

Ruby - Feature #14683

IRB with Ripper

04/12/2018 03:06 PM - aycabta (aycabta .)

Status:	Closed
Priority:	Normal
Assignee:	keiju (Keiju Ishitsuka)
Target version:	
Description <p>I replaced lexical analyzer with Ripper. It's important for supporting new syntax.</p> <p>I explain what I did.</p> <h3>Replace with Ripper</h3> <p>I talked with @matz (Yukihiro Matsumoto) about it and I decided to use mirb of mruby as a reference. https://github.com/mruby/mruby/blob/1905091634a6a2925c911484434448e568330626/mrbgems/mruby-bin-mirb/tools/mirb.c#L118-L217</p> <p>The mirb uses compile error messages and token information such as token type and lex_state for that whether or not inputted code continues(and PROMPT_N). I used RubyVM::InstructionSequence#compile and Ripper for the same action.</p> <p>ref. IRB#each_top_level_statement, IRB#lex and IRB#check_code_block</p> <h3>Continue(PROMPT_N), literal type(PROMPT_S), nesting level(%NNi)</h3> <p>IRB needs statuses of code snippets. @nobu (Nobuyoshi Nakada) implemented it as Ripper's features at irb-ripper branch on GitHub. https://github.com/nobu/ruby/commits/feature/irb-ripper</p> <p>I implemented by pure Ruby because I heard what @nobu (Nobuyoshi Nakada) thinks pure Ruby implementation for it. I think that it's the best way for that Ripper keeps simple.</p> <p>ref. IRB#process_continue, IRB#process_nesting_level, IRB#check_string_literal and IRB#process_literal_type</p> <h3>Erase --irb_debug option</h3> <p>Because this is just for pure Ruby lexical analyzer.</p>	

History

#1 - 04/12/2018 04:50 PM - aycabta (aycabta .)

ref. <https://github.com/ruby/rdoc/pull/512>

#2 - 04/13/2018 12:49 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

- Assignee set to keiju (Keiju Ishitsuka)

#3 - 04/19/2018 04:17 PM - keiju (Keiju Ishitsuka)

I am developing Reirb which is new version of irb, now.
Reirb is a reborn irb, and new implementation for irb,
I would like to replacing irb with reirb(as irb).

Therefore, I want to pending big changes until RubyKaigi 2018 is over.

#4 - 05/05/2018 10:53 AM - aycabta (aycabta .)

I discussed this issue with [@hsbt \(Hiroshi SHIBATA\)](#), [@matsuda \(Akira Matsuda\)](#), and [@matz \(Yukihiro Matsumoto\)](#) .

In the present day, IRB needs compatibility completely.

- Bundler generates template files of new RubyGems with example console as "bin/console" for development. It uses IRB. Developers of gems can edit the file because it's just template file of customized IRB console.
- All Ruby on Rails applications have "rails console" command.
- Many users customize .irbrc deeply includes me.

Ruby needs a simple REPL as IRB with perfect compatibility. And [@matz \(Yukihiro Matsumoto\)](#) said that IRB should not be replaced without compatibility in the discussion with me. IRB is still contributing greatly to Ruby's system and we must continue to maintain IRB's compatibility in the future. I think that you'd better develop new Ruby REPL(or shell) without the weighty responsibility of heavy compatibility.

If you can provide perfect compatibility for the new version of IRB, please publish the source code of it because I want to contribute. If not, please discuss this issue without the new version of IRB.

At RubyKaigi 2018, I'll talk about IRB's new features.

<http://rubykaigi.org/2018/presentations/aycabta.html>

So I need a source code of IRB that I can contribute.

[@keiju \(Keiju Ishitsuka\)](#) san, I watched all your presentations of Ruby shell in RubyKaigi 2014, 2015, and 2017. I'm looking forward to living on your Ruby shell.

#5 - 05/28/2019 12:06 AM - aycabta (aycabta .)

- Status changed from Assigned to Closed

Files

irb.patch	51.8 KB	04/12/2018	aycabta (aycabta .)
-----------	---------	------------	---------------------