Ruby master - Bug #14918

Use Reline for fallback of ext/readline

07/18/2018 04:48 AM - aycabta (aycabta .)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>aycabta (aycabta .)</td>
</tr>
<tr>
<td>Target version:</td>
<td>ruby -v:</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.5: UNKNOWN, 2.6: UNKNOWN</td>
</tr>
</tbody>
</table>

**Description**

Ruby loses readline standard library when the system doesn't have libreadline during Ruby installation, At [https://bugs.ruby-lang.org/issues/11084](https://bugs.ruby-lang.org/issues/11084), hsbt (Hiroshi SHIBATA) proposed pure Ruby implementation. But I think that rb-readline is not good for Ruby standard library and I described it at [http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/86213](http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/86213). The important reason for it personally is that rb-readline doesn't have vi mode.

So I'm implementing Reline for a new standard library as GNU Readline and existing ext/readline compatible. I can maintain it. [https://github.com/aycabta/reline](https://github.com/aycabta/reline)

I don't think that Reline should replace ext/readline fully, it's just for fallback. But it's important for Ruby.

**Associated revisions**

Revision 67645 - 04/20/2019 08:51 AM - aycabta (aycabta .)

IRB is improved with Reline and RDoc

Reline is a readline stdlib compatible library. It also supports multiline input. IRB is improved with Reline and supports multiline. Besides, supports showing documents when completed.

**History**

#1 - 07/18/2018 05:18 AM - aycabta (aycabta .)
- Backport deleted (2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN)
- Tracker changed from Bug to Feature

#2 - 04/20/2019 11:35 AM - naruse (Yui NARUSE)
- Assignee set to aycabta (aycabta .)
- Status changed from Open to Closed

r67645 introduces this feature.

#3 - 12/25/2019 06:12 PM - josh.cheek (Josh Cheek)
- Backport set to 2.5: UNKNOWN, 2.6: UNKNOWN
- Tracker changed from Feature to Bug
- File Screen Shot 2019-12-25 at 11.47.25 AM.png added

aycabta (aycabta .) wrote:

Ruby loses readline standard library when the system doesn't have libreadline during Ruby installation, At [https://bugs.ruby-lang.org/issues/11084](https://bugs.ruby-lang.org/issues/11084), hsbt (Hiroshi SHIBATA) proposed pure Ruby implementation. But I think that rb-readline is not good for Ruby standard library and I described it at [http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/86213](http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/86213). The important reason for it personally is that rb-readline doesn't have vi mode.

So I'm implementing Reline for a new standard library as GNU Readline and existing ext/readline compatible. I can maintain it. [https://github.com/aycabta/reline](https://github.com/aycabta/reline)

I don't think that Reline should replace ext/readline fully, it's just for fallback. But it's important for Ruby.

Reline does not seem to support bracketed paste mode:
```bash
$ ruby -v
ruby 2.7.0p0 (2019-12-25 revision 647ee6f091) [x86_64-darwin18]

$ # has 2 lines of input
  pbpaste
  "abc" .upcase

$ # readline respects bracketed paste mode, reads in both lines
  ruby -r readline -e 'p Readline.readline "> "'
  > "abc"
  .upcase
  "\"abc\"\n.upcase"

$ # reline does not support bracketed paste mode, reads them in as 2 separate lines
  ruby -r reline -e '2.times { p Reline.readline "> " }'
  > "abc"
  "\"abc\""
  > .upcase
  ".upcase"

$ cat ~/.inputrc | sed '/^$/q'
set enable-bracketed-paste on
set blink-matching-paren on
set colored-completion-prefix on
set colored-stats on
set keyseq-timeout 250
set show-all-if-ambiguous on
set visible-stats on

I also checked one of the other config options (blink-matching-paren) and it blinks with readline, but not reline. Makes me wonder if it isn't reading the ~/.inputrc file (it may be Mac specific).

Reporting it here as the Github project doesn't seem to have an issues section.
```

Files

Screen Shot 2019-12-25 at 11.47.25 AM.png  83.4 KB  12/25/2019  josh.cheek (Josh Cheek)