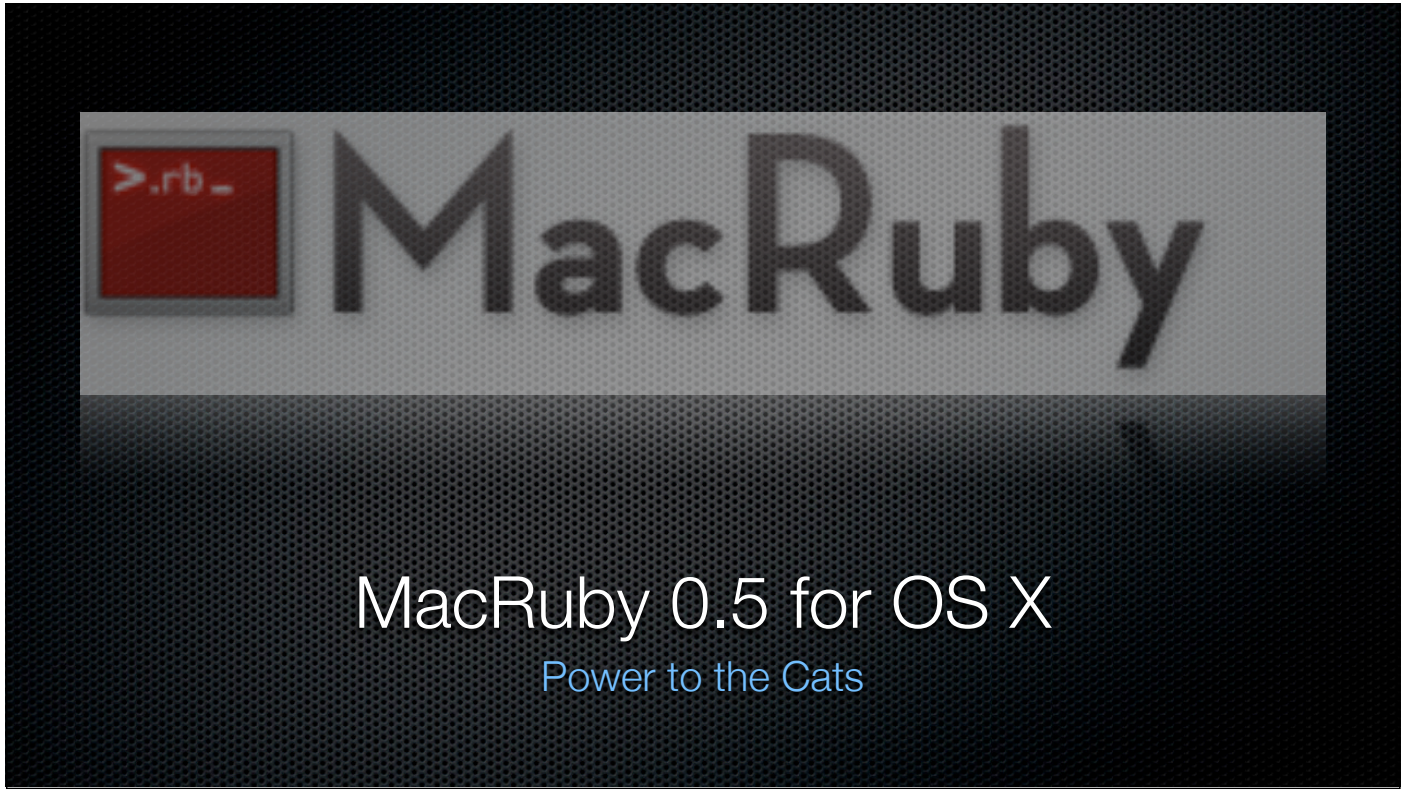
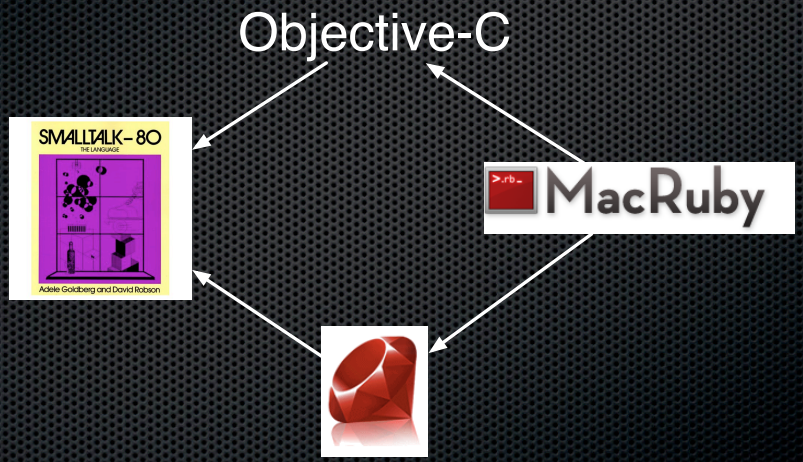


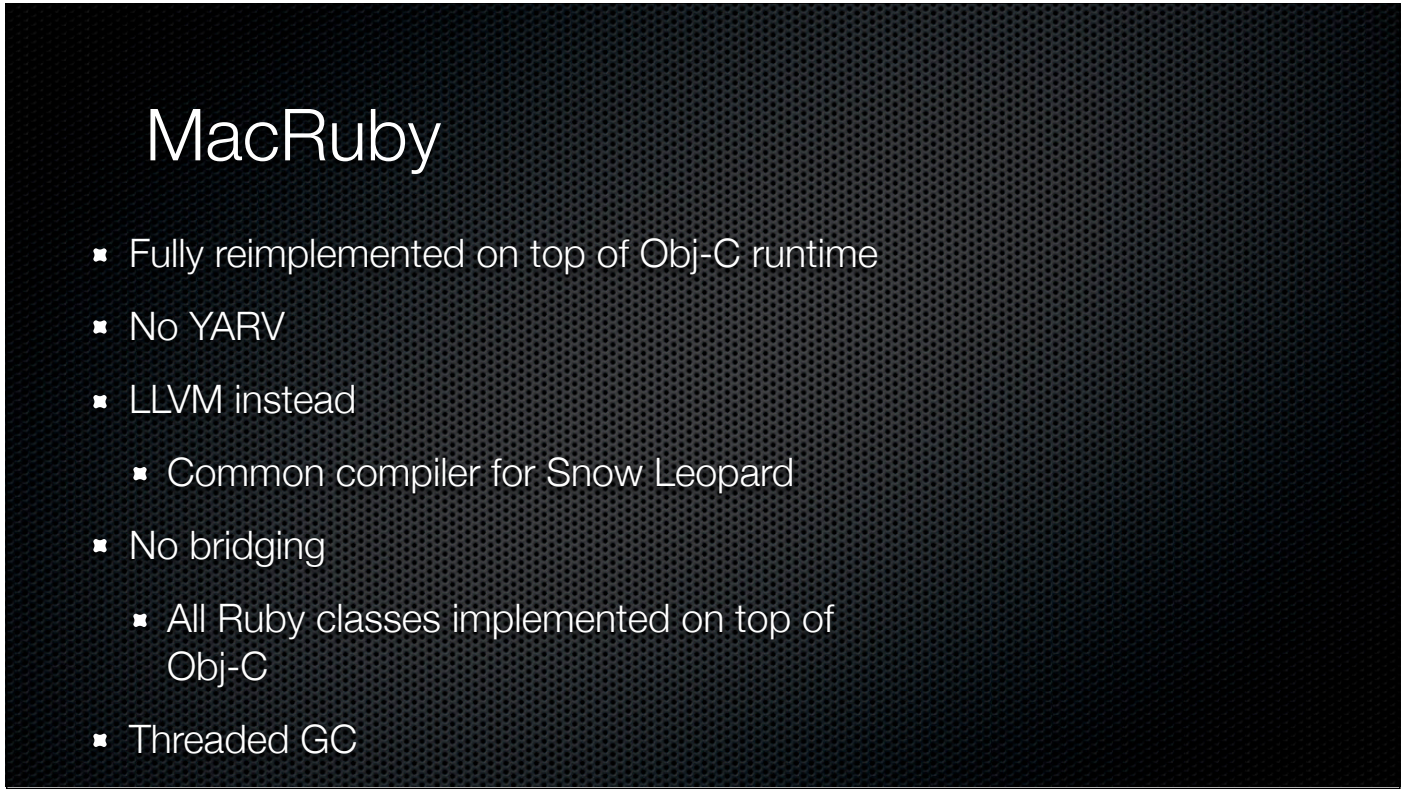
MacRuby 0.5 - A Ruby specifically for and to take advantage of the Mac OS X Platform features.

MacRuby's Origins



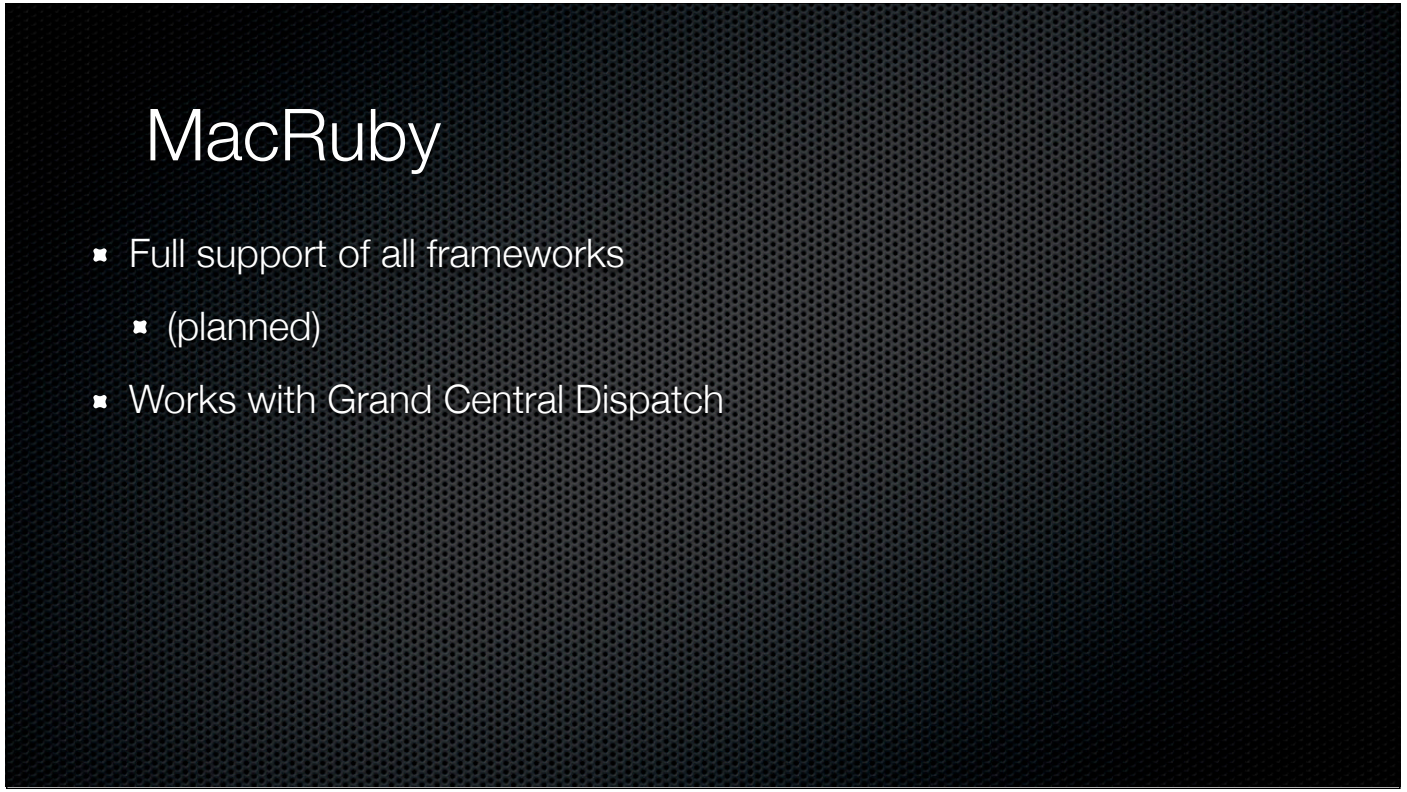
Current Ruby on Mac

- 1.8.7
- Standard interpreter
- Bridge library called RubyCocoa for native apps
- Issues
  - Duplicate class trees in each runtime space
  - Seperate GC cycles
  - Type/object conversion across bridge



4 compiliers available really but all headed towards LLVM core.

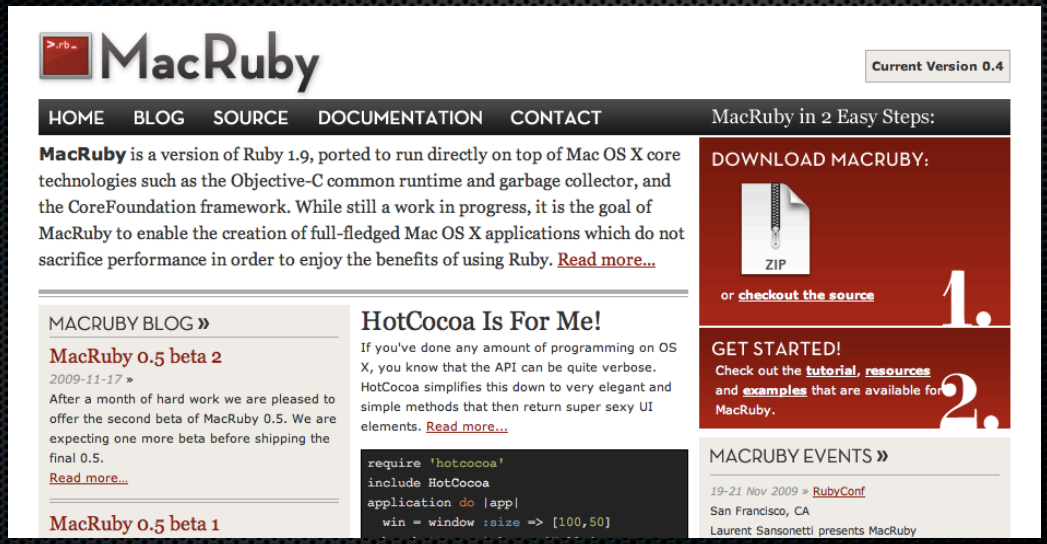
Ruby classes are Obj-C classes, Ruby methods are Obj-C methods, etc.



4 compiliers available really but all headed towards LLVM core.

Ruby classes are Obj-C classes, Ruby methods are Obj-C methods, etc.

MacRuby Site - www.macruby.org



Gems

- Macgems is the command
- Still early in lifecycle so some gems work some don't
- Gems are currently put in:
  - /Library/Frameworks/MacRuby.framework/Versions/0.5/usr/lib/ruby/Gems/1.9.0/gems

```
$ macgem sources -a http://gemcutter.org
$ sudo macgem install textorize-mr
$ textorize -f"Papyrus" -s200 "MacRuby 0.5b1"
$ open output.png

Gems are stored in:
/Library/Frameworks/MacRuby.framework/Versions/0.5/usr/lib/ruby/Gems/1.9.0/gems
```

Core

- Available in:
  - /Library/Frameworks/MacRuby.framework/Versions/0.5/usr/lib/ruby/1.9.0

Macirb

- [~]$ macirb
- irb(main):001:0> "foo".class
- => NSMutableString

Macirb - cont 2

- irb(main):002:0> Object.methods.sort
- => [:!, :!=, :!~, :<, :<=, :<=>, :==, :===, :=~, :>, :>=, :Complex, :Rational, :URI, :__callee__, :__id__, :__meta__?, :__method__, :__native__?, :__send__, :allocate, :ancestors, :autoload, :autoload?, :class_eval, :class_exec, :class_variable_defined?, :class_variable_get, :class_variable_set, :class_variables, :clone, :const_defined?, :const_get, :const_missing, :const_set, :constants, :define_singleton_method, :dup, :enum_for, :eql?, :equal?, :extend, :freeze, :frozen?, :gem, :gem_original_require, :hash, :include?, :included_modules, :inspect, :instance_eval, :instance_exec, :instance_method, :instance_methods, :instance_of?, :instance_variable_defined?, :instance_variable_get, :instance_variable_set, :instance_variables, :is_a?, :kind_of?, :load_bridge_support_file, :method, :method_defined?, :methods, :module_eval, :module_exec, :name, :new, :nil?, :object_id, :private_class_method, :private_instance_methods, :private_method_defined?, :private_methods, :protected_instance_methods, :protected_method_defined?, :protected_methods, :public_class_method, :public_instance_method, :public_instance_methods, :public_method, :public_method_defined?, :public_methods, :public_send, :remove_class_variable, :require, :respond_to?, :send, :singleton_methods, :taint, :tainted?, :tap, :to_enum, :to_s, :to_yaml, :to_yaml_properties, :to_yaml_style, :trust, :untaint, :untrust, :untrusted?, :y, :yaml_as, :yaml_new]

Macirb - cont 3

- irb(main):003:0> Object.ancestors
- => [NSObject, Kernel]
- Hmm - That's different?

XCode

- MacRuby projects just like any XCode project
- Lets take a look

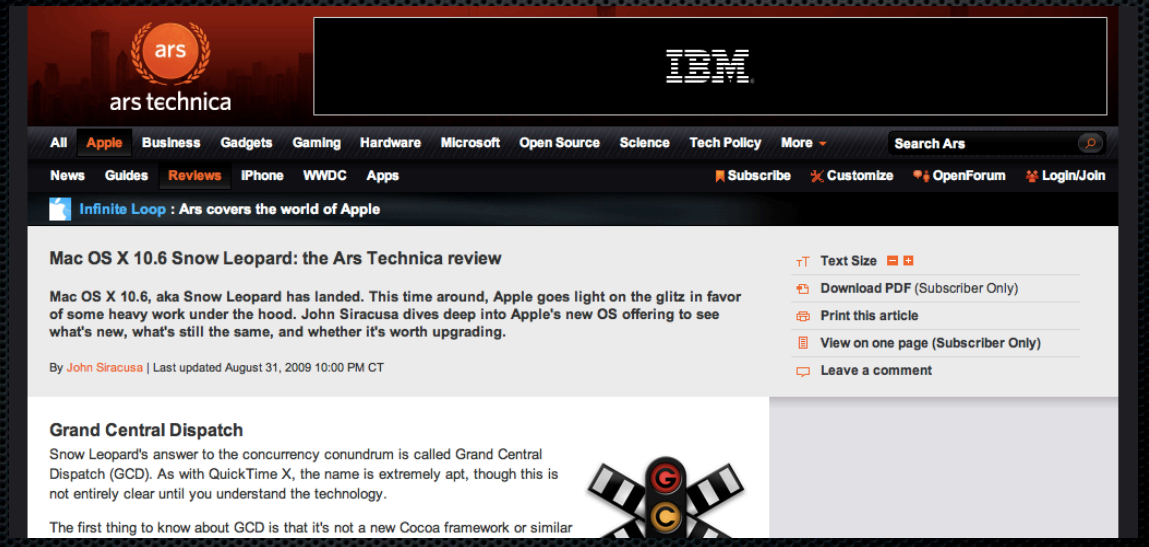
```
Add Dog class
    class Dog
    def name
          @name
    end

    def setName(name)
          @name = name
    end
end

In main function put this.  Then build and show console.

d = Dog.new
d.setName("Bob")
pp d
```

Grand Central Dispatch




```
Add Dog class
    class Dog
    def name
          @name
    end

    def setName(name)
          @name = name
    end
end

In main function put this.  Then build and show console.

d = Dog.new
d.setName("Bob")
pp d
```

GCD Sample Code

```
waiting_chairs = Dispatch::Queue.new('com.apple.waiting_chairs')
semaphore = Dispatch::Semaphore.new(3)
index = -1
while true
  index += 1
  success = semaphore.wait(Dispatch::TIME_NOW)
  if success != 0
    puts "Customer turned away #{index}"
    next
  end
  waiting_chairs.dispatch do
    semaphore.signal
    puts "Shave and a haircut #{index}"
  end
```

Paste this into demo:

```ruby
waiting_chairs = Dispatch::Queue.new('com.apple.waiting_chairs')
semaphore = Dispatch::Semaphore.new(3)
index = -1
while true
  index += 1
  success = semaphore.wait(Dispatch::TIME_NOW)
  if success != 0
    puts "Customer turned away #{index}"
    next
  end
  waiting_chairs.dispatch do
    semaphore.signal
    puts "Shave and a haircut #{index}"
  end
end
```

Interface Builder

- Instance based GUI building
- Demo

Instruments

- Lets you perf-mon your MacRuby apps
  - Monitor memory, threads, # of cores used, etc.

Hot Cocoa

- DSL for building Cocoa GUI's if you don't want to do instance based building of your GUI's

Hot Cocoa Simplest App

```
require 'hotcocoa'

class Application
  include HotCocoa

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