

Cocoa Programming with Ruby & RubyCocoa

Rod Schmidt

Who Is This Guy?

- Apple, NeXT fan since the beginning.
- Programming since age of 12.
- Always been interested in different programming languages.
- NeXTSTEP (1993), Mac OS X, Objective-C, Cocoa (2001)
- Started infiniteNIL in 2002
 - PhoneWord
 - first client in 2003 (StickyBrain)
 - PackRat in 2006

Cocoa

- Set of object-oriented frameworks for writing Mac OS X applications (Foundation, AppKit, etc.)
- Development Tools (free)
 - Xcode IDE
 - Interface Builder (GUI Designer)
- Languages: Objective-C, Java
- Delegation rather than inheritance
 - less code
- Proven
 - In use for over 10 years . Used in NeXTSTEP
- Portable

Objective-C

- Superset of C. Simple syntax (No ugly C++)
 - `[list insertObject: object atIndex: 0];`
- Similarities to Ruby
 - Smalltalk influence
 - Categories (Classes are always open)
 - Duck typing (if you want)
- Differences to Ruby
 - Basically C
 - No garbage collection. Reference counting.

RubyCocoa

- Ruby/Objective-C bridge
- Created by Hisakuni Fujimoto
- <http://rubycocoa.sourceforge.net/doc>
- Currently at version 0.4.2
- Open source. LGPL
- Uses Mac OS X pre-installed ruby (1.8.2)

Why RubyCocoa?

- Fun and productivity of a dynamic, interpreted language. (Faster development cycle. No compiling.)
- Garbage collection. No more reference counting!
- Automatic accessors: `attr_reader`, `attr_writer`, `attr_accessor`
- More conventional syntax (dot notation)
- Closer to Smalltalk than Objective-C (blocks, etc.)

Installing RubyCocoa

- <http://rubycocoa.sourceforge.net/doc>
 - click on Download then RubyCocoa-0.4.2-tiger.dmg (1/2 way down)
 - run RubyCocoa-0.4.2.pkg
- What gets installed
 - /Library/Frameworks/RubyCocoa.framework
 - RubyCocoa library (ruby files) - /usr/lib/ruby/site_ruby/1.8/osx
 - RubyCocoa extended library - /usr/lib/ruby/site_ruby/1.8/powerpc-darwin8.0/rubycocoa.bundle
 - Xcode templates - /Library/Application Support/Apple/Developer Tools
 - Documentation - /Developer/Documentation/RubyCocoa
 - Examples - /Developer/Examples/RubyCocoa

RubyCocoa Basics

- `require 'osx/cocoa'`
- `include OSX` or `OSX::`
- Creating instances - `Classname.alloc.init`
- Calling methods
 - Replace colons with underscore (leave off trailing underscore)
 - `[dict setObject: @"ruby" forKey: @"language"]`
 - `dict.setObject_forKey("ruby", "language")`
 - `dict.setObject("ruby", :forKey, "language")`
 - If Objective-C method returns BOOL use '?' suffix
 - `s1.isEqualToString?(s2)`

Type Conversion

- Calling an Objective-C method returns an Objective-C instance
 - Use `to_s`, `to_a`, `to_range`, etc. to convert to Ruby instances
- Parameters are automatically converted:
 - Ruby strings => NSString
 - Ruby ranges => NSRange
 - Ruby arrays => NSArray
 - also NSRect and NSPoint
- Special cases (NSRect, NSPoint, NSRange)
 - defined as Ruby classes
 - allocate with `new`

Creating Cocoa Subclasses

- Inherit from Cocoa class
 - `class RubyDocument < OSX::NSDocument`
 - For a custom class:

```
OSX.ns_import('BaseClass')  
  
class RubyClass < OSX::BaseClass
```
- Overriding methods
 - `ns_overrides :documentNibName`
 - `ib_overrides 'mouseUp:', :drawRect_`
- Calling superclass implementation - prepend `super_`
 - `super_documentNibName(sender)`

Subclasses (cont.)

- Initialization
 - Normally do initialization in Obj-C style ‘init...’ method
 - If you do define initialize only call Ruby methods
 - Calling an objective-C method would be bad because the instance hasn’t been initialized yet.
 - Return self. All Obj-C ‘init...’ methods must return self.
- Bindings
 - `kvc_accessor :dollarsToConvert, :exchangeRate`
 - `kvc_depends_on([:dollarsToConvert, :exchangeRate], :amountInOtherCurrency)`

Misc.

- Functions, Types, and Constants
 - defined in Ruby, just prefix with `OSX::`, or include
- Method Name Conflicts
 - use `OC_` prefix to denote Obj-C method.
- Exceptions
 - Cocoa exceptions translated into Ruby exceptions
 - `OSX::OCException` wraps original `NSException`

Ruby and Xcode

- Xcode understands Ruby and already syntax colors Ruby source code.
- Even provides drop-down list for navigation
- You can specify an external editor for Ruby source
 - like TextMate
- Also add scripts to Xcode's User Scripts menu.
 - “Using Scripts To Customize Xcode” in the Xcode docs.
 - Example: validate syntax command.

Ruby and Interface Builder

- Outlets
 - in Ruby code
 - `ib_outlets :textView`
 - `ns_outlets :messageField, :nameField`
 - In Interface Builder just create in inspector
- Actions
 - Ruby: define the method with the right signature
 - `def doAction(sender)`
 - InterfaceBuilder: create in inspector

Creating a RubyCocoa Application with Xcode

- File → New Project
 - Cocoa-Ruby Application
 - Cocoa-Ruby Core Data Application
 - Cocoa-Ruby Core Data Document-based Application
 - Cocoa-Ruby Document-based Application
- Demo
 - Cocoa app to display Today's events from iCal
 - Uses `iCalReader` - Ruby module for reading iCal files
 - Download it from the infiniteNIL website (developers section)
 - Read the O'Reilly article (Read iCal Data with Ruby)

Debugging

- Logging
 - puts
 - log4r - log4j for Ruby
- irb - Interactive Ruby Shell
- `App.app/Contents/MacOS/App -r debug`
 - Launches rdb debugger with your app
 - Similar to gdb

Packaging Your App

- Just like any other Cocoa app - Installer or App bundle
- Application Bundle
 - Embed RubyCocoa.framework in app bundle
 - Add framework to project (Project → Add Frameworks)
 - Add Copy Files Build Phase (Project → New Copy Files Build Phase)
 - Copy Files section of Target Settings
 - Where: Frameworks
 - Drag RubyCocoa.framework to new Copy Files section

Gotchas

- Conversion from Cocoa types to Ruby types
 - Make sure you are comparing Ruby to Ruby or Cocoa to Cocoa
 - use `to_s` or `isEqualToString`
 - `NSString stringWithFormat:@"%d", 3).to_s == '3'`
 - Also arrays, ranges, points, etc.
- Constants: Use method call if not found
 - `NSDragPboard()`
- Don't forget your `ib_overrides`
- Only 0.4.2, so there might still be some bugs

Wrap Up

- Links
 - RubyCocoa: <http://rubycocoa.sourceforge.net>
 - downloads, documentation, mailing lists, etc
 - Cocoa: <http://developer.apple.com/cocoa>,
 - <http://www.macdevcenter.com>
- Cocoa Programming for Mac OS X (2nd Ed.) by Aaron Hillegass
- Contact Info
 - www.infinitenil.com
 - rod@infinitenil.com
- Q & A