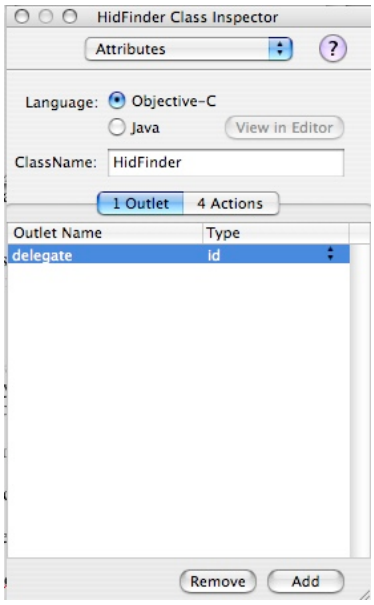
10. Add three more actions called “pasteElement”, “load” and “save”.



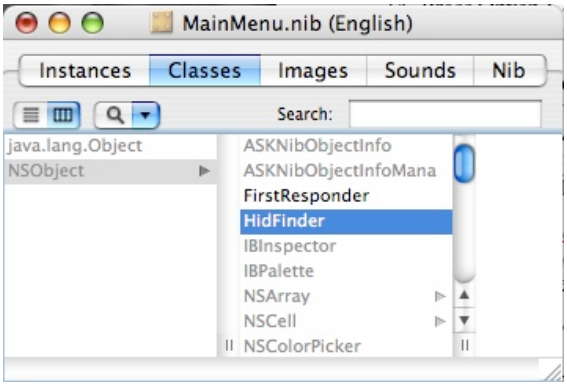
11. Under “outlets”, click “add”. This will create a new outlet called “myOutlet”.



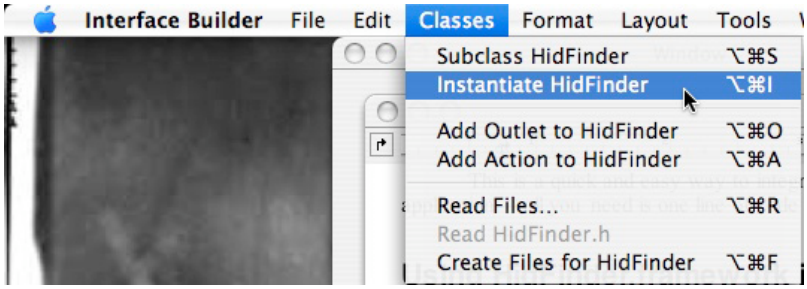
12. Rename this “delegate”.

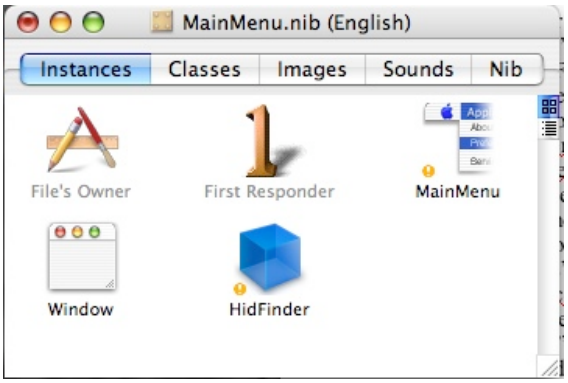


13. Return to the MainMenu and select “HidFinder”

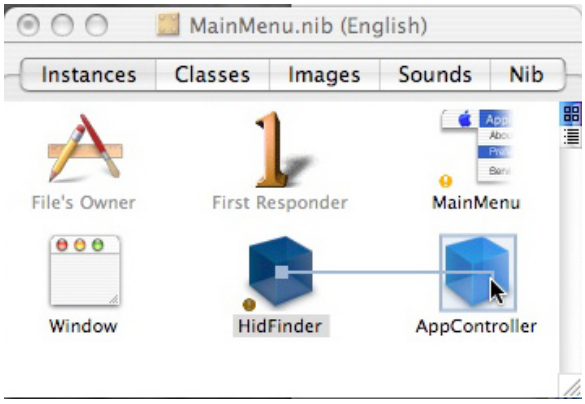


14. Press Option-Command-I or go to the menu “Classes” and select “Instantiate HidFinder”. This will bring you back to the “Instances” tab in the main window with a new object “HidFinder”.

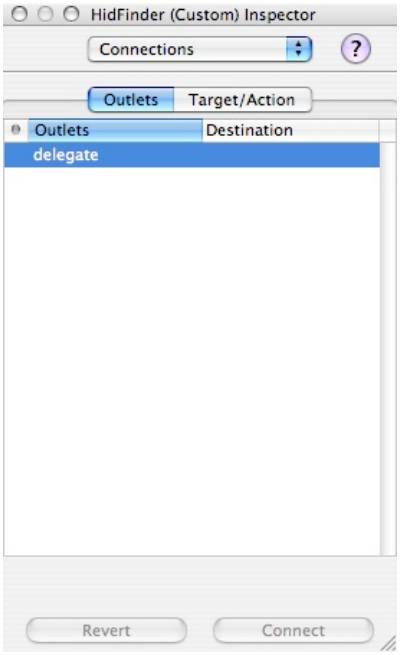




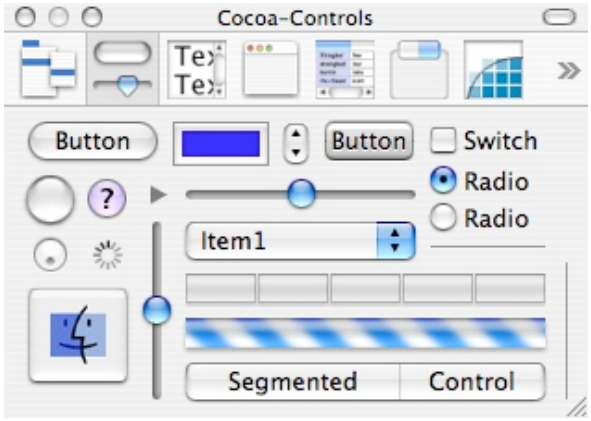
15. Control-drag from “HidFinder” to the class that you will be using in your application. In this image, it is called “AppController”.



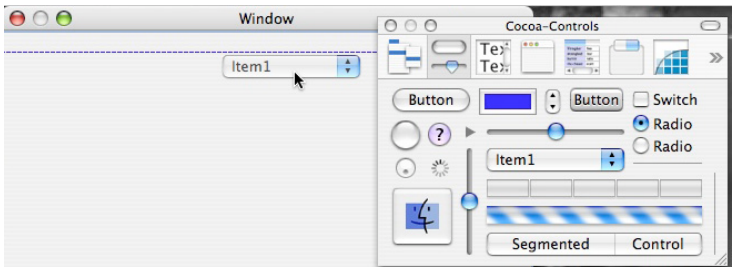
16. The inspector window will open – Select “delegate” and press “Connect”.



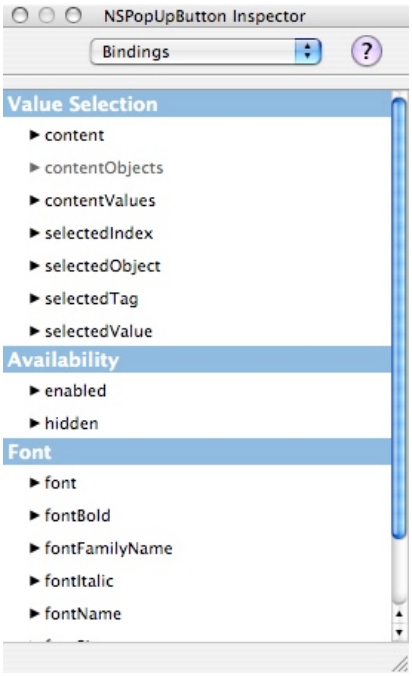
17. In your GUI window (the window your application show) you will need to create a few objects to use HidFinder. Press Command-/ to open the palettes window.



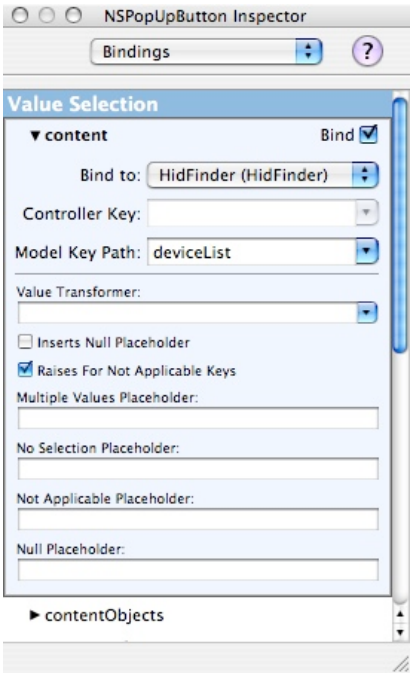
18. From the palettes window, drag an NSPopUpButton to your GUI window. Once we set it up, this will list the devices connected to your computer.



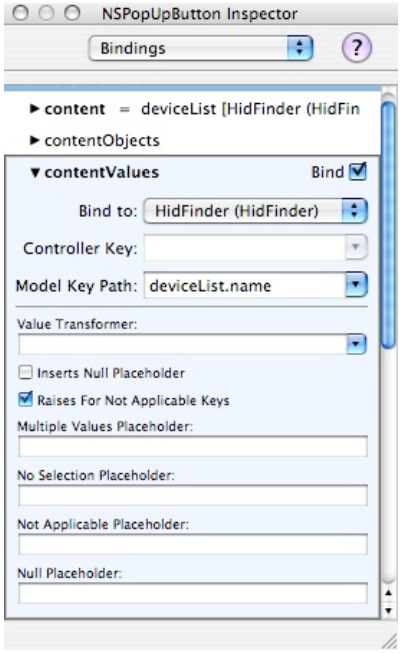
19. Select it and press Command-4. This will open up the inspector, this time in “Bindings”.



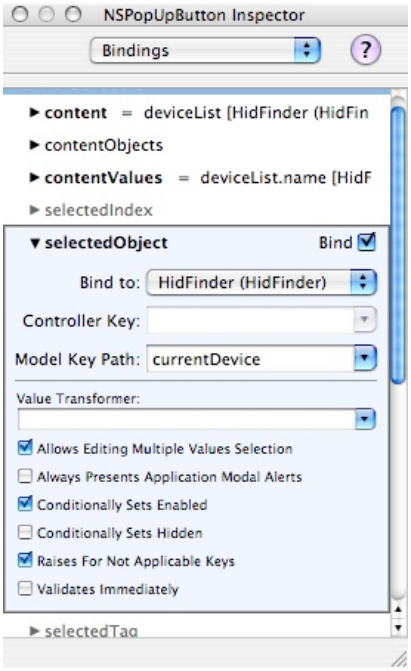
20. Select “content”. Bind to “HidFinder” and set the Model Key Path to “deviceList”.



21. Close “content” and open “contentValues”. Bind to “HidFinder” and set the Model Key Path to “deviceList.name”.

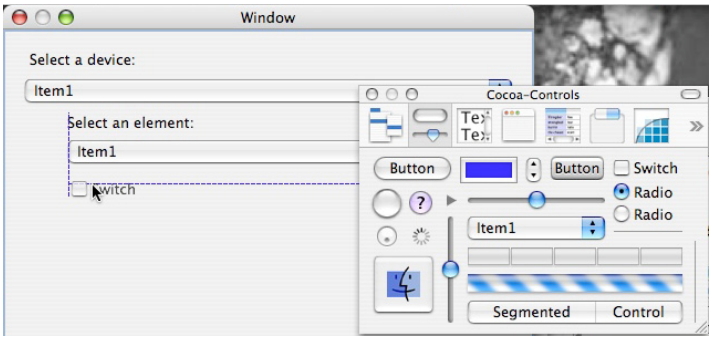


22. Close “contentValues” and select “selectedObject”. Bind to “HidFinder” and set the Model Key Path to “currentDevice”.

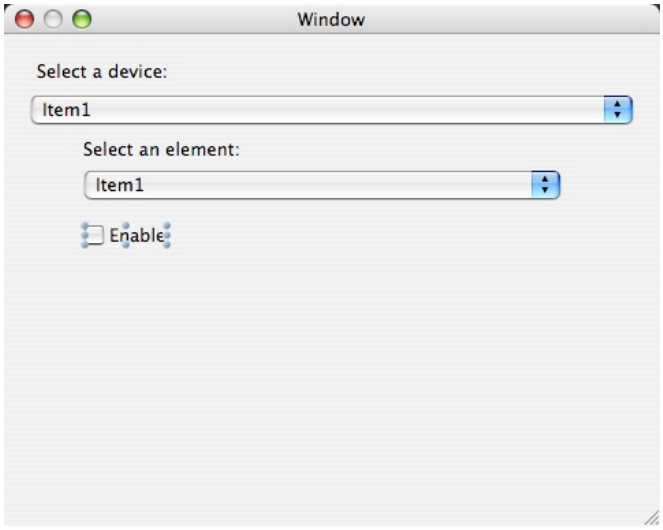


23. Drag another NSPopUpButton to your GUI window. This will list the elements within each device.
24. Select it and press Command-4.
25. Select “content”. Bind to “HidFinder” and set the Model Key Path to “currentDevice.elementList”.

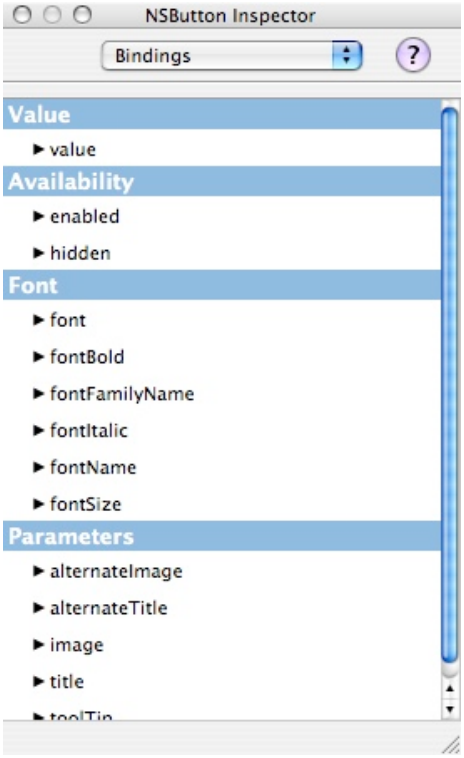
26. Close “content” and open “contentValues”. Bind to “HidFinder” and set the Model Key Path to “currentDevice.elementList.name”.
27. Close “contentValues” and open “selectedObject”. Bind to “HidFinder” and set the Model Key Path to “currentDevice.currentElement”.
28. From the palette window, drag the NSButton labeled “Switch” to your GUI. This will be the enable button for the selected element.



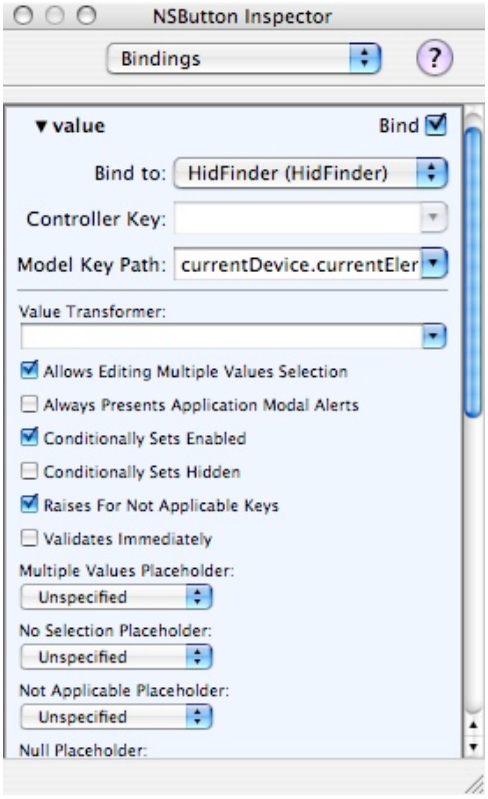
29. Rename the button “Enable”.



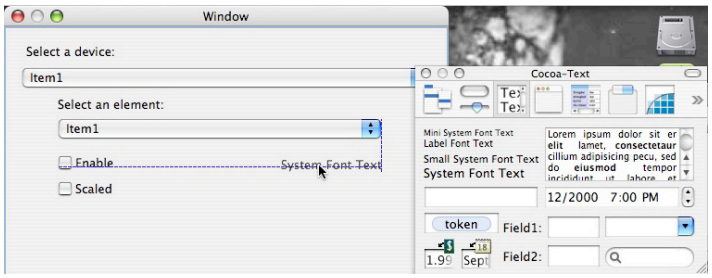
30. Select the button and press Command-4. This will open the inspector window under “Bindings”.



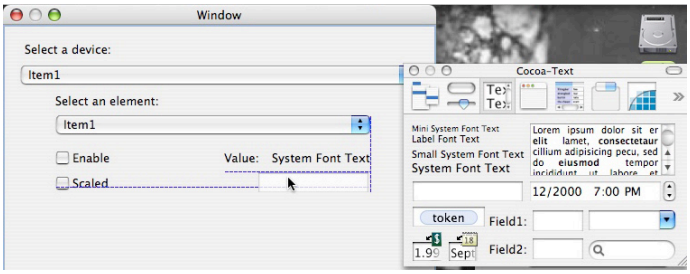
31. Select “value”. Bind to “HidFinder” and set the Model Key Path to “currentDevice.currentElement.isEnabled”.



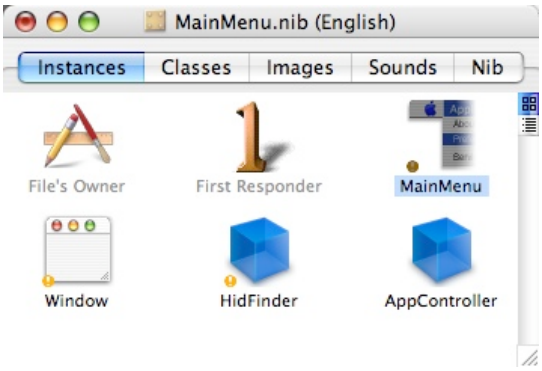
32. Drag another switch-type NSButton to your GUI and name it “Scaled”. This will scale the output to be between 0 and 1, which can be useful when working with joysticks.
33. Select the button and press Command-4.
34. Select “value”. Bind to “HidFinder” and set the Model Key Path to “currentDevice.currentElement.isScaled”.
35. Drag a non-editable NSTextField object from the palette window. This will display the current value of the element if it is enabled.



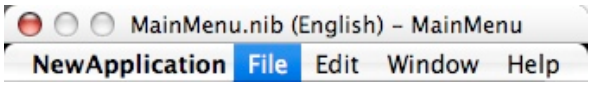
36. Select the NSTextField and press Command-4.
37. Select “value”. Bind to “HidFinder” and set the Model Key Path to “currentDevice.currentElement.value”.
38. Drag an editable NSTextField from the palettes window. This will let the user edit a string associated with the selected element.



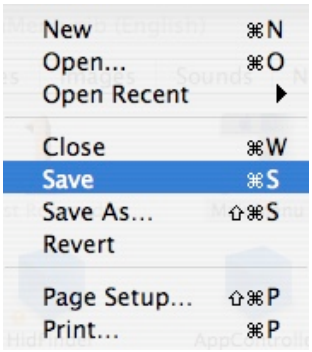
39. Select the NSTextField and press Command-4.
40. Select “value”. Bind to “HidFinder” and set the Model Key Path to “currentDevice.currentElement.string”.
41. In the main window, double click on “MainMenu”.



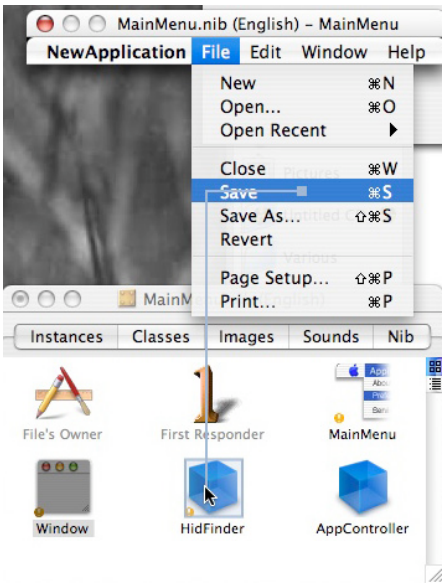
42. Select “File”.



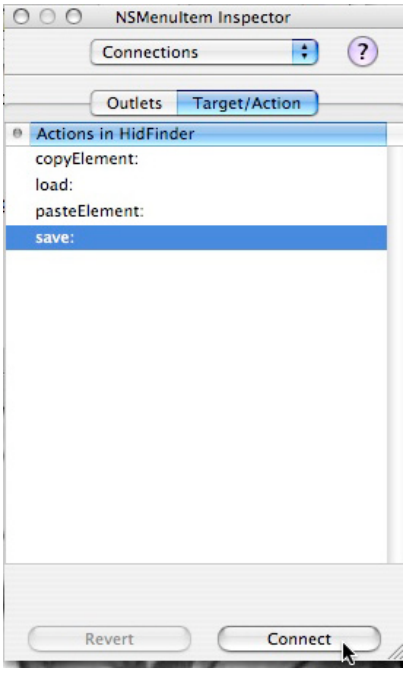
43. Select “Save”.



44. Hold Control and drag from “Save” to “HidFinder”.



45. Select “save:”



46. Repeat this process for “Open” and “Copy” and “Paste” in the “Edit” menu.
47. Save your changes to the nib file and exit Interface Builder.
48. In the class that you set as delegate in step 14, you will have to implement a single method called :

```
- (void) elementUpdated:(HidElement*) element;
```

This method will be called any time an enabled element has a new value. You can access the value by calling `[element value]`. You can access the string from the NSTextField by calling `[element string]`.

Look in the examples folder for sample code.