



MacRuby (the bloodthirsty)

Mario Aquino
<http://marioaquino.blogspot.com>

MacRuby vantz ta drink yer blud





JRuby



Java™



MacRuby

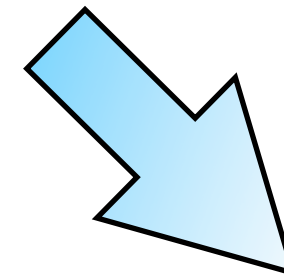
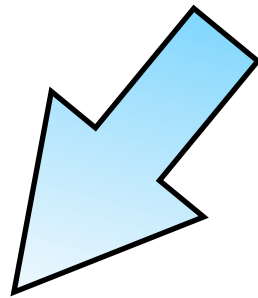


OBJECTIVE-C



IronRuby

Microsoft®
.net™



	Ruby	Objective-C	Smalltalk-80
Typing	dynamic	dynamic/static	dynamic
Runtime access to method names	yes	yes	yes
Runtime access to class names	yes	yes	yes
Runtime access to instance variable names	yes	yes	yes
Forwarding	yes	yes	yes
Metaclasses	yes	yes	yes
Inheritance	mix-in	single	single
Access to super method	super	super	super
root class	Object	Object (can have multiple)	Object (can have multiple)
Receiver name	self	self	self
Private Data	yes	yes	yes
Private methods	yes	no	no
Class Variables	yes	no	yes
Garbage Collection	yes	yes	yes

From http://www.approximity.com/ruby/Comparison_rb_st_m_java.html

Syntactic Comparison



obj.method parameter

[obj method:parameter]

items = []

```
NSMutableArray *items =  
[[NSMutableArray alloc] init];
```

‘Heynow’

@”Heynow”

Objective-C: Categories

```
#import <objc/Object.h>

@interface Integer : Object
{
    int integer;
}

- (int) integer;
- (id) integer: (int) _integer;
@end
```

```
#import "Integer.h"

@implementation Integer
- (int) integer
{
    return integer;
}

- (id) integer: (int) _integer
{
    integer = _integer;
    return self;
}
@end
```

```
#import "Integer.h"

@interface Integer (Arithmetic)
- (id) add: (Integer *) addend;
@end
```

```
#import "Arithmetic.h"

@implementation Integer (Arithmetic)
- (id) add: (Integer *) addend
{
    return [self integer: [self integer] + [addend integer]];
}
```

Ruby: Open Classes

```
class Integer
  attr_reader :integer

  def integer=(_integer)
    @integer = _integer
    self
  end
end
```



```
class Integer
  def add(addend)
    integer = integer + addend.integer
  end
end
```


Informal Protocols vs. method_missing



```
@interface NSObject ( MyXMLSupport )  
- initWithXMLRepresentation:(NSXMLElement *)XMLElement;  
@property (nonatomic, readonly) (NSXMLElement *)XMLRepresentation;  
@end
```

```
class MyXMLSupport  
  def method_missing(name, *args, &block)  
    # Handle stuff  
  end  
end
```



Ruby 1.9 Syntax Variance

```
# Taken from: http://gist.github.com/116876
class Thing
  def foo(bar, baz:raz)
    puts "#{bar}, #{raz}"
  end
  def foo(bar, hash)
    puts "I like #{hash}"
  end
end

t = Thing.new
t.foo("cat", baz:"monkey")
#=> cat, monkey
t.foo("cat", raz:"turtles")
#=> I like {:raz=>"turtles"}
```

```
mario:(git)vending[macruby]/doc$ ruby1.9 evil_macruby.rb
```

```
evil_macruby.rb:3: syntax error, unexpected tLABEL
```

```
  def foo(bar, baz:raz)
```

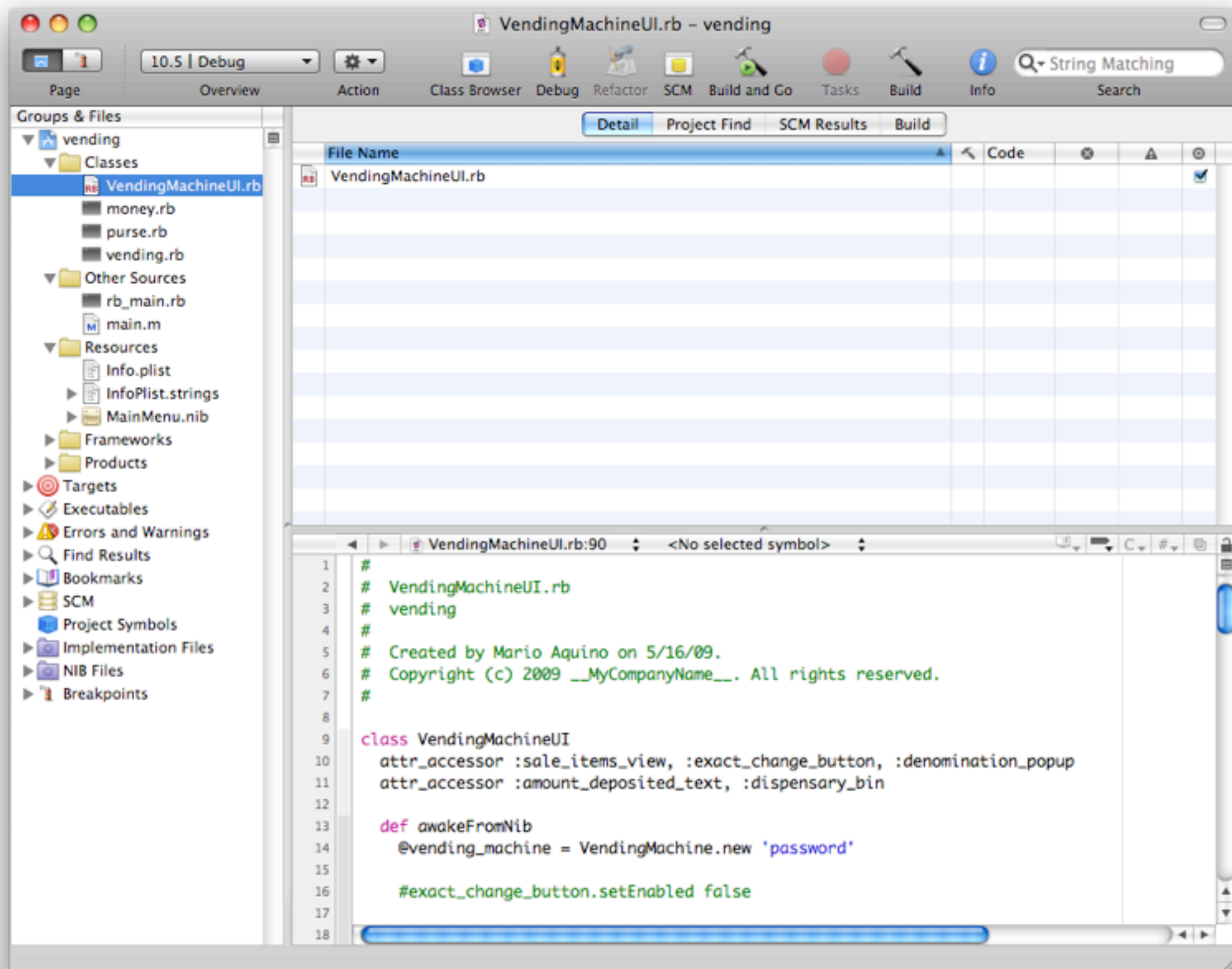
```
    ^
```

```
evil_macruby.rb:9: syntax error, unexpected keyword_end, expecting $end
```

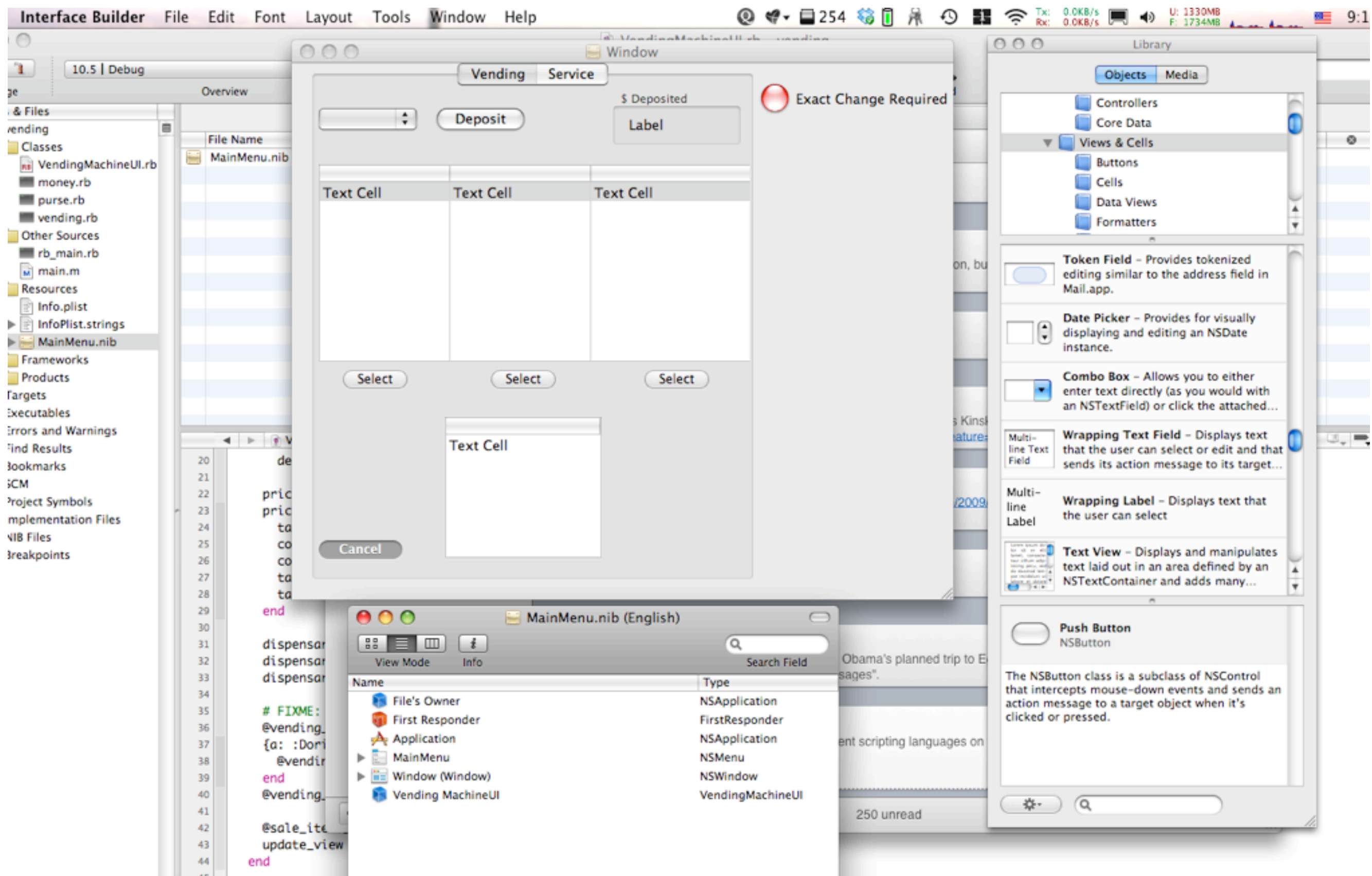
Ruby tools

- macruby
- macrake
- macirb
- macgem
- macri / macrdoc

Xcode



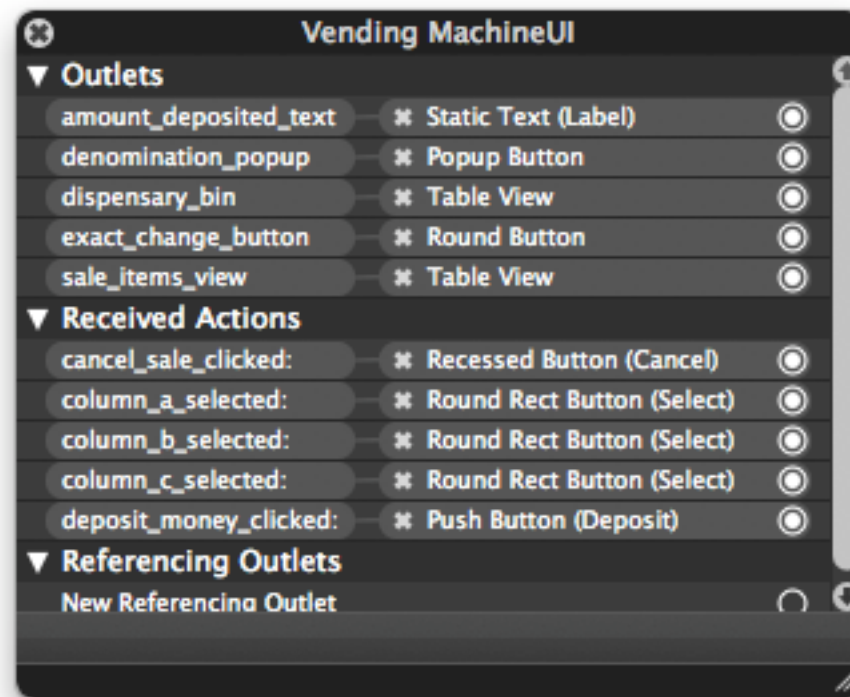
Interface Builder



Interface Builder

```
class VendingMachineUI
  attr_accessor :sale_items_view, :exact_change_button, :denomination_popup
  attr_accessor :amount_deposited_text, :dispensary_bin
```

Outlets

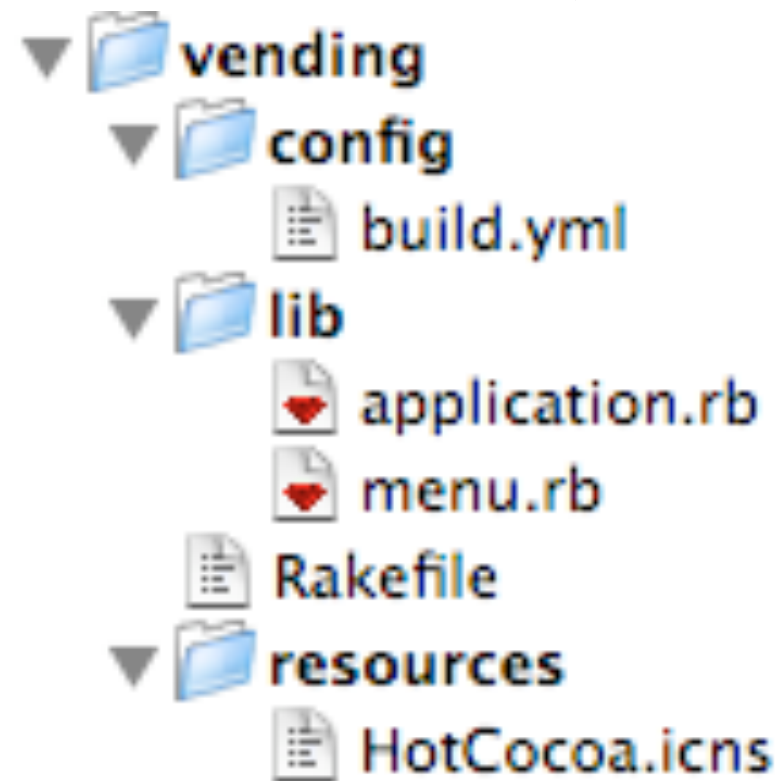


Actions

```
def column_c_selected(sender)
  @vending_machine.select :c
end
```

Hot Cocoa

\$ hotcocoa vending



macrake

```
mario:~/projects/macruby/vending$ macrake -P  
(in /Users/mario/projects/macruby/vending)  
rake build  
rake clean  
rake default  
    run  
rake deploy  
    clean  
rake run  
    build
```

DSL for Graphical Ruby UIs

```
class Application

  include HotCocoa

  def start
    application :name => "Vending" do |appl|
      app.delegate = self
      window :frame => [100, 100, 500, 500], :title => "Vending" do |win|
        win << label(:text => "Hello from HotCocoa", :layout => {:start => false})
        win.will_close { exit }
      end
    end
  end
end
```

UI Event Delegation

```
# file/open  
def on_open(menu)  
end
```

```
# file/new  
def on_new(menu)  
end
```

```
# help menu item  
def on_help(menu)  
end
```

```
# This is commented out, so the minimize menu item is disabled  
#def on_minimize(menu)  
#end
```

```
# window/zoom  
def on_zoom(menu)  
end
```

...

MacRuby Goals

- Make Ruby a 1st-class language for writing native OSX applications
- Embed MacRuby into Objective-C apps
- Leverage LLVM compiler infrastructure

State of Implementation

- 0.4 latest stable release
- Passes 85% ruby language specs
- RSpec doesn't work :(=> Test::Unit does :p
- Missing GUI testing toolkit
- Introduced a language variance