

## Ruby trunk - Feature #13849

### Show --dump options in help

08/30/2017 05:14 AM - sonots (Naotoshi Seo)

<b>Status:</b>	Closed
<b>Priority:</b>	Normal
<b>Assignee:</b>	sonots (Naotoshi Seo)
<b>Target version:</b>	Next Major
<b>Description</b>	
Current help does not show any helps for --dump options such as --dump=insns. I want to get them.	
<pre>\$ ./ruby --help Usage: ./ruby [switches] [--] [programfile] [arguments]   -0[octal]          specify record separator (\0, if no argument)   -a                autosplit mode with -n or -p (splits \$_ into \$F)   -c                check syntax only   -Cdirectory       cd to directory before executing your script   -d, --debug       set debugging flags (set \$DEBUG to true)   -e 'command'      one line of script. Several -e's allowed. Omit [programfile]   -Eex[:in], --encoding=ex[:in]                     specify the default external and internal character encodings   -Fpattern         split() pattern for autosplit (-a)   -i[extension]    edit ARGV files in place (make backup if extension supplied)   -Idirectory       specify \$LOAD_PATH directory (may be used more than once)   -l               enable line ending processing   -n               assume 'while gets(); ... end' loop around your script   -p               assume loop like -n but print line also like sed   -rlibrary         require the library before executing your script   -s               enable some switch parsing for switches after script name   -S               look for the script using PATH environment variable   -T[level=1]      turn on tainting checks   -v, --verbose    print version number, then turn on verbose mode   -w               turn warnings on for your script   -W[level=2]      set warning level; 0=silence, 1=medium, 2=verbose   -x[directory]    strip off text before #!ruby line and perhaps cd to directory   --copyright      print the copyright   --enable=feature[,...], --disable=feature[,...]                     enable or disable features   --external-encoding=encoding, --internal-encoding=encoding                     specify the default external or internal character encoding   --version        print the version   --help          show this message, -h for short message  Features:   gems              rubygems (default: enabled)   did_you_mean      did_you_mean (default: enabled)   rubyopt           RUBYOPT environment variable (default: enabled)   frozen-string-literal                     freeze all string literals (default: disabled)</pre>	

### History

#### #1 - 08/30/2017 05:14 AM - sonots (Naotoshi Seo)

- Target version set to 2.6

#### #2 - 08/30/2017 05:15 AM - sonots (Naotoshi Seo)

ruby 2.4.1 (actually, trunk)

#### #3 - 08/30/2017 05:57 AM - sonots (Naotoshi Seo)

version, copyright, usage, help, yydebug, syntax, parsetree, parsetree\_with\_comment, insns are available as an argument of --dump option. I now wonder which I should support. I want yydebug, parsetree, and insns (parsetree\_with\_comment is too long to show in help?)

Any opinions?

```
diff --git a/ruby.c b/ruby.c
index f2f3c86e09..808a2bdce6 100644
--- a/ruby.c
+++ b/ruby.c
@@ -191,7 +191,7 @@ static struct {
     static void
     show_usage_line(const char *str, unsigned int namelen, unsigned int secondlen, int help)
     {
-         const unsigned int w = 16;
+         const unsigned int w = 17;
         const int wrap = help && namelen + secondlen - 2 > w;
         printf("  %.*s%-*.s%-*s%s\n", namelen-1, str,
                (wrap ? 0 : w - namelen + 1),
@@ -238,6 +238,9 @@ usage(const char *name, int help)
         M("-W[level=2]", "", "set warning level; 0=silence, 1=medium, 2=verbose"),
         M("-x[directory]", "", "strip off text before #!ruby line and perhaps cd to direct
ory"),
         M("-h", "", "show this message, --help for more info"),
+         M("--dump=insns", "", "dump YARV instructions"),
+         M("--dump=yydebug", "", "dump yydebug of yacc parser generator"),
+         M("--dump=parsetree", "", "dump parsetree"),
     };
     static const struct message help_msg[] = {
         M("--copyright", "", "print the copyright"),
     }
};
```

#### #4 - 08/30/2017 08:02 AM - naruse (Yui NARUSE)

sonots (Naotoshi Seo) wrote:

version, copyright, usage, help, yydebug, syntax, parsetree, parsetree\_with\_comment, insns are available as an argument of --dump option. I now wonder which I should support. I want yydebug, parsetree, and insns (parsetree\_with\_comment is too long to show in help?) Any opinions?

Write detailed description elsewhere, and then just show the link in --help.

#### #5 - 08/30/2017 08:08 AM - sonots (Naotoshi Seo)

- Target version changed from 2.6 to Next Major

elsewhere ... where are you supposed?

#### #6 - 08/30/2017 08:19 AM - nobu (Nobuyoshi Nakada)

Or add the message "try --dump=? to show valid arguments".

#### #7 - 08/30/2017 09:37 AM - shevegen (Robert A. Heiler)

+1

The "--dump=?" may also work. I do however think that Naotoshi Sao's suggestion, although more verbose, may be best, since people can simply read what these options do in the short description, such as "dump YARV instructions". (I actually did not know that this option exists altogether.)

#### #8 - 08/30/2017 09:47 AM - sonots (Naotoshi Seo)

I feel splitting help to --dump=? is nice idea.

By the way, the current behavior is as:

bash

```
$ ./ruby --dump=?
./ruby: warning: don't know how to dump `?',
./ruby: warning: but only [version, copyright, usage, help, yydebug, syntax, parsetree, parsetree_with_comment, insns].
```

zsh (without setopt nonomatch)

```
$ ./ruby --dump=?
zsh: no matches found: --dump=?
```

```
$ ./ruby '--dump=?'
./ruby: warning: don't know how to dump `?',
./ruby: warning: but only [version, copyright, usage, help, yydebug, syntax, parsetree, parsetree_with_comment, insns].
```

Since --dump=help is already used, I will pick --dump=?, but please note the zsh behavior.

#### #9 - 08/30/2017 12:09 PM - znz (Kazuhiro NISHIYAMA)

bash also expand glob if files exist.

And bash with shopt -s failglob can raise error like zsh.

So I think users should not depend such behavior.

```
$ touch ./--dump=a
$ ruby --dump=?
ruby: warning: don't know how to dump `a',
ruby: warning: but only [version, copyright, usage, yydebug, syntax, parsetree, parsetree_with_comment, insns]
.
$ rm ./--dump=a
$ shopt -s failglob
$ ruby --dump=?
-bash: 000000 : --dump=?
```

#### #10 - 08/30/2017 04:52 PM - sonots (Naotoshi Seo)

It sounds --dump=? is troublesome, and also it is not easy to implement.

It looks --dump=help and --help is same. Can I drop the current --dump=help, and use --dump=help instead of --dump=? ?

#### #11 - 08/31/2017 05:41 AM - sonots (Naotoshi Seo)

I removed old --dump=help, and --dump=usage, and changed --dump=help to show help messages of --dump.

<https://github.com/ruby/ruby/pull/1688>

```
$ ./ruby --help
Usage: ./ruby [switches] [--] [programfile] [arguments]
...
--dump=target[,...]
    try --dump=help to show help messages of --dump
```

(this is not shown with short help, that is, -h)

```
$ ./ruby --dump=help
Usage: ./ruby [switches] [--] [programfile] [arguments]
--dump=insns      dump instruction sequences
--dump=yydebug    dump yydebug of yacc parser generator
--dump=parsetree, --dump=parsetree_with_comment
                  dump AST or AST with comments.
```

#### #12 - 08/31/2017 06:40 AM - nobu (Nobuyoshi Nakada)

What about --help=dump (--help=feature and so on)?

#### #13 - 08/31/2017 07:44 AM - sonots (Naotoshi Seo)

Changed as:

```
$ ./ruby --help
Usage: ./ruby [switches] [--] [programfile] [arguments]
-0[octal]        specify record separator (\0, if no argument)
-a              autosplit mode with -n or -p (splits $_ into $F)
-c              check syntax only
-Cdirectory      cd to directory before executing your script
-d, --debug      set debugging flags (set $DEBUG to true)
-e 'command'     one line of script. Several -e's allowed. Omit [programfile]
-Eex[:in], --encoding=ex[:in]
                  specify the default external and internal character encodings
-Fpattern        split() pattern for autosplit (-a)
-i[extension]    edit ARGV files in place (make backup if extension supplied)
-Idirectory      specify $LOAD_PATH directory (may be used more than once)
-l              enable line ending processing
-n              assume 'while gets(); ... end' loop around your script
-p              assume loop like -n but print line also like sed
-rlibrary        require the library before executing your script
-s              enable some switch parsing for switches after script name
-S              look for the script using PATH environment variable
```

```

-T[level=1]      turn on tainting checks
-v, --verbose   print version number, then turn on verbose mode
-w             turn warnings on for your script
-W[level=2]     set warning level; 0=silence, 1=medium, 2=verbose
-x[directory]  strip off text before #!ruby line and perhaps cd to directory
--copyright    print the copyright
+ --dump=target[,...]
+             try --help=dump to show help messages of --dump
--enable=feature[,...], --disable=feature[,...]
              enable or disable features
--external-encoding=encoding, --internal-encoding=encoding
              specify the default external or internal character encoding
--version      print the version
--help        show this message, -h for short message
Features:
  gems          rubygems (default: enabled)
  did_you_mean  did_you_mean (default: enabled)
  rubyopt       RUBYOPT environment variable (default: enabled)
  frozen-string-literal
               freeze all string literals (default: disabled)

```

```

$ ./ruby --help=dump
Usage: ./ruby [switches] [--] [programfile] [arguments]
--dump=insns      dump instruction sequences
--dump=yydebug    dump yydebug of yacc parser generator
--dump=parsetree, --dump=parsetree_with_comment
                 dump AST or AST with comments.

```

```

$ ./ruby --help=xxx
./ruby: warning: don't know how to show help `xxx',
./ruby: warning: but only [dump].

```

Now, --dump=help is kept.

#### #14 - 09/01/2017 12:46 AM - sonots (Naotoshi Seo)

Sorry for raising a discussion again.

I noticed that Features list are shown bottom at --help.

```

$ ruby --help
...
--enable=feature[,...], --disable=feature[,...]
              enable or disable features
--external-encoding=encoding, --internal-encoding=encoding
              specify the default external or internal character encoding
--version      print the version
--help        show this message, -h for short message
Features:
  gems          rubygems (default: enabled)
  did_you_mean  did_you_mean (default: enabled)
  rubyopt       RUBYOPT environment variable (default: enabled)
  frozen-string-literal
               freeze all string literals (default: disabled)

```

I now feel showing dump lists with --help=dump does not have a consistency.  
I have two idea.

#### (1) Show dump list at bottom as features list

```

$ ruby --help
Usage: ruby [switches] [--] [programfile] [arguments]
-0[octal]       specify record separator (\0, if no argument)
-a             autosplit mode with -n or -p (splits $_ into $F)
-c             check syntax only
-Cdirectory    cd to directory before executing your script
-d, --debug     set debugging flags (set $DEBUG to true)
-e 'command'   one line of script. Several -e's allowed. Omit [programfile]
-Eex[:in], --encoding=ex[:in]
               specify the default external and internal character encodings
-Fpattern      split() pattern for autosplit (-a)
-i[extension]  edit ARGV files in place (make backup if extension supplied)
-Idirectory    specify $LOAD_PATH directory (may be used more than once)
-l             enable line ending processing
-n             assume 'while gets(); ... end' loop around your script
-p            assume loop like -n but print line also like sed

```

```

-rlibrary      require the library before executing your script
-s            enable some switch parsing for switches after script name
-S           look for the script using PATH environment variable
-T[level=1]   turn on tainting checks
-v, --verbose print version number, then turn on verbose mode
-w          turn warnings on for your script
-W[level=2]  set warning level; 0=silence, 1=medium, 2=verbose
-x[directory] strip off text before #!ruby line and perhaps cd to directory
--copyright  print the copyright
--dump={insns|parsetree|...}[,...]
            dump debug information. see below for available dump list
--enable={gems|rubyopt|...}[,...], --disable={gems|rubyopt|...}[,...]
            enable or disable features. see below for available features
--external-encoding=encoding, --internal-encoding=encoding
            specify the default external or internal character encoding
--version    print the version
--help      show this message, -h for short message

```

Dump List:

```

insns          instruction sequences
yydebug       yydebug of yacc parser generator
parsetree     AST
parsetree_with_comment
              AST with comments

```

Features:

```

gems          rubygems (default: enabled)
did_you_mean  did_you_mean (default: enabled)
rubyopt       RUBYOPT environment variable (default: enabled)
frozen-string-literal
              freeze all string literals (default: disabled)

```

(2) Show dump list and features list by --help=dump and --help=feature respectively

```
$ ./ruby --help
```

```

Usage: ruby [switches] [--] [programfile] [arguments]
-0[octal]    specify record separator (\0, if no argument)
-a          autosplit mode with -n or -p (splits $_ into $F)
-c          check syntax only
-Cdirectory  cd to directory before executing your script
-d, --debug  set debugging flags (set $DEBUG to true)
-e 'command' one line of script. Several -e's allowed. Omit [programfile]
-Eex[:in], --encoding=ex[:in]
            specify the default external and internal character encodings
-Fpattern    split() pattern for autosplit (-a)
-i[extension] edit ARGV files in place (make backup if extension supplied)
-Idirectory  specify $LOAD_PATH directory (may be used more than once)
-l          enable line ending processing
-n          assume 'while gets(); ... end' loop around your script
-p          assume loop like -n but print line also like sed
-rlibrary    require the library before executing your script
-s          enable some switch parsing for switches after script name
-S          look for the script using PATH environment variable
-T[level=1]  turn on tainting checks
-v, --verbose print version number, then turn on verbose mode
-w          turn warnings on for your script
-W[level=2]  set warning level; 0=silence, 1=medium, 2=verbose
-x[directory] strip off text before #!ruby line and perhaps cd to directory
--copyright  print the copyright
--dump={insns|parsetree|...}[,...]
            dump debug information. see --help=dump for details
--enable={gems|rubyopt|...}[,...], --disable={gems|rubyopt|...}[,...]
            enable or disable features. see --help=feature for details
--external-encoding=encoding, --internal-encoding=encoding
            specify the default external or internal character encoding
--version    print the version
--help      show this message, -h for short message

```

```
$ ./ruby --help=dump
```

```

Usage: ./ruby [switches] [--] [programfile] [arguments]
--dump={insns|parsetree|...}[,...]
            dump debug information. followings are available dump list

```

Dump List:

```

insns          instruction sequences
yydebug       yydebug of yacc parser generator
parsetree     AST
parsetree_with_comment

```

AST with comments

```
$ ./ruby --help=feature
Usage: ./ruby [switches] [--] [programfile] [arguments]
  --enable={gems|rubyopt|...}[,...], --disable={gems|rubyopt|...}[,...]
      enable or disable features. followings are available features
Features:
  gems          rubygems (default: enabled)
  did_you_mean  did_you_mean (default: enabled)
  rubyopt       RUBYOPT environment variable (default: enabled)
  frozen-string-literal
                freeze all string literals (default: disabled)
```

I feel (2) is better based on the matz principle to keep help as compact as possible.

**#15 - 09/01/2017 02:31 AM - nobu (Nobuyoshi Nakada)**

sonots (Naotoshi Seo) wrote:

I feel (2) is better based on the matz principle to keep help as compact as possible.

Compact help is -h, --help doesn't need to be.

**#16 - 09/01/2017 02:52 AM - sonots (Naotoshi Seo)**

!! So, (1) was fine.

Hmm, I feel showing everything by one command makes easier to find options, so I will change implementation to (1).

**#17 - 09/01/2017 03:31 AM - sonots (Naotoshi Seo)**

Changed implementation to (1).

<https://github.com/ruby/ruby/pull/1688>

<https://github.com/sonots/ruby/blob/d762ee5c1648da5a2b82c19c106fcc219b92df61/ruby.c>

**#18 - 09/20/2017 05:05 AM - sonots (Naotoshi Seo)**

fixed via <https://bugs.ruby-lang.org/projects/ruby-trunk/repository/revisions/59973>

**#19 - 09/20/2017 05:05 AM - sonots (Naotoshi Seo)**

- Status changed from Open to Closed