

CoCoA-5 - Bug #441

Emacs UI: C-c C-e does not ignore keywords inside strings

12 Feb 2014 12:00 - John Abbott

Status:	Closed	Start date:	12 Feb 2014
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	EmacsUI	Estimated time:	5.00 hours
Target version:	CoCoA-5.1.0 Easter14	Spent time:	5.35 hours
Description			
The emacs key sequence C-c C-e should automatically insert the correct End... keyword to close a syntactical construct. It works unless a CoCoA keyword appears inside a string literal -- I quite often write strings containing the word "for".			
Fix it (at least for normal string literals)!			
Related issues:			
Related to CoCoA-5 - Bug #470: Emacs UI: cocoa5-close-block should beep when ...		Closed	12 Mar 2014

History

#1 - 06 Mar 2014 20:53 - John Abbott

I believe the relevant function is cocoa5-close-block. The bug probably lies inside cocoa5-last-unended-begin (defn starts on line 860).

I noticed this very strange line (881); the first error is to catch all errors, the second error is to throw an error with message "Couldn't find..."

```
(error (error "Couldn't find unended command"))
```

#2 - 07 Mar 2014 11:36 - John Abbott

- Status changed from New to In Progress

- Assignee set to John Abbott

- % Done changed from 0 to 50

I have added a regexp "suffix" so that keywords are recognised "outside strings".

Here is the regexp

```
"[^\\"]*\\(\\\"[^\\"]*\"\\) *[^\\"]*$"
```

This regexp effectively just counts how many double quote characters there are after the keyword, so it will be confused by strings which contain (an odd number of) escaped double quote characters. Since strings containing escaped double quote characters are rare, I think this may be a fair compromise.

I did try creating a regexp which handles escaped double quotes but it didn't work -- I'm not entirely surprised as it was quite a convoluted regexp :-)

Should we accept my current regexp as an adequate solution? (and change this issue to "feedback"?)

#3 - 07 Mar 2014 12:03 - John Abbott

My regexp doesn't work quite as I hoped: it fails if the matching start keyword is on the same line as point. Here's a specific example:

```
If N > 1 Then PrintLn "For";
```

Executing C-c C-e at the end of this line does not find If. :(

#4 - 07 Mar 2014 12:48 - John Abbott

- Status changed from In Progress to Resolved

- % Done changed from 50 to 80

I now have a new regexp which seems to work better:

```
"[^"]*" * \\ ( \\ "[^"]*" * \\ \\ ) * [^"]*" * \\ ( $ \\ | \\ ' \\ \\ ) "
```

I have also added a save-restriction to the fn cocoa5-last-unended-begin

I have a more complicated regexp commented out; it seems not to work, but I cannot figure out why :(

Marking issue as "resolved"; but it needs more testing...

#5 - 07 Mar 2014 13:01 - John Abbott

Here are two test cases. The first works OK; the second does not.

```
Define works(X)  
  PrintLn "str1 keyword For";  
  If X > 1 Then PrintLn "str2", "str3";
```

Two uses of C-c C-e should correctly close the If and then the Define.

```
Define NotWorks(X)  
  PrintLn "str1 keyword For";  
  If X > 1 Then PrintLn "str2 quote mark \" ", "str3";
```

The simple regexp just counts " symbols and so thinks that If is inside a string and that For is outside.

Now I believe my fancier regexp works properly. It is very confusing with all those backslash escape characters! I do not put the regexp here because it confuses redmine (which loses almost the entire post!

