

flat.txt  
 examples/flat.txt  
 Last Update: Tue Nov 11 17:48:31 -0700 2008

## Flat Profiles

Flat profiles show the total amount of time spent in each method. As an example, here is the output from running `printers_test.rb`.

Thread ID: 21277412

%self	cumulative	total	self	children	calls	self/call	total/call	name
46.34	4.06	8.72	4.06	4.66	501	0.01	0.02	Integer#upto
23.89	6.16	2.09	2.09	0.00	61	0.03	0.03	Kernel.sleep
15.12	7.48	1.33	1.33	0.00	250862	0.00	0.00	Fixnum#%
14.13	8.72	1.24	1.24	0.00	250862	0.00	0.00	Fixnum#==
0.18	8.74	0.02	0.02	0.00	1	0.02	0.02	Array#each_index
0.17	8.75	6.64	0.01	6.63	500	0.00	0.01	Object#is_prime
0.17	8.77	6.66	0.01	6.64	1	0.01	6.66	Array#select
0.00	8.77	0.00	0.00	0.00	501	0.00	0.00	Fixnum#-
0.00	8.77	0.00	0.00	0.00	1	0.00	0.00	Array#first
0.00	8.77	0.00	0.00	0.00	1	0.00	0.00	Array#length
0.00	8.77	0.00	0.00	0.00	1	0.00	0.00	Array#initialize
0.00	8.77	8.77	0.00	8.77	1	0.00	8.77	Object#run_primes
0.00	8.77	0.00	0.00	0.00	1	0.00	0.00	Integer#to_int
0.00	8.77	6.66	0.00	6.66	1	0.00	6.66	Object#find_primes
0.00	8.77	2.09	0.00	2.09	1	0.00	2.09	Object#find_largest
0.00	8.77	0.02	0.00	0.02	1	0.00	0.02	Object#make_random_array
0.00	8.77	0.00	0.00	0.00	1	0.00	0.00	Class#new
0.00	8.77	0.00	0.00	0.00	500	0.00	0.00	Array#[ ]=
0.00	8.77	0.00	0.00	0.00	61	0.00	0.00	Fixnum#>
0.00	8.77	0.00	0.00	0.00	61	0.00	0.00	Array#[ ]
0.00	8.77	8.77	0.00	8.77	1	0.00	8.77	#toplevel
0.00	8.77	0.00	0.00	0.00	500	0.00	0.00	Kernel.rand

All values are in seconds.

The columns are:

%self	- The percentage of time spent in this method, derived from <code>self_time/total_time</code>
cumulative	- The sum of the time spent in this method and all the methods listed above it.
total	- The time spent in this method and its children.
self	- The time spent in this method.
children	- The time spent in this method's children.
calls	- The number of times this method was called.
self/call	- The average time spent per call in this method.
total/call	- The average time spent per call in this method and its children.
name	- The name of the method.

Methods are sorted based on %self, therefore the methods that execute the longest are listed first.

The interpretation of method names is:

- toplevel - The root method that calls all other methods
- MyObject#test - An instance method “test” of the class “MyObject”
- test - The <> characters indicate a singleton method on a singleton class.

For example, we can see that `Integer#upto` took the most time, 4.06 seconds. An additional 4.66 seconds were spent in its children, for a total time of 8.72 seconds.

[Hanna RDoc template](#) hand-crafted by [Mislav](#)