RubyConf*MI

Libraries for Developers

Pat Eyler
pat.eyler@gmail.com

Who am I?

Ruby Hacker

Writer

Ruby Community Builder

$0.25 tour

example code
test/unit – ZenTest, rspec, & rcov
Profile & Profiler__ – ruby-prof
benchmark – RubyInline

Philosophy

Test

Code

Refactor

Profile

Optimize

Example Code
short and easy to read

```
#  RedGreen is a filter for Test::Unit tests run from the
#  commandline.  It will colorize passing tests green, and failing
#  tests red.

STDIN.readlines.each do |line|
  if line =~ /\d+ tests, \d+ assertions, 0 failures, 0 errors/
    line = "\e[32m#{$&}\e[0m"
  elsif line =~ /\d+ tests, \d+ assertions, (\d+) failures, (\d+) errors/
    if $1 != 0 || $2 != 0
      line = "\e[31m#{$&}\e[0m"
    end
  end
  puts line
end
```

Umm, no.

Let's try something else instead.

Example Code

```
for num in 1..1_000 do
  is_prime = 1
  for x in 2..(num - 1) do
    if (num % x == 0)
      is_prime = x
      break
    end
  end
  if is_prime == 1
    puts "#{num} is a prime number"
  else
    puts "#{num} equals #{is_prime} * #{num/is_prime}"
  end
end
```

test/unit

Test your code before you write it

assertion based

test/unit

```
require 'test/unit'
require 'SimpleSpread'

class TestSpread < Test::Unit::TestCase

  def test_get_empty_cell
    sheet = Sheet.new()
    assert_equal("", sheet.get("A1"))
    assert_equal("", sheet.get("ZX347"))
  end

end
```

test/unit

```
bash$ ruby test_spread.rb
test_spread.rb:2:in `require': no such file to load -- SimpleSpread
(LoadError)
        from test_spread.rb:2

bash$ ruby test_spread.rb
Loaded suite test_spread
Started
E
Finished in 0.000743 seconds.

  1) Error:
test_get_empty_cell(TestSpread):
NoMethodError: undefined method `get' for #<Sheet:0x401fb9f8>
    test_spread.rb:8:in `test_get_empty_cell'

1 tests, 0 assertions, 0 failures, 1 errors


bash$ ruby test_spread.rb
Loaded suite test_spread
Started
.
Finished in 0.000525 seconds.

1 tests, 2 assertions, 0 failures, 0 errors
bash$
```

test/unit

```
class Sheet
  def get(cell)
    ""
  end
end


class TestSpread < Test::Unit::TestCase

  def test_get_empty_cell
    sheet = Sheet.new()
    assert_equal("", sheet.get("A1"))
    assert_equal("", sheet.get("ZX347"))
  end

  def test_put
    sheet = Sheet.new()
    a1 = "A string"
    sheet.put("A1", a1)
    assert_equal(a1, sheet.get("A1"))
    sheet.put("A1", "foo")
    assert_equal("foo", sheet.get("A1"))
    sheet.put("A1", "")
    assert_equal("", sheet.get("A1"))
  end

end
```

test/unit

```
bash$ ruby test_spread.rb
Loaded suite test_spread
Started
.E
Finished in 0.001017 seconds.

  1) Error:
test_put(TestSpread):
NoMethodError: undefined method `put' for #<Sheet:0x401faabc>
    test_spread.rb:15:in `test_that_text_cells_are_stored'

2 tests, 2 assertions, 0 failures, 1 errors
bash$
```

test/unit

```
class Sheet
  def initialize
    @cells = Hash.new
    @cells.default = ""
  end

  def get(cell)
    @cells[cell]
  end

  def put(cell, value)
    @cells[cell] = value
  end
end
```

```
bash$ ruby test_spread.rb
Loaded suite test_spread
Started
..
Finished in 0.000955 seconds.

2 tests, 5 assertions, 0 failures, 0 errors
bash$
```

test/unit

```
bash$ ruby test_spread.rb -n test_put

Loaded suite test_spread
Started
.
Finished in 0.000735 seconds.

1 tests, 3 assertions, 0 failures, 0 errors
```

test/unit

```
bash$  ruby -Ilib test/test_r43.rb
Loaded suite test/test_r43
Started
.....................................................................
.........................................
Finished in 9.543823 seconds.
105 tests, 650 assertions, 0 failures, 0 errors
```

```
bash$  ruby test_cipher.rb
Loaded suite test_cipher
Started
F...
Finished in 0.023828 seconds.
  1) Failure:
test_char2num(TestCipher)
[test_cipher.rb:9]:
<[4, 0]> expected but was
<[4, 1]>.
4 tests, 9 assertions, 1 failures, 0 errors
```

test/unit

```
  1) Failure:
test_xml_is_same(TestXML) [bogus_test.rb:8]:
<"<?xml version='1.0' encoding='UTF-8'?><feed
xmlns:dc='http://purl.org/dc/elents/1.1/'
xmlns='http://43things.com/xml/2005/rc#'>\n  <title>People search relts for
\"Sean\" on 43 Things</title>\n  <link
href='http://www.43things.com/srch/query?q=Sean' rel='alternate'
type='text/html'/>\n  <query>Sean</query>\n<gination>\n
<offset>0</offset>\n  <max>20</max>\n  <total>11</total>\n  <nextffset/>\n
<previous_offset/>\n</pagination>\n<person person_id='Sean'>\n
<usname>Sean</username>\n  <name>Sean McNulty</name>\n
<url>http://seanmcnulty.bgspot.com/</url>\n
<flickr_username>Sean</flickr_username>\n
<num_open_goals5</num_open_goals>\n  <profile_image_url>
http://images.43things.com/profile/000/09/2357s.jpg</profile_image_url>\n
```

(oh, yeah ... it goes on for another 118 lines)

unit_diff

```
bash$ ruby my_test.rb | unit_diff


1) Failure:
test_xml_is_same(TestXML) [bogus_test.rb:8]:
17c17
<   <num_open_goals>15</num_open_goals>
---
>   <num_open_goals>16</num_open_goals>

1 tests, 1 assertions, 1 failures, 0 errors
```

autotest

Continuous Testing

Immediate Feedback

Fewer Context Switches

rspec

behavior (specification) based

bash$ *spec my_specfile*

rspec

```
context "Empty spread sheet" do
  setup do
    @sheet = Sheet.new
  end
  specify "cells exist and are empty" do
    @sheet.get('A1').should_equal ''
  end
end


context "Text values in spread sheet" do
  setup do
    @sheet = Sheet.new
    @sheet.put('A1', "some text")
  end
  specify "cells with text values return strings" do
    @sheet.get('A1').should_be_an_instance_of String
  end
  specify "cells store text values correctly" do
    @sheet.get('A1').should_equal "some text"
  end
end
```

DSLs (a diversion)

Domain Specific Languages

closer to the user

not just for programming

DSLs (a diversion)

What is (or isn't) a DSL?

```
def test_char2num
  code = Cipher.new
  assert_equal([0,0], code.char2num('A'))
  assert_equal([4,0], code.char2num('a'))
end
```

```
specify "cells store text values correctly" do
  @sheet.get('A1').should_equal "some text"
end
```

rspec

```
bash$ spec rspec_spread.rb

...

Finished in 0.002249 seconds

3 specifications, 0 failures
```

rspec

```
Usage: spec [options] (FILE|DIRECTORY|GLOB)+

        --diff                       Show unified diff of Strings that are
                                     expected to be equal when they are not
    -s, --spec SPECIFICATION_NAME    Execute a single specification
    -f, --format FORMAT              Builtin formats: specdoc|s|rdoc|r|html|h
                                     You can also specify a custom formatter class
                                     (in which case you should also specify
                                     --require)
    -r, --require FILE               Require FILE before running specs
                                     Useful for loading custom formatters or other
                                     extensions
    -b, --backtrace                  Output full backtrace
    -d, --dry-run                    Don't execute specs
    -v, --version                    Show version
    -h, --help                       You're looking at it
```

rspec

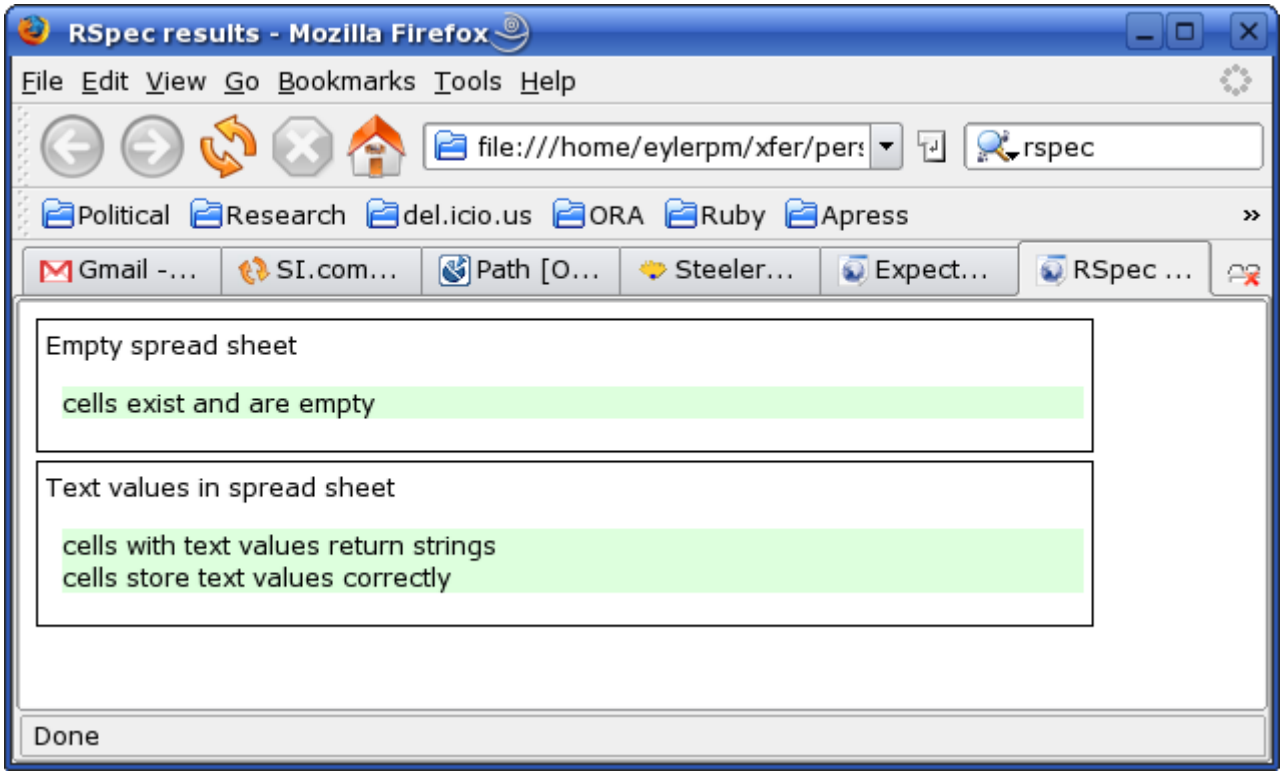
```
bash$ spec -s “cells exist and are empty” rspec_foo.rb


.

Finished in 0.000975 seconds

1 specification, 0 failures
bash$
```

rspec

```
bash$ spec -f s rspec_foo.rb


Empty spread sheet
- cells exist and are empty

Text values in spread sheet
- cells with text values return strings
- cells store text values correctly

Finished in 0.002767 seconds

3 specifications, 0 failures
```

rspec

```
bash$  spec -f h rspec_spread.rb
```



rcov

show test coverage

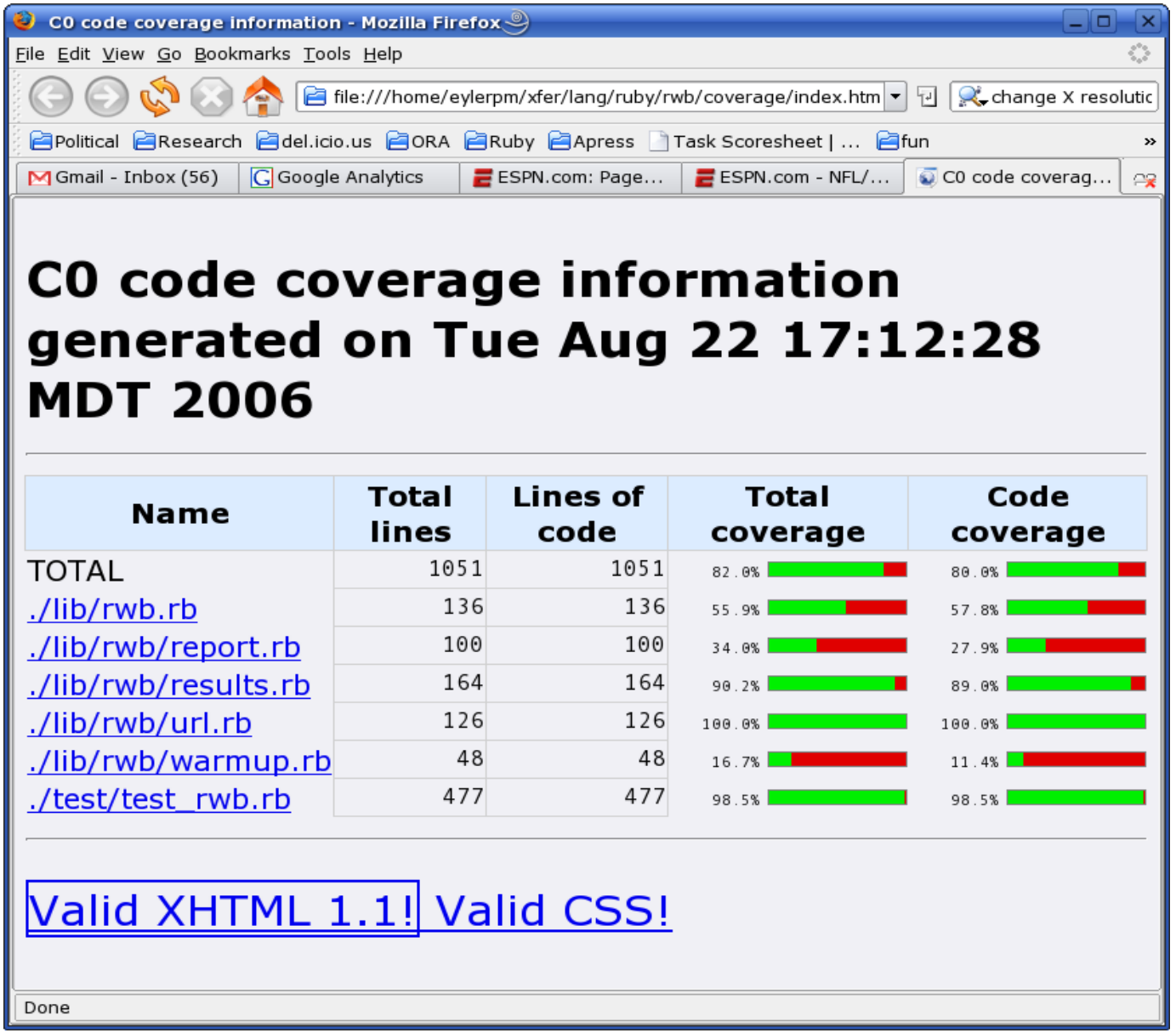
cross references

bash$ *rcov my_code*

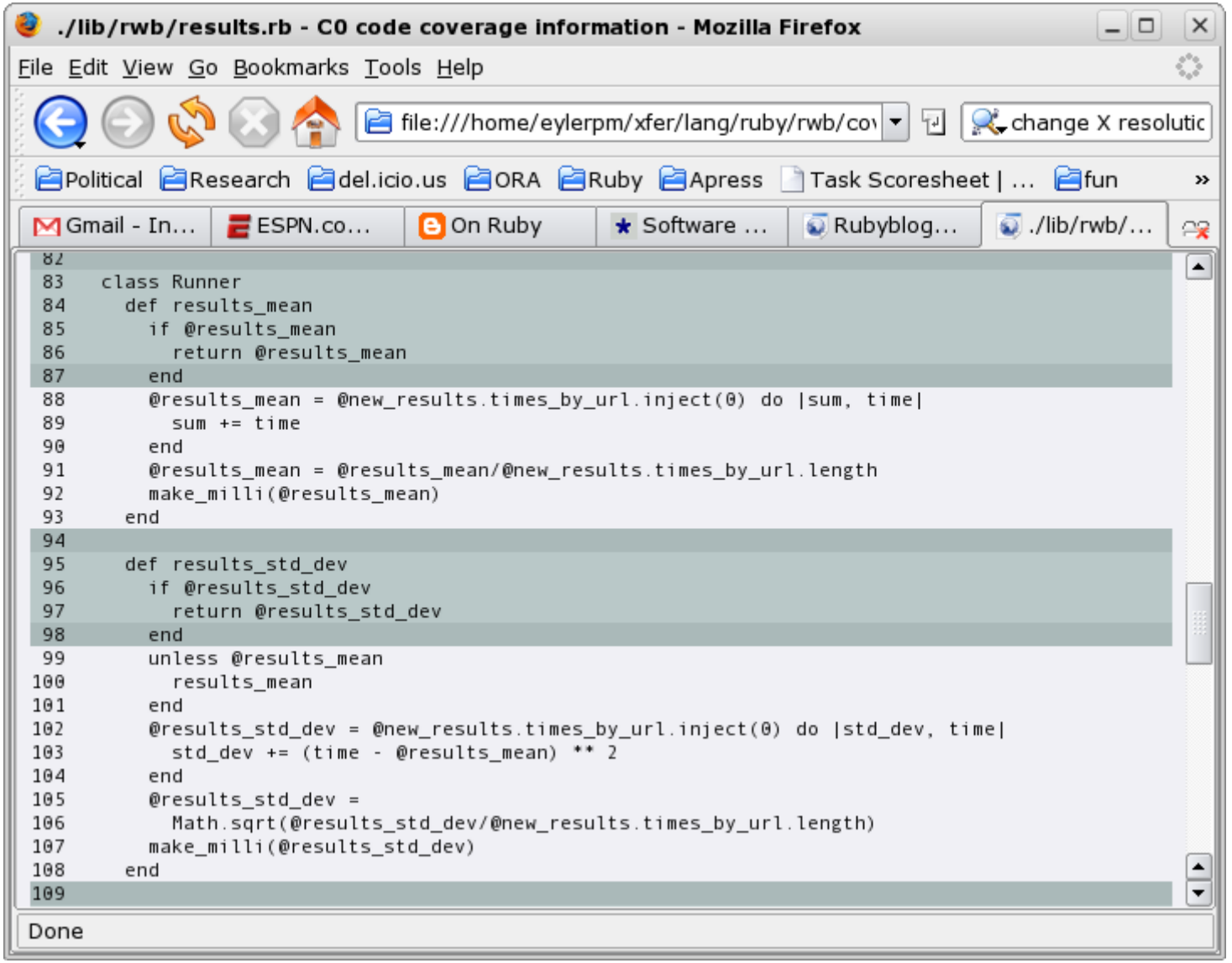
rcov

```
$ rcov -Ilib test/test_rwb.rb
Loaded suite test/test_rwb
Started
..................................................
Finished in 0.020664 seconds.

52 tests, 95 assertions, 0 failures, 0 errors
Generating "coverage/_-lib-rwb-url_rb.html"
Generating "coverage/_-lib-rwb-warmup_rb.html"
Generating "coverage/_-lib-rwb_rb.html"
Generating "coverage/_-lib-rwb-report_rb.html"
Generating "coverage/_-test-test_rwb_rb.html"
Generating "coverage/_-lib-rwb-results_rb.html"
```

rcov


C0 code coverage information - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
file:///home/eylerpm/xfer/lang/ruby/rwb/coverage/index.htm
change X resolutic
Political Research del.icio.us ORA Ruby Apress Task Scoresheet | ... fun
Gmail - Inbox (56) Google Analytics ESPN.com: Page... ESPN.com - NFL/... C0 code coverag...
C0 code coverage information generated on Tue Aug 22 17:12:28 MDT 2006
Name Total lines Lines of code Total coverage Code coverage
TOTAL 1051 1051 82.0% 80.0%
./lib/rwb.rb 136 136 55.9% 57.8%
./lib/rwb/report.rb 100 100 34.0% 27.9%
./lib/rwb/results.rb 164 164 90.2% 89.0%
./lib/rwb/url.rb 126 126 100.0% 100.0%
./lib/rwb/warmup.rb 48 48 16.7% 11.4%
./test/test_rwb.rb 477 477 98.5% 98.5%
Valid XHTML 1.1! Valid CSS!
Done

rcov


./lib/rwb/results.rb - C0 code coverage information - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
file:///home/eylerpm/xfer/lang/ruby/rwb/co
change X resolutic
Political Research del.icio.us ORA Ruby Apress Task Scoresheet | ... fun
Gmail - In... ESPN.co... On Ruby Software ... Rubyblog... ./lib/rwb/...
82
83 class Runner
84 def results_mean
85 if @results_mean
86 return @results_mean
87 end
88 @results_mean = @new_results.times_by_url.inject(0) do |sum, time|
89 sum += time
90 end
91 @results_mean = @results_mean/@new_results.times_by_url.length
92 make_milli(@results_mean)
93 end
94
95 def results_std_dev
96 if @results_std_dev
97 return @results_std_dev
98 end
99 unless @results_mean
100 results_mean
101 end
102 @results_std_dev = @new_results.times_by_url.inject(0) do |std_dev, time|
103 std_dev += (time - @results_mean) ** 2
104 end
105 @results_std_dev =
106 Math.sqrt(@results_std_dev/@new_results.times_by_url.length)
107 make_milli(@results_std_dev)
108 end
109
Done

profile

From the command line

See what your code is doing

bash$ *ruby -r profile my_code.rb*

Profiler__

in your code

specific

```
require 'rwb'
require 'profiler'
urls = RWB::Builder.new()
urls.add_url(10, "http://localhost")
tests = RWB::Runner.new(urls, 1000, 200)
tests.run
Profiler__::start_profile
tests.report_header
tests.report_overall
Profiler__::stop_profile
Profiler__::print_profile($stderr)
```

profile

```
 %   cumulative   self              self     total
time   seconds   seconds    calls  ms/call  ms/call  name
70.74    598.91    598.91    10001    59.89   168.69  Range#each
14.93    725.30    126.39  5775223     0.02     0.02  Fixnum#%
14.02    843.97    118.67  5785223     0.02     0.02  Fixnum#==
 0.13    845.07      1.10    10000     0.11     0.17  Kernel.puts
 0.07    845.67      0.60    20000     0.03     0.03  IO#write
 0.07    846.27      0.60    27540     0.02     0.02  Fixnum#to_s
 0.02    846.43      0.16    10000     0.02     0.02  Fixnum#-
 0.02    846.58      0.15     8770     0.02     0.02  Fixnum#/
 0.00    846.58      0.00        1     0.00 846580.00  #toplevel


real
16m10.003s
user    14m6.581s
sys     0m13.129s
```

OUCH!

ruby-prof

Save Time

More Features

ruby-prof

```
bash$  time ruby-prof primes.rb

Thread ID: 537836452
 %self  cumulative  total  self hildren  calls self/call total/call  name
 63.03     21.96     34.84 21.96 12.88    10001  0.00     0.00     Range#each
  0.20     22.03      0.18  0.07  0.11    10000  0.00     0.00     Kernel#puts
  0.00     22.03     34.84  0.00 34.84        1  0.00    34.84     Kernel#load
  0.32     22.14      0.11  0.11  0.00    20000  0.00     0.00     IO#write
  0.14     22.19      0.05  0.05  0.00    27540  0.00     0.00     Fixnum#to_s
 17.25     28.20      6.01  6.01  0.00  5785223  0.00     0.00     Fixnum#==
  0.00     28.20      0.00  0.00  0.00     8770  0.00     0.00     Fixnum#/
  0.06     28.22      0.02  0.02  0.00    10000  0.00     0.00     Fixnum#-
 18.92     34.81      6.59  6.59  0.00  5775223  0.00     0.00     Fixnum#%
  0.09     34.84      0.03  0.03  0.00    10001  0.00     0.00     <Class::Object>#allocate
  0.00     34.84     34.84  0.00 34.84        1  0.00    34.84     #toplevel


 real    0m37.225s
 user    0m22.029s
 sys     0m13.001s
```

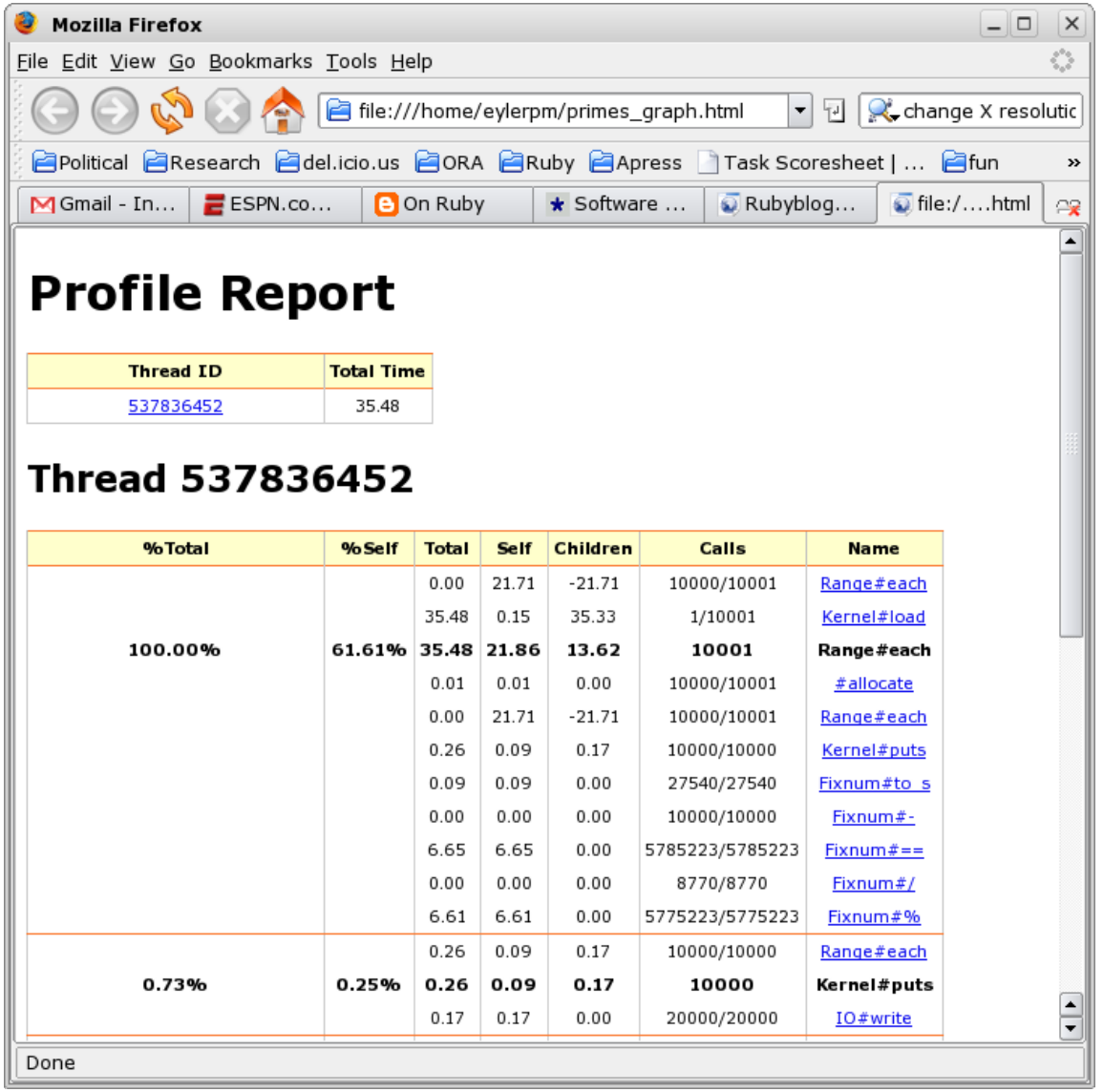
ruby-prof

```
-p, --printer=printer            Select a printer:
                                   flat - Prints a flat profile as text (default).
                                   graph - Prints a graph profile as text.
                                   graph_html - Prints a graph profile as html.
-m, --min_percent=min_percent    The minimum percent a method must take before
                                   being included in output reports
-f, --file=path                  Output results to a file instead of standard out.
-c, --clock-mode=clock_mode      Select a clock mode:
                                   process - Use process time (default).
                                   wall - Use wall time.
                                   cpu - Use the CPU clock counter
                                         (only supported on Pentium and PowerPCs).
```

ruby-prof

```
Thread ID: 537836452
  %total   %self     total      self    children      calls   Name
--------------------------------------------------------------------------------
                     0.00      0.31     -0.31    1000/1001     Range#each
                     0.50      0.03      0.47       1/1001     Kernel#load
 100.00%  68.00%     0.50      0.34      0.16         1001     Range#each
                     0.00      0.00      0.00    1000/1001     <Class::Object>#allocate
                     0.00      0.31     -0.31    1000/1001     Range#each
                     0.04      0.01      0.03    1000/1000     Kernel#puts
                     0.00      0.00      0.00    2662/2662     Fixnum#to_s
                     0.00      0.00      0.00    1000/1000     Fixnum#-
                     0.07      0.07      0.00  79022/79022     Fixnum#==
                     0.00      0.00      0.00     831/831      Fixnum#/
                     0.05      0.05      0.00  78022/78022     Fixnum#%
--------------------------------------------------------------------------------
                     0.04      0.01      0.03    1000/1000     Range#each
   8.00%   2.00%     0.04      0.01      0.03         1000     Kernel#puts
                     0.03      0.03      0.00    2000/2000     IO#write
--------------------------------------------------------------------------------
```

ruby-prof



Profile Report

Thread ID	Total Time
537836452	35.48

Thread 537836452

%Total	%Self	Total	Self	Children	Calls	Name
		0.00	21.71	-21.71	10000/10001	Range#each
		35.48	0.15	35.33	1/10001	Kernel#load
100.00%	**61.61%**	**35.48**	**21.86**	**13.62**	**10001**	**Range#each**
		0.01	0.01	0.00	10000/10001	#allocate
		0.00	21.71	-21.71	10000/10001	Range#each
		0.26	0.09	0.17	10000/10000	Kernel#puts
		0.09	0.09	0.00	27540/27540	Fixnum#to_s
		0.00	0.00	0.00	10000/10000	Fixnum#-
		6.65	6.65	0.00	5785223/5785223	Fixnum#==
		0.00	0.00	0.00	8770/8770	Fixnum#/
		6.61	6.61	0.00	5775223/5775223	Fixnum#%
		0.26	0.09	0.17	10000/10000	Range#each
0.73%	**0.25%**	**0.26**	**0.09**	**0.17**	**10000**	**Kernel#puts**
		0.17	0.17	0.00	20000/20000	IO#write

benchmark

bm -- simple benchmark

bmbm -- benchmark with rehearsal

benchmark

```
require 'benchmark'
n = 10_000
Benchmark.bmbm(10) do |x|
  x.report('naive:') {
    for num in 1..n do
      is_prime = 1
      for x in 2..(num - 1) do
        if (num % x == 0)
          is_prime = x
          break
        end
      end
    end
  }
  x.report('sqrt:') {
    for num in 1..n do
      is_prime = 1
      for x in 2..(Math.sqrt(num)) do
        if (num % x == 0)
          is_prime = x
          break
        end
      end
    end
  }
end
```

benchmark

```
$ ruby benchmark_primes.rb
Rehearsal ----------------------------------------------
naive:      2.950000   0.000000   2.950000 (  2.959115)
sqrt:       0.170000   0.000000   0.170000 (  0.165972)
------------------------------------- total: 3.120000sec

                user     system      total        real
naive:      2.750000   0.000000   2.750000 (  2.754461)
sqrt:       0.170000   0.000000   0.170000 (  0.166599)
```

benchmark

```
require 'benchmark'

n = 1_000_000

Benchmark.bm(15) do |x|
  x.report("times:")    { n.times do ; a = "1"; end }
  x.report("upto:")      { 1.upto(n) do ; a = "1"; end }
  x.report("for loop:") { for i in 1..n; a = "1"; end }
end
```

benchmark

```
bash$ ruby bench_loops.rb
                    user     system      total        real
times:          0.520000   0.000000   0.520000 (  0.524952)
upto:           0.330000   0.000000   0.330000 (  0.339263)
for loop:       0.290000   0.000000   0.290000 (  0.295294)
```

benchmark

```
require 'benchmark'

n = 1_000_000

Benchmark.bmbm(15) do |x|
  x.report("for loop:") { for i in 1..n; a = "1"; end }
  x.report("times:") { n.times do ; a = "1"; end }
  x.report("upto:") { 1.upto(n) do ; a = "1"; end }
end
```

benchmark

```
bash$ ruby bench_loops.rb
Rehearsal ----------------------------------------------------
for loop:        0.480000   0.000000   0.480000 (  0.520487)
times:           0.340000   0.000000   0.340000 (  0.330952)
upto:            0.330000   0.000000   0.330000 (  0.331707)
------------------------------------------- total: 1.150000sec

                     user     system      total        real
for loop:        0.290000   0.000000   0.290000 (  0.291192)
times:           0.330000   0.000000   0.330000 (  0.331896)
upto:            0.330000   0.000000   0.330000 (  0.335952)
```

RubyInline

it's so easy, it's almost like cheating

RubyInline

```
for num in 1..100_000 do
  is_prime = 1
  for x in 2..(num - 1) do
    if (num % x == 0)
      is_prime = x
      break
    end
  end
  if is_prime == 1
    puts "#{num} is a prime number"
  else
    puts "#{num} equals #{is_prime} * #{num/is_prime}"
  end
end
```

RubyInline

```
require "inline"
class Primes
  inline do |builder|
    builder.c '
    int prime(int num) {
      int x;
      for (x = 2; x < (num - 1) ; x++) {
        if (num == 2) { return 1; }
        if (num % x == 0) { return x; }
      }
      return 1;
    }'
  end
end
p = Primes.new
for num in 2..100_000 do
  is_prime = p.prime(num)
  if is_prime == 1
    puts "#{num} is a prime number"
  else
    puts "#{num} equals #{is_prime} * #{num/is_prime}"
  end
end
```

RubyInline

```
bash$  time ruby primes.rb > /dev/null
real    9m47.877s
user    9m16.215s
sys     0m0.888s
```

```
bash$  time ruby cprimes.rb > /dev/null
real    0m13.466s
user    0m11.905s
sys     0m0.032s
```

Wrapping Up

zentest.rubyforge.org

rspec.rubyforge.org

eigenclass.org/rcov

ruby-prof.rubyforge.org

rubyforge.org/projects/rubyinline

?