Ruby master - Feature #11181

Add a line directive to Ruby

05/26/2015 07:05 AM - gam3 (Allen Morris)

Status:	Open
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	

Description

Add a **line directive** to Ruby

```
  #line {nn} ["filename"]
```

This is done by creating a array of filenames and using the upper bits of the line_number to determine the current filename. The original filename is in position 0.

An extra node is added by the parser that informs the compiler of the filenames so the backtrace code can also use the correct file names.

The __**LINE**__ and __**FILE**__ constants are updated and compile time warnings are also effected.

There is a pull request at https://github.com/ruby/ruby/pull/911

The patch does not have any affect on current programs unless a line matching '#\s*line \d+(\s+"(*.)")?\s*$' is found in the ruby source code.

More tests need to be written before this change sould be applied.

Use case:

This is helpful for debugging any generated code but is particularlly helpful for literate programming using Noweb.

History

#1 - 05/26/2015 07:56 AM - gam3 (Allen Morris)

using 'line' as the key word causes the 'test-rubyspec' tests to fail as they contain:

```
# line 3
```

in the __**LINE**__ test.

Using 'pline' for the moment.

#2 - 05/26/2015 09:30 PM - gam3 (Allen Morris)

Reason why 'eval "string", b, file, line' can't be used in place of line directive.

Use case: Input file looks like

```
file: alice
<alice>=
   puts "'#{__FILE__} #{__LINE__}'"  # alice line 2
   <bob>
   puts "'#{__FILE__} #{__LINE__}'"  # alice line 4

file: bob
<bob>=
   puts "'#{__FILE__} #{__LINE__}'"  # bob line 2
   <charlie>
   puts "'#{__FILE__} #{__LINE__}'"  # bob line 4

file: charly
<charlie>=
   puts "'#{__FILE__} #{__LINE__}'"  # charlie line 2
```

Some buildy thingy is run on this that reads all three files and used "<alice>' as the start and generate something like:

```
eval <<-'RUBY', nil, 'alice', 1
  puts "'#{__FILE__} #{__LINE__}' = 'alice 2'"
  eval <<-'RUBY2', nil, 'bob', 1
puts "'#{__FILE__} #{__LINE__}' = 'bob 2'"
eval <<-'RUBY3', nil, 'charlie', 1
  puts "'#{__FILE__} #{__LINE__}' = 'charlie 2'"
RUBY3
puts "'#{__FILE__} #{__LINE__}' != 'bob 4'"  # ERROR: displays as 'bob 5'
  RUBY2
  puts "'#{__FILE__} #{__LINE__}' != 'alice 4'"   # ERROR: displays as 'alice 9'
RUBY
```

Then there is the less likeley case where you have non-complete 'code blocks'. This would be where we (for some odd reseason) wanted to have the start of a block in one file and the end in another:

```
 file a = "class Bob"
 file b = "end # class Bob"
```

```
  #line 1 "a"
  class bob
  #line 1 "b"
  end # class Bob
```

#3 - 05/28/2015 05:34 PM - gam3 (Allen Morris)

I have changed the code to use the **magic comment** feature. To use the directive you now use:

```
# -*- line: filename 1000 -*-
```

or

```
# -*- line: 1000 -*-
```

Where the filename is optional and the line number is greater than zero.

The filename if included is parsed in the same way as the arguemnt to the 'coding magic comment'. The line number is parsed as a base 10 unsigned long.

This is all done in a funtion called magic_line in parse.y.

#4 - 06/21/2015 07:13 AM - shevegen (Robert A. Heiler)

Today on IRC someone else also wanted such a feature:

sphex> hey. does ruby support something like "#line" directives?
sphex> I mean a cpp-like directive to let the parser know which file name / line number the input came from.
sphex> this is for generated code. I'd like to change the parser's idea of the current line number.
sphex> so that it reports the original line number from which the parsed code was generated in stack traces, syntax error exceptions, etc. C and Perl (and prolly others) change their idea of the current line number when they encounter "#line N" comments, which allows just that. (yacc/lex use that, for example)
sphex> someone else already asked for this feature: https://bugs.ruby-lang.org/issues/11181

#5 - 10/22/2015 05:55 PM - gam3 (Allen Morris)

- Description updated

The pull request is now at https://github.com/ruby/ruby/pull/911

#6 - 11/24/2015 06:20 AM - duerst (Martin Dürst)

- Assignee changed from ruby-core to matz (Yukihiro Matsumoto)

I have changed the assignee from ruby-core (all committers) to Matz to make sure this doesn't show up in everybody's todo list.

#7 - 01/26/2016 02:04 AM - gam3 (Allen Morris)

Pull request up to date as of 2016-01-25