

Ruby trunk - Feature #13518

Indented multiline comments

04/28/2017 02:47 AM - tscheingeld (Terry Scheingeld)

Status:	Rejected
Priority:	Normal
Assignee:	
Target version:	
Description	
I'd like to submit the idea that multiline comments could be indented. That is, =begin and =end do not have to start at column zero. That would allow for more flexibility in documenting and commenting code.	

History

#1 - 04/28/2017 02:58 AM - tscheingeld (Terry Scheingeld)

- Description updated

#2 - 04/28/2017 06:58 PM - shevegen (Robert A. Heiler)

I have no particular pro or con opinion here (I do not use =begin or =end anyway, only '#') but would this even be possible?

#3 - 04/29/2017 05:39 AM - nobu (Nobuyoshi Nakada)

Possible of course, but I'm not a big fan of this.

```
diff --git i/misc/ruby-mode.el w/misc/ruby-mode.el
index blabd18a9e..5e8a6a1646 100644
--- i/misc/ruby-mode.el
+++ w/misc/ruby-mode.el
@@ -135,7 +135,7 @@
   (concat "[?$/%(){}#\"'`.:]\\\\<<\\\\\\\\[\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\<\\\\("
     ruby-block-beg-re
     "\\\)\\\\_>\\\\|" ruby-block-end-re
-     "\\\|^=begin\\\\|" ruby-here-doc-beg-re)
+     "\\\|^[\t]*=begin\\\\|" ruby-here-doc-beg-re)
   )

(defconst ruby-negative
@@ -671,8 +671,8 @@ Emacs to Ruby."
  (looking-at "\\. [a-zA-Z_0-9]+")
  (looking-at "\\.")
  (goto-char (match-end 0)))
- ((looking-at "^=begin")
- (if (re-search-forward "=end" end t)
+ ((looking-at "^[\t]*=begin")
+ (if (re-search-forward "^[\t]*=end" end t)
    (forward-line 1)
    (setq in-string (match-end 0))
    (goto-char end)))
@@ -954,10 +954,10 @@ An end of a defun is found by moving forward from the beginning of one."
  (cond
    ((looking-at "^\\s *$"))
    ((looking-at "^\\s *#"))
-   ((and (> n 0) (looking-at "^=begin\\>"))
-    (re-search-forward "=end\\>"))
-   ((and (< n 0) (looking-at "^=end\\>"))
-    (re-search-backward "=begin\\>"))
+   ((and (> n 0) (looking-at "^[\t]*=begin\\>"))
+    (re-search-forward "^[\t]*=end\\>"))
+   ((and (< n 0) (looking-at "^[\t]*=end\\>"))
+    (re-search-backward "^[\t]*=begin\\>"))
    (t
     (setq pos (current-indentation))
     (cond
@@ -1410,12 +1410,12 @@ buffer position `limit' or the end of the buffer."
      . ruby-font-lock-syntactic-keywords))))
```

```

- (defun ruby-font-lock-docs (limit)
+ (if (re-search-forward "^=begin\\(\\s \\|$\\)" limit t)
+ (if (re-search-forward "^[ \\t]*=begin\\(\\s \\|$\\)" limit t)
    (let (beg)
      (beginning-of-line)
      (setq beg (point))
      (forward-line 1)
- (if (re-search-forward "^=end\\(\\s \\|$\\)" limit t)
+ (if (re-search-forward "^[ \\t]*=end\\(\\s \\|$\\)" limit t)
      (progn
        (set-match-data (list beg (point)))
        t))))))
@@ -1423,12 +1423,12 @@ buffer position `limit' or the end of the buffer."
- (defun ruby-font-lock-maybe-docs (limit)
+ (let (beg)
    (save-excursion
- (if (and (re-search-backward "=\\(begin\\|end\\)\\(\\s \\|$\\)" nil t)
+ (if (and (re-search-backward "[ \\t]*=\\(begin\\|end\\)\\(\\s \\|$\\)" nil t)
      (string= (match-string 1) "begin"))
      (progn
        (beginning-of-line)
        (setq beg (point))))))
- (if (and beg (and (re-search-forward "=\\(begin\\|end\\)\\(\\s \\|$\\)" nil t)
+ (if (and beg (and (re-search-forward "[ \\t]*=\\(begin\\|end\\)\\(\\s \\|$\\)" nil t)
      (string= (match-string 1) "end")))
      (progn
        (set-match-data (list beg (point)))
diff --git i/parse.y w/parse.y
index 028b89c070..4d3de062c6 100644
--- i/parse.y
+++ w/parse.y
@@ -7851,6 +7851,18 @@ parse_ident(struct parser_params *parser, int c, int cmd_state)
 }

 static int
+parser_indented_bol_p(struct parser_params *parser)
+{
+  const char *p = lex_pbeg;
+  const char *e = lex_p - 1;
+  while (p < e) {
+    if (!ISSPACE(*p)) return FALSE;
+    p++;
+  }
+  return TRUE;
+}
+
+static int
parser_yylex(struct parser_params *parser)
{
  register int c;
@@ -8039,7 +8051,7 @@ parser_yylex(struct parser_params *parser)
  return '!';

  case '=':
-  if (was_bol()) {
+  if (was_bol() || parser_indented_bol_p(parser)) {
    /* skip embedded rd document */
    if (strncmp(lex_p, "begin", 5) == 0 && ISSPACE(lex_p[5])) {
      int first_p = TRUE;
@@ -8057,6 +8069,7 @@ parser_yylex(struct parser_params *parser)
      compile_error(PARSER_ARG "embedded document meets end of file");
      return 0;
    }
+  while (ISSPACE(c)) c = nextc();
+  if (c != '=') continue;
+  if (c == '=' && strncmp(lex_p, "end", 3) == 0 &&
      (lex_p + 3 == lex_pend || ISSPACE(lex_p[3]))) {
diff --git i/test/ruby/test_parse.rb w/test/ruby/test_parse.rb
index 56e1020c5a..a51b5471e4 100644
--- i/test/ruby/test_parse.rb
+++ w/test/ruby/test_parse.rb
@@ -988,6 +988,15 @@
  assert_equal(-100, e.backtrace_locations.first.lineno, bug)
end

```

```
+ def test_indented_multiline_comment
+   assert_valid_syntax("#{<<-\"begin;\"\n#\n#<<-\"end;\"}")
+   begin;
+     =begin
+       <.><^>
+     =end
+   end;
+ end
+
=begin
def test_past_scope_variable
  assert_warning(/past scope/) {catch {|tag| eval("BEGIN{throw tag}; tap {a = 1}; a")}}
```

#4 - 04/30/2017 12:17 AM - tscheingeld (Terry Scheingeld)

nobu (Nobuyoshi Nakada) wrote:

Possible of course, but I'm not a big fan of this.

Respect for your ability to quickly modify the code for what I suggested.

Just to clarify: are you not a big fan of the feature request in general, or just not a fan of the particular implementation you displayed?

#5 - 05/01/2017 08:04 AM - duerst (Martin Dürst)

Nobu can correct me if I'm wrong, but he meant the feature when he said "not a big fan". Same for me. =begin/=end are rare, and it's better to make them stick out.

#6 - 05/06/2017 10:13 PM - tscheingeld (Terry Scheingeld)

duerst (Martin Dürst) wrote:

=begin/=end are rare, and it's better to make them stick out.

I would make the case that =begin/=end is rare because it can't be indented. Multiline comments are very handy. It makes syntax highlighting much cleaner for embedded content (in my case, documentation) if every line doesn't have to start with a comment from the outer language.

#7 - 05/19/2017 08:14 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Feedback

How about that matching indentation of =begin and =end?

```
=begin
  in a comment
  =end
  still in a comment
=end
# end of the comment
```

#8 - 06/13/2017 02:22 AM - tscheingeld (Terry Scheingeld)

nobu (Nobuyoshi Nakada) wrote:

How about that matching indentation of =begin and =end?

```
=begin
  in a comment
  =end
  still in a comment
=end
# end of the comment
```

Makes sense to me.

#9 - 06/16/2017 08:57 AM - matz (Yukihiro Matsumoto)

- Status changed from Feedback to Rejected

=begin and =end are ugly appendix inherited from Perl. I'd rather remove them than enhance them.

Matz.