Subject: Surprising behavior of CParser
Posted by dolik.rce on Sat, 26 Apr 2014 11:18:52 GMT
View Forum Message <> Reply to Message

Hi everyone,

I just want to share a bit of knowledge about CParser I just learned the hard way This is not a rant, rather a cautionary tale:

There is a method SkipTerm, which does exactly that, skips one term. When in Spaces(true) mode, which is CParsers default, it also skips any whitespace after the string. So far it sounds reasonable a logical...

The surprising part to me was, that comments (both /* */ and //) are considered whitespace. Well, they fit the definition well. It kind of makes CParse unusable for many other languages, but I guess it can be explained by the 'C' in CParser

I hit this problem in Ini parser, which is part of U++ (in one particular part of it, that I contributed myself - so I'll also have to fix it). Ini file is definitely a C-like file and should not be treated like one. I forgot about this, and perhaps Mirek did as well, when he applied my patch that replaces environment variables with their values.

For illustration, here is a very simplified example: CParser p("http://example.com");
 while(!p.IsEof()) {
  if(p.IsId())
   Cout() << p.ReadId() << "\n";
  else
   p.SkipTerm();
 } My expectation was that code like this would print all ids in the string, that is "http". "domain" and "com". But in reality it prints only "http", because everything after "//" is discarded as a comment when SkipTerm() is called to skip ":".

I'm aware that this would be rather hard to fix in backward compatible way. Perhaps adding CParser::Comments(bool enable=true) method that would turn this behavior off only when required would be good idea. The main reason I write this post is to warn the rest of U++ user: It is dangerous to treat CParser as generic parser applicable to any text file. It is sure possible to parse almost anything with it, but one has to be really careful.

Hope this helps anyone

Honza