

It doesn't matter how easy a language is to use, it usually contains some bugs if it is more than a few lines long. To help deal with bugs, the standard distribution of Ruby includes a debugger.

In order to start the Ruby debugger, load the debug library using the command-line option `-r debug`. The debugger stops before the first line of executable code and asks for the input of user commands.

Usage Syntax:

Here is the usage syntax to use ruby debugger:

```
$ ruby -r debug filename[, ...]
```

Ruby Debugger Commands:

Here is a complete list of commands, which you can use while debugging your program. Here, it is not necessary to use complete keyword to give a command, part given inside [...] is option.

SN	Command with Description
1	b[reak] [< file class>:]< line method> Sets breakpoint to some position. Breakpoint is a place where you want to pause program execution for debugging purpose.
2	wat[ch] expression Sets watchpoints
3	cat[ch] exception off Sets catchpoint to an exception.
4	b[reak] Displays breakpoints and watchpoints
5	del[ete] [n] Deletes breakpoints
6	disp[lay] expression Displays value of <i>expression</i>
7	undisp[lay] [n] Removes display of n
8	c[ont] Continues execution

- 9 **s[tep] [n]**
Executes next n lines stepping into methods
- 10 **n[ext] [n]**
Executes next n lines stepping over methods
- 11 **w[here]**
Displays stack frame
- 12 **f[rame]**
Synonym for where
- 13 **l[ist][<-| n- m>]**
Displays source lines from n to m
- 14 **up [n]**
Moves up n levels in the stack frame
- 15 **down [n]**
Moves down n levels in the stack frame
- 16 **fin[ish]**
Finishes execution of the current method
- 17 **tr[ace] [on|off]**
Toggles trace mode on and off
- 18 **q[uit]**
Exits debugger
- 19 **v[ar] g[lobal]**
Displays global variables
- 20 **v[ar] l[ocal]**
Displays local variables
- 21 **v[ar] i[instance] *object***
Displays instance variables of *object*
- 22 **v[ar] c[onst] *object***
Displays constants of *object*

- 23 **m[method] i[instance] object**
 Displays instance methods of *object*

- 24 **m[method] class| module**
 Displays instance methods of the *class or module*

- 25 **th[read] l[ist]**
 Displays threads

- 26 **th[read] c[ur[rent]]**
 Displays current thread

- 27 **th[read] n**
 Stops specified thread

- 28 **th[read] stop >**
 Synonym for th[read] n

- 29 **th[read] c[ur[rent]] n>**
 Synonym for th[read] n

- 30 **th[read] resume >**
 Resumes thread n

- 31 **p expression**
 Evaluates the *expression*

- 32 **h[elp]**
 Displays help message

- 33 **everything else**
 Evaluates.

Example:

Consider the following file *hello.rb*, which needs to be debugged:

```
#!/usr/bin/env ruby
class Hello
  def initialize( hello )
    @hello = hello
  end
  def hello
    @hello
  end
end
```

```
salute = Hello.new( "Hello, Mac!" )  
puts salute.hello
```

Here is one interactive session captured. Given commands are written in bold:

```
[root@ruby]# ruby -r debug hello.rb  
Debug.rb  
Emacs support available.  
  
hello.rb:3:class Hello  
(rdb:1) v 1  
    salute => nil  
(rdb:1) b 10  
Set breakpoint 1 at hello.rb:10  
(rdb:1) c  
Hello, Mac!  
[root@ruby]#
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js