Ruby master - Feature #5011

Add Perl 6-like main method definition support: accept command-line arguments as method arguments

07/10/2011 08:11 AM - rosenfeld (Rodrigo Rosenfeld Rosas)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>matz (Yukihiro Matsumoto)</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
</tbody>
</table>

Description
Take a look at this:

http://perlgeek.de/en/article/5-to-6#post_14

I'm not really sure how this would look like in Ruby, but I like the idea of defining a main method could extract their params from command-line params.

This would also be an alternative for issue #4824:


History

#1 - 10/18/2011 09:16 AM - naruse (Yui NARUSE)
- Project changed from Ruby master to CommonRuby
- Target version deleted (3.0)

#2 - 10/23/2011 05:21 PM - naruse (Yui NARUSE)
- Project changed from CommonRuby to Ruby master

#3 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)
- Status changed from Open to Assigned

#4 - 06/08/2012 03:09 AM - rosenfeld (Rodrigo Rosenfeld Rosas)
- File feature-5011.odp added

#5 - 06/08/2012 08:30 AM - mame (Yusuke Endoh)

Received.

Your proposal is interesting, but slightly vague.
I understand motivation, but not the detailed behavior.

If you tolerate the details, I guess you can do it with gem, without modifying ruby interpreter.

[load_main.rb]

```ruby
at_exit do
  if respond_to?(:main, true) && method(:main).source_location[0] == $0
    options = []
    # parse command line options...
    main(ARGF, **options)
  end
end
```

[bar.rb]

```ruby
def main(input)
p :bar
end
```

[foo1.rb]
require "bar"
def main(input, opt)
p :foo1
end

[foo2.rb]
require "bar"

$ ruby -rload_main foo1.rb
:foo
$ ruby -rload_main foo2.rb

--
Yusuke Endoh mame@tsg.ne.jp

#6 - 06/08/2012 09:58 PM - rosenfeld (Rodrigo Rosenfeld Rosas)
Not sure if it is that simple as I would need some way of extracting the argument names and default values from the method. How can I do that?

Anyway, this certainly requires much more discussion about the details on how to design that. I'm more concerned with the overall idea. I don't really care about the exact details on how this would work as long as parsing arguments could be made much easier, like in Perl 6...

I'll try to make some experiment this weekend by parsing the source line in some simple way for demonstrating purposes of the overall idea... Maybe you could provide me some implementation for querying the arguments and default values in ruby-head, than I could install it with RVM and try it without any hack like parsing the source itself...

#7 - 07/23/2012 10:17 PM - mame (Yusuke Endoh)
- Status changed from Assigned to Rejected

Rodrigo Rosenfeld Rosas,

Sorry but this proposal was rejected at the developer meeting (7/21).

Matz said that it was not a bad idea to improve optparse, but that this proposal was not in the right direction.

--
Yusuke Endoh mame@tsg.ne.jp

#8 - 07/24/2012 12:47 AM - rosenfeld (Rodrigo Rosenfeld Rosas)
Great, I'll love to see when the right direction appear ;)

Files
feature-5011.odp 19.1 KB 06/08/2012 rosenfeld (Rodrigo Rosenfeld Rosas)