Subject: [solved] CParser GetColumn bugfix
Posted by cbpporter on Tue, 22 Oct 2019 09:24:46 GMT
View Forum Message <> Reply to Message

Calling GetColumn when the internal CParser::lineptr points to another line (which is rare in practice, but possible, can provide test case but don't think its needed) does not handle new-lines.

Fix:

```
int CParser::Pos::GetColumn(int tabsize) const
{
 int pos = 1;
 for(const char *s = lineptr; s < ptr; s++) {
  if(*s == CParser::LINEINFO_ESC) {
   s++;
   while(s < ptr && *s != CParser::LINEINFO_ESC)
    if(*s++ == '\3')
     pos = atoi(s);
  }
  else
  if(*s == '\n')
   pos = 0;  // FIX
  else
  if(*s == '\t')
   pos = (pos + tabsize - 1) / tabsize * tabsize + 1;
  else
   pos++;
 }
 return pos;
}
```

Additionally, I'm using CParser to parse more complicated input where simple SkipSpaces(true) doesn't works so I'm manually skipping whitespaces, but comments are still automatic.

I added the following overload to handle this:

```
CParser::CParser(const char *ptr, bool skipSpaces)
: term(ptr), wspc(ptr), lineptr(ptr)
{
 line = 1;
 skipspaces = skipSpaces;
 skipcomments = true;
 nestcomments = false;
 uescape = true;
 if (skipSpaces)
  Spaces();
}
```

I need a solution that support hundreds of CParser created on the spot for random code jumping that do not eat whitespaces since they are syntactically significant.

Subject: Re: CParser GetColumn bugfix
Posted by mirek on Tue, 22 Oct 2019 21:52:29 GMT
View Forum Message <> Reply to Message

Should not that be pos = 1?

Mirek

Subject: Re: CParser GetColumn bugfix
Posted by mirek on Tue, 22 Oct 2019 21:55:24 GMT
View Forum Message <> Reply to Message

I think the whitespace overloaded can be worked around by default contructor, setting CParser up (e.g. NoWhiteSpace), then use Set to start parsing. This is how it was meant...

Subject: Re: CParser GetColumn bugfix
Posted by mirek on Wed, 23 Oct 2019 13:57:43 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Tue, 22 October 2019 11:24Calling GetColumn when the internal CParser::lineptr points to another line (which is rare in practice, but possible, can provide test case but don't think its needed) does not handle new-lines.

Would it be possible to provide testcase anyway? More I think about this, less apparent it seems.

Subject: Re: CParser GetColumn bugfix
Posted by cbpporter on Mon, 28 Oct 2019 10:24:05 GMT
View Forum Message <> Reply to Message

It is indeed pos = 1.

It was 1 at one time then I tried to fix it in Spaces and that fix required 0, but it is easier and more reliable to fix in GetColumn.

This is the shortest real life example I could find:

```
CONSOLE_APP_MAIN {
 String s =
"+ /* \n"
"  */ a";

 CParser p(s);

 if (p.Char('+')) {
  Point z(p.GetLine(), p.GetPos().GetColumn());
  Cout() << z.x << ", " << z.y << ": " << p.ReadId() << "\n";
 }
}
```

Subject: Re: CParser GetColumn bugfix
Posted by cbpporter on Mon, 28 Oct 2019 10:26:13 GMT

View Forum Message <> Reply to Message

mirek wrote on Wed, 23 October 2019 00:55I think the whitespace overloaded can be worked around by default contructor, setting CParser up (e.g. NoWhiteSpace), then use Set to start parsing. This is how it was meant...
I'll test and see if that works. Thank you!

Subject: Re: CParser GetColumn bugfix
Posted by mirek on Wed, 30 Oct 2019 07:40:09 GMT

View Forum Message <> Reply to Message

cbpporter wrote on Mon, 28 October 2019 11:24It is indeed pos = 1.

It was 1 at one time then I tried to fix it in Spaces and that fix required 0, but it is easier and more reliable to fix in GetColumn.

This is the shortest real life example I could find:

```
CONSOLE_APP_MAIN {
 String s =
"+ /* \n"
"  */ a";

 CParser p(s);

 if (p.Char('+')) {
  Point z(p.GetLine(), p.GetPos().GetColumn());
  Cout() << z.x << ", " << z.y << ": " << p.ReadId() << "\n";
 }
}
```

Thank you. I believe I have that fixed 'correctly' (in Spaces0 etc...), in the trunk.

Mirek

Subject: Re: CParser GetColumn bugfix
Posted by cbpporter on Wed, 30 Oct 2019 09:12:31 GMT

View Forum Message <> Reply to Message

Here is my exact code from Spaces0 :o :

```
if(*term++ == '\n') {
      line++;
      //lineptr = term;
     }
```

This is the fix I retired for my new one, that worked better. But I didn't update GetChar and Skip Term.

But I'll test out you new fix.

Luckily, I have 170 test cases ready to run, some of them testing column too.

Subject: Re: CParser GetColumn bugfix
Posted by cbpporter on Thu, 31 Oct 2019 10:33:46 GMT

View Forum Message <> Reply to Message

The bugfix looks like it works! The GetChar did it.

I also changed my code to use Set.

New CParser constructor not needed anymore.

Thank you!