

# Use C to Tune Your Rails Application

by Jared Richardson

Pragmatic author, developer, and speaker

Technical Lead, Cisco Systems



Agile Artisans

<http://AgileArtisans.com>

# Why Bother?

- Existing Libraries
- Legacy Hardware Libraries
- Re-write Hot Spots
- Ruby VM



# Four Parts

- Ruby C Extensions
- InlineRuby
- RubyToC
- Rails and C



# Tools We'll Use

- ruby
- rails
- gcc
- mkmf
- Ruby Inline



# Ruby C Extension

- Fairly easy
- Use a few conventions
- Can even share variables



# What We Need

- C code
- Make file
- Ruby code



# C Code

```
#include "ruby.h"

static VALUE SampleC;

static VALUE exec_me(VALUE self) {
    printf("\n\n exec_me, written in C, run from Ruby! \n\n");
    return Qnil;
}

void Init_SampleC() {
    SampleC = rb_define_class("SampleC", rb_cObject);
    rb_define_method(SampleC, "exec", exec, 0);
}
```



# C Code

```
#include "ruby.h"

static VALUE SampleC;

static VALUE exec_me(VALUE self) {
    printf("\n\n exec_me, written in C, run from Ruby! \n\n");
    return Qnil;
}

void Init_SampleC() {
    SampleC = rb_define_class("SampleC", rb_cObject);
    rb_define_method(SampleC, "exec", exec, 0);
}
```



# C Code

```
#include "ruby.h"

static VALUE SampleC;

static VALUE exec_me(VALUE self) {
    printf("\n\n exec_me, written in C, run from Ruby! \n\n");
    return Qnil;
}

void Init_SampleC() {
    SampleC = rb_define_class("SampleC", rb_cObject);
    rb_define_method(SampleC, "exec", exec, 0);
}
```



# C Code

```
#include "ruby.h"

static VALUE SampleC;

static VALUE exec_me(VALUE self) {
    printf("\n\n exec_me, written in C, run from Ruby! \n\n");
    return Qnil;
}

void Init_SampleC() {
    SampleC = rb_define_class("SampleC", rb_cObject);
    rb_define_method(SampleC, "exec", exec, 0);
}
```



# C Code

```
#include "ruby.h"

static VALUE SampleC;

static VALUE exec_me(VALUE self) {
    printf("\n\n exec_me, written in C, run from Ruby! \n\n");
    return Qnil;
}

void Init_SampleC() {
    SampleC = rb_define_class("SampleC", rb_cObject);
    rb_define_method(SampleC, "exec_me", exec_me, 0);
}
```



# extconf.rb

```
require 'mkmf'  
create_makefile("SampleC")
```

To create your Makefile...

```
ruby extconf.rb
```



# Practical Tip

- If mkmf isn't installed, you'll see "ruby: no such file to load -- mkmf (LoadError)"
- "sudo apt-get install ruby1.8-dev"



# Ruby Code

```
require "SampleC"
```

```
handle = SampleC.new
```

```
handle.exec_me
```



# Quick Demo



Agile Artisans

# More Information

- What else can I do?



# Ruby Objects to C

With Type Checking

`int => NUM2INT(Numeric)`

`long => NUM2LONG(Numeric)`



# Ruby Objects to C

Without Type Checking (faster)

int => FIX2INT(Numeric)

long => FIX2LONG(Numeric)



# Converting C to Ruby Objects

`INT2NUM(int) => Fixnum`

`INT2FIX(int) => Fixnum`

`rb_str_new2(char *) => String`



# Sharing Variables

```
VALUE column_names = rb_ary_new();
```

```
rb_iv_set(self, "@column_names", column_names);
```

```
rb_ary_push( column_names, next_columnName);
```

```
rb_ary_push( column_names, rb_str_new2(next_columnName));
```



# Eval Ruby in C

- `rb_eval_string(my_list.each{|x| x.do_something }");`
- `rb_funcall(receiver, method_id, argc, ...)`



# Inline C

- Ruby gem
- `gem install RubyInline`



# Ruby Inline Code

```
require "rubygems"  
require "inline"  
class RubyToC  
  inline do |builder|  
    builder.c %Q{  
      public void exec_me() {  
        printf("Written in C, run from Ruby.");  
      }  
    }  
  end  
end  
t = RubyToC.new()  
t.exec_me
```



# Ruby Inline Code

```
require "rubygems"  
require "inline"  
class RubyToC  
  inline do |builder|  
    builder.c %Q{  
      public void exec_me() {  
        printf("Written in C, run from Ruby.");  
      }  
    }  
  end  
end  
t = RubyToC.new()  
t.exec_me
```



# Ruby Inline Code

```
require "rubygems"  
require "inline"  
class RubyToC  
  inline do |builder|  
    builder.c %Q{  
      public void exec_me() {  
        printf("Written in C, run from Ruby.");  
      }  
    }  
  end  
end  
t = RubyToC.new()  
t.exec_me
```



# Ruby Inline Code

```
require "rubygems"  
require "inline"  
class RubyToC  
  inline do |builder|  
    builder.c %Q{  
      public void exec_me() {  
        printf("Written in C, run from Ruby.");  
      }  
    }  
  end  
end  
t = RubyToC.new()  
t.exec_me
```



# Ruby Inline Code

```
require "rubygems"  
require "inline"  
class RubyToC  
  inline do |builder|  
    builder.c %Q{  
      public void exec_me() {  
        printf("Written in C, run from Ruby.");  
      }  
    }  
  end  
end  
t = RubyToC.new()  
t.exec_me
```



# Ruby Inline Code

```
require "rubygems"  
require "inline"  
class RubyToC  
  inline do |builder|  
    builder.c %Q{  
      public void exec_me() {  
        printf("Written in C, run from Ruby.");  
      }  
    }  
  end  
end  
t = RubyToC.new()  
t.exec_me
```



# Ruby Inline Code

```
require "rubygems"  
require "inline"  
class RubyToC  
  inline do |builder|  
    builder.c %Q{  
      public void exec_me() {  
        printf("Written in C, run from Ruby.");  
      }  
    }  
  end  
end  
t = RubyToC.new()  
t.exec_me
```



# Ruby Inline Code

```
require "rubygems"  
require "inline"  
class RubyToC  
  inline do |builder|  
    builder.c %Q{  
      public void exec_me() {  
        printf("Written in C, run from Ruby.");  
      }  
    }  
  end  
end  
handle = RubyToC.new()  
handle.exec_me
```



# Quick Demo



Agile Artisans

# Who Codes C?

- 'Write less code' -DHH
- Don't write C
- Generate it



# Ruby To C

```
gem install rubytoc
```

hat tip to Laurent Sansonetti

<http://chopine.be/lrz/>



Agile Artisans

# Sample Code

```
class F
  def factorial(n)
    f = 1
    n.downto(2) { |x| f *= x }
    return f
  end

  def main
    return factorial(5)
  end
end
```



# Usage

```
./translate2.rb demo/factorial.rb > f.c
```



# Converts To...

```
long factorial(long n) {  
    long f;  
    long x;  
    f = 1;  
    x = n;  
    while (x >= 2) {  
        f = f * x;  
        x = x - 1;  
    };  
    return f;  
}
```



# The Author's Warning...

48

49

```
INSTALL:
```

50

51

```
+ Um. Please don't install this crap yet...
```

52

from the README.TXT



Agile Artisans

# More Info

- <http://rubyforge.org/projects/ruby2c/>
- [http://community.livejournal.com/ruby\\_lang/5157.html](http://community.livejournal.com/ruby_lang/5157.html)
- <http://www.zenspider.com/pipermail/ruby/2004-October/001658.html>



# That's Ruby

- What about Rails?
- Find the hot spots



# Rails Logging

Processing MainController#search\_students (for 127.0.0.1 at 2007-07-03 15:29:26) [GET]

Session ID: b4c3826607bd5ff51f3fa34b45d96c76

Parameters: {"action"=>"search\_students", "controller"=>"main"}

Student Columns (0.001862) SHOW FIELDS FROM students

Student Load (0.000149) SELECT \* FROM students WHERE (students.`user\_name` = 'jared') LIMIT 1

Student Load (0.000171) SELECT \* FROM students WHERE (students.`city` = 'Gotham' AND students.

Found 13 matches

Rendering main/index

Rendered main/\_logo (0.00008)

Rendered main/\_tabs (0.00012)

Rendered main/\_top\_of\_page (0.00167)

Rendered main/\_search\_students (0.00639)

Rendered main/\_nav\_left\_hand\_side (0.00088)

Rendered main/\_footer (0.00010)

Completed in 0.02753 (36 reqs/sec) | Rendering: 0.01113 (40%) | DB: 0.00298 (10%) | 200 OK

[http://blah\_mongrel/main/search\_students]



# Rails Logging

Processing MainController#search\_students (for 127.0.0.1 at 2007-07-03 15:29:26) [GET]

Session ID: b4c3826607bd5ff51f3fa34b45d96c76

Parameters: {"action"=>"search\_students", "controller"=>"main"}

Student Columns (0.001862) SHOW FIELDS FROM students

Student Load (0.000149) SELECT \* FROM students WHERE (students.`user\_name` = 'jared') LIMIT 1

Student Load (0.000171) SELECT \* FROM students WHERE (students.`city` = 'Gotham' AND students.

Found 13 matches

Rendering main/index

Rendered main/\_logo (0.00008)

Rendered main/\_tabs (0.00012)

Rendered main/\_top\_of\_page (0.00167)

Rendered main/\_search\_students (0.00639)

Rendered main/\_nav\_left\_hand\_side (0.00088)

Rendered main/\_footer (0.00010)

**Completed in 0.02753 (36 reqs/sec) | Rendering: 0.01113 (40%) | DB: 0.00298 (10%) | 200 OK**  
**[http://blah\_mongrel/main/search\_students]**



# Rails Logging

Completed in 0.02753 (36 reqs/sec) | Rendering:  
0.01113 (40%) | DB: 0.00298 (10%) | 200 OK  
[[http://blah\\_mongrel/main/search\\_students](http://blah_mongrel/main/search_students)]



# Quick and Dirty

```
start = Time.now
```

```
  # run code in here ...
```

```
done = Time.now
```

```
elapsed_time = done - start
```

```
logger.info("Spent #{elapsed_time} seconds ")
```



# Tuning Algorithm

- Watch the log
- Find slow controllers
- Zero in with timing statements
- Convert to C



# More on Tuning Rails

<http://www.rubyinside.com/how-to-profile-your-rails-application-and-make-rails-go-vroom-565.html>



Agile Artisans

# Demo

- Native Ruby on Rails
- Rails with Inline C
- Rails with Ruby C Extension



# Rails and Ruby Inline

- Reloads break RubyInline
- Put code in `/lib`
- Or run in *production* mode



# Rails and C Extensions

- Be sure to "make install"
- Then require "Sample\_C"



# Existing C Extensions

- REXML (ruby) vs. libxml (C)
- 10x faster



# Performance Implications

- Ruby C Extensions -> Little Overhead
- Inline C -> First Time Hit

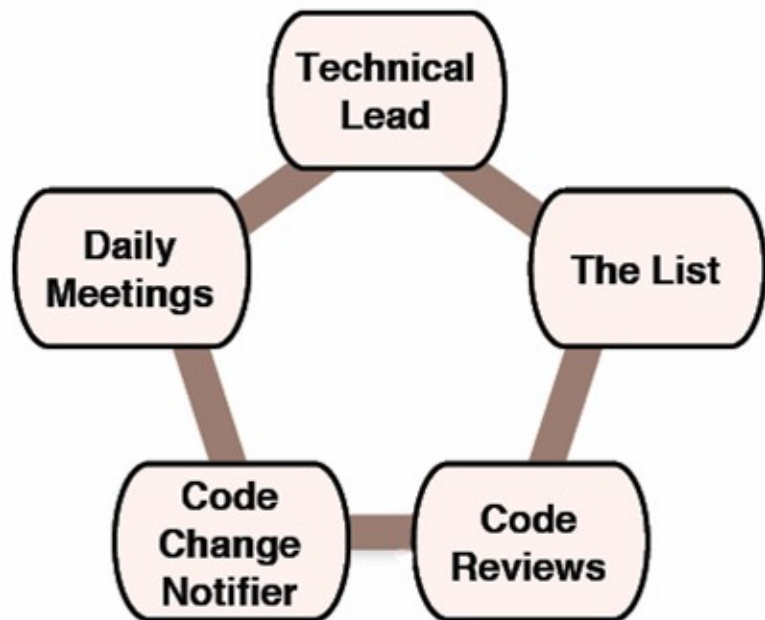


# Web Resources

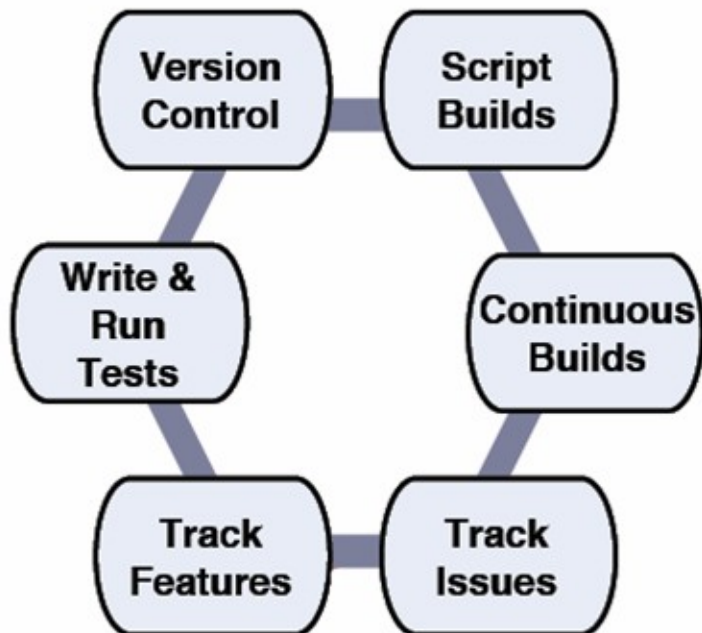
- [http://www.rubycentral.com/pickaxe/ext\\_ruby.html](http://www.rubycentral.com/pickaxe/ext_ruby.html)
- <http://www.zenspider.com/ZSS/Products/RubyInline/>
- <http://www.jaredrichardson.net/blog/2006/03/25/>
- <http://libxml.rubyforge.org/>
- <http://agileartisans.com/main/blog/41>
- <http://www.rubyinside.com/how-to-profile-your-rails-application-and-make-rails-go-vroom-565.html>



## Techniques

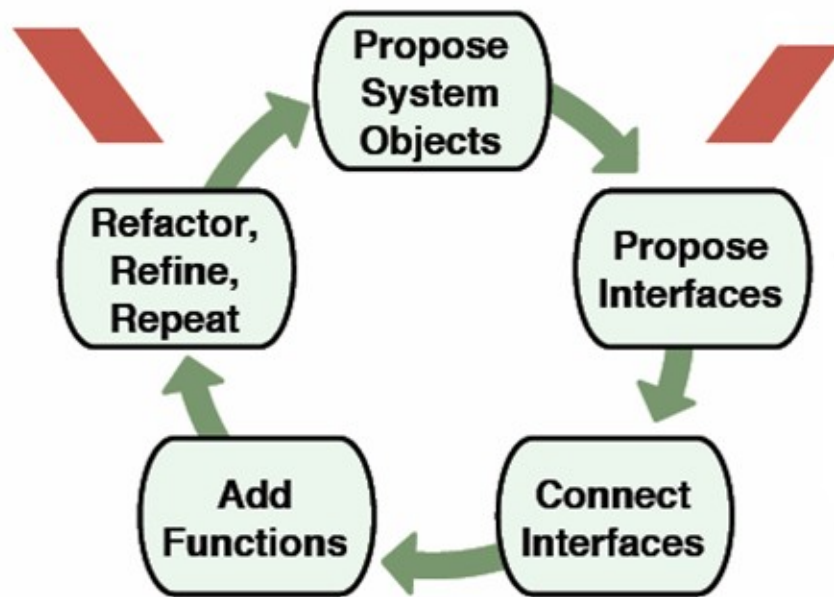


## Infrastructure



# Ship It!

*A Practical Guide to Successful Software Projects*



## Process

