Subject: CodeEditor with user-definable syntax highlighting
Posted by dolik.rce on Sun, 24 Jun 2012 08:01:00 GMT
View Forum Message <> Reply to Message

Few days ago, I thought that it might be fun to reimplement CodeEditor to support user-defined highlighting as an excersise to avoid my brain slowly rotting from all that Java code I have to write at work I even started working on it in my free time, and here is a little tasting of the work done so far:

The image shows CodeEditor using the highlight definition file specifying highlighting of higlighting definition files to highlight highlight definition file specifying highlighting of higlighting definition files. Pretty meta, right?

It is based on pcre and caching. Basic highlighting works just fine, some of the higher level features (scope highlighting, matching brackets, ...) are not yet implemented and some are using the old CodeEditor code. The goal is to allow loading user specified highlighters, both at compile time (by include) and at run-time. It requires some pcre skill (or copy&paste skill) to write one, but otherwise it is very simple format as you can see above - it takes only 16 lines to describe C++ This file imitates current CodeEditor:Quote:HL_PATTERN(INK_OPERATOR,
"[\\Q!+-*/\%&|=[]:?<>.~^\\E]")
HL_PATTERN(INK_KEYWORD, " \\b(__(asm|cdecl|declspec|except|fastcall|finally|inline|int
(16|32|64|8)|leave|stdcall|try|uuidof)|auto|bool|break|case|
catch|char|class|const|const_cast|continue|default|delete|dl
l(ex|im)port|do|double|dynamic_cast|else|enum|explicit|exter
n|false|float|for|force_inline|friend|goto|if|inline|int|lon
g|mutable|namespace|new|operator|private|protected|public|re
gister|reinterpret_cast|return|short|signed|sizeof|static|st
atic_cast|struct|switch|template|this|thread|throw|true|try|
typedef|typeid|typename|union|unsigned|using|virtual|void|vo latile|while)\\b ")
HL_PATTERN(INK_UPPER, "\\b[A-Z][A-Z_0-9]*\\b")
HL_PATTERN(INK_UPP, "\\bupp\\b")
HL_PATTERN(INK_MACRO, " ^[[:space:]]*#[[:space:]]*(define|include(_next)?|(un|ifn?)d
ef|endif|el(if|se)|if|warning|error|pragma|using)\\b ")
HL_PATTERN(INK_UPPMACROS, "
\\b(CLASSNAME|(THIS|PTE)BACK([1-4]?)|QUOTE|X?ASSERT|X?NEVER|
X?CHECK|NAMESPACE_UPP|END_UPP_NAMESPACE|NEVER_|ASSERT_)\\b ")
HL_PATTERN(INK_UPPLOGS, "
\\b([DLR]?DUMP(C|CC|CC|HEX|M)?|[DLR]?LOG(F|HEX|BEGIN|END|BLO
CK|HEXDUMP|SRCPOS)?|[DLR]?TIMING|DEBUGCODE|RQUOTE)\\b ")
HL_PATTERN(INK_CONST_INT, "\\b[0-9]+\\b")
HL_PATTERN(INK_CONST_HEX, "\\b0(x|X)[[:xdigit:]]+\\b")
HL_PATTERN(INK_CONST_OCT, "\\b0[0-7]+\\b")
HL_PATTERN(INK_CONST_FLOAT, "\\b[0-9]+\\.[0-9]*\\b")
HL_PATTERN(INK_CONST_STRINGOP, "TODO")
HL_PATTERN(INK_CONST_STRING, "[\"](\\.|[^\"])*[\"]")
HL_PATTERN(INK_COMMENT, "//.*")
HL_PATTERN_ML(INK_COMMENT, "/*", "*/")
HL_SCOPE("[{]", "[}]")

Best regards,
Honza

PS: Reposted from this thread.

Subject: Re: CodeEditor with user-definable syntax highlighting
Posted by dolik.rce on Sun, 24 Jun 2012 08:09:44 GMT
View Forum Message <> Reply to Message

koldo wrote on Sun, 24 June 2012 09:20Hello Honza

It is interesting for me. I use a special language and I implemented some very basic syntax highlighting but I would like to use something much better.

Thank you .

Hi Koldo,

Would you like to use it in your application or in TheIDE? I plan to keep the API compatible with current CodeEditor, so they are switchable, but I haven't tried to compile it in TheIDE yet.

There are still some bugs that need to be fixed before it is usable at all. E.g: It does only work in Debug mode now, Optimal breaks it But considering that the main algorithms were written in one night, it is not that bad... I believe it could work reasonably well in couple weeks, if I have enough time.

I'll post updates here to keep you informed and I will upload the code when it gets into shape where I'm not ashamed for it

Honza

Subject: Re: CodeEditor with user-definable syntax highlighting
Posted by dolik.rce on Sun, 24 Jun 2012 08:25:21 GMT
View Forum Message <> Reply to Message

Didier wrote on Sun, 24 June 2012 09:55Hi Honza,

Very intersting

Although I don't need it for the moment, it can get handy when using a DSL or wrting a new one.

What is the perfomance on large files ?

Hi Didier,

The performance is much better than I expected

The worst text I tested it on was uppsrc/plugin/sqlite3/lib/sqlite3.c, which is a 131071 lines long beast. Even current CodeEditor chokes on it quite badly on my system (Intel Atom N280 @1.66GHz, 1Gb RAM)

It tears a lot when scrolling a too quick (i.e. thousands of lines at once), but so does current CodeEditor. This one might be slightly slower according to my benchmarks, but the majority of time is still spent in underlaying LineEditor code in both cases, so the difference is not really noticeable. Id say that subjectively the current and my new implementation are pretty much the same performance-wise.

Also, I haven't done much optimizations yet, only the basics. So it will probably get better over time (compared to current state, not to current CodeEditor code)

Honza

Subject: Re: CodeEditor with user-definable syntax highlighting
Posted by mirek on Sun, 24 Jun 2012 10:42:51 GMT
View Forum Message <> Reply to Message

I think we should definitely add this to theide in the next development cycle...

The question is, can this support something like html? (from the first naive evaluation, I guess not).

Speaking about it, I think that current hightlighting code became a little bit outdated now, I plan to do quite a bit refactoring into more modular approach..

Mirek

Subject: Re: CodeEditor with user-definable syntax highlighting
Posted by dolik.rce on Sun, 24 Jun 2012 14:40:46 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 24 June 2012 12:42I think we should definitely add this to theide in the next development cycle... Ok, why not. If my code is stable enough by then, of course.

mirek wrote on Sun, 24 June 2012 12:42The question is, can this support something like html? (from the first naive evaluation, I guess not). I plan to add support for "scoped matches", mainly as the mechanism behind the scope highlighting. It should be definitely possible to make it work for simple xml files with fixed number of known tags. The more general xml and html-like syntaxes might be little more tricky, but perhaps we can find some tricks to do that as well (it might work

with backreferences, but haven't tried it in practice yet).

mirek wrote on Sun, 24 June 2012 12:42Speaking about it, I think that current hightlighting code became a little bit outdated now, I plan to do quite a bit refactoring into more modular approach..The current highlighting code and the rest of the CodeEditor are bit too rigid for this I think. Letting user specify the rules opens much more possibilities, but it comes with it's own price. The CParser based approach is probably more efficient, but I have few ideas how to make the pcre perform a bit better too (I'll submit some patches soon).

Honza

Subject: Re: CodeEditor with user-definable syntax highlighting
Posted by koldo on Sun, 24 Jun 2012 16:22:47 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sun, 24 June 2012 10:09koldo wrote on Sun, 24 June 2012 09:20Hello Honza

It is interesting for me. I use a special language and I implemented some very basic syntax highlighting but I would like to use something much better.

Thank you .

Hi Koldo,

Would you like to use it in your application or in TheIDE? I plan to keep the API compatible with current CodeEditor, so they are switchable, but I haven't tried to compile it in TheIDE yet.

There are still some bugs that need to be fixed before it is usable at all. E.g: It does only work in Debug mode now, Optimal breaks it But considering that the main algorithms were written in one night, it is not that bad... I believe it could work reasonably well in couple weeks, if I have enough time.

I'll post updates here to keep you informed and I will upload the code when it gets into shape where I'm not ashamed for it

Honza
Hello Honza

In my case I want to include it in separate programs. However if it is enough good it could match in TheIDE.

Subject: Re: CodeEditor with user-definable syntax highlighting
Posted by mirek on Tue, 26 Jun 2012 18:37:36 GMT
View Forum Message <> Reply to Message

BTW, looking for the list of css keywords, I have stumbled upon this:

http://code.google.com/p/google-code-prettify/source/browse/ #svn%2Ftrunk%2Fsrc

Could be perhaps helpful for implementing support for other languages...

Mirek

Subject: Re: CodeEditor with user-definable syntax highlighting
Posted by dolik.rce on Sat, 07 Jul 2012 08:32:23 GMT
View Forum Message <> Reply to Message

Hi everyone,

As promised, I uploaded the first partly functional version of "CodeEditor2". You can find it in sandbox along with a simple test application that demonstrates an ability to tweak the highlighting at runtime.

There are still some rough corners and some missing functionality, but overall it seems to do the job. Ideas and suggestions are welcomed

Honza

Subject: Re: CodeEditor with user-definable syntax highlighting
Posted by koldo on Sun, 08 Jul 2012 19:33:14 GMT
View Forum Message <> Reply to Message

Great Honza

I am going to use immediately .

Subject: Re: CodeEditor with user-definable syntax highlighting
Posted by dolik.rce on Sun, 08 Jul 2012 20:36:04 GMT
View Forum Message <> Reply to Message

koldo wrote on Sun, 08 July 2012 21:33Great Honza

I am going to use immediately .
Thanks Koldo, but please keep in mind that it is definitely not a stable code. I would not recommend you to use it in production environment. It quite possibly still crashes sometimes and also the code will definitely change a lot before everything is finished.

Honza

Subject: Re: CodeEditor with user-definable syntax highlighting
Posted by koldo on Mon, 09 Jul 2012 00:22:54 GMT
View Forum Message <> Reply to Message

Oh well. So I will wait ...