
Subject: C++ Parser

Posted by [unodgs](#) on Wed, 28 Sep 2011 12:03:46 GMT

[View Forum Message](#) <> [Reply to Message](#)

Qt developers wrote today (<http://labs.qt.nokia.com/2011/09/28/qt-creator-editor-news/>) that they're experimenting with clang as a parser for intellisense. It seems to be a good idea because clang supports "fuzzy parsing". Maybe we should follow the same way?

Subject: Re: C++ Parser

Posted by [unodgs](#) on Wed, 28 Sep 2011 12:11:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

Here is a plugin for vim that uses clang for autocompletion

https://github.com/Rip-Rip/clang_complete

Subject: Re: C++ Parser

Posted by [unodgs](#) on Wed, 28 Sep 2011 12:16:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

This vim plugin seems to be very accurate:

<http://zwiener.org/vimautocomplete.html>

Subject: Re: C++ Parser

Posted by [mirek](#) on Wed, 28 Sep 2011 21:14:58 GMT

[View Forum Message](#) <> [Reply to Message](#)

Is it as fast as we need too?

Subject: Re: C++ Parser

Posted by [unodgs](#) on Wed, 28 Sep 2011 21:27:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Wed, 28 September 2011 17:14: Is it as fast as we need too?

I'm gonna find out (just built the newest version..)

Subject: Re: C++ Parser

Posted by [Novo](#) on Thu, 29 Sep 2011 14:30:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

unodgs wrote on Wed, 28 September 2011 08:11: Here is a plugin for vim that uses clang for

autocompletion
https://github.com/Rip-Rip/clang_complete

Thanks for a hint.

Subject: Re: C++ Parser
Posted by [unodgs](#) on Tue, 25 Oct 2011 12:27:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

I did some further work on integration of clang and code editor. Here's the result

When I fix some bugs I will upload binaries here so everybody could see how it works.
I'm also trying to parse all the up project functions/methods. When it's ready I'll publish the results about accuracy and speed of parsing. Anyway for now clang seems to be quite promising.

File Attachments

1) [clangeditor.png](#), downloaded 1342 times

```
Code editor + clang

public:
    String()
    {
    }
    String(const char* s)
    {
    }
    char operator[](int n)
    {
        return buff[n];
    }
    int GetLength()
    {
        String ss;
        ss.
        ret
    }
    String&
    {
        ret
    }
};

#define SS Stri

int Test(int a)
{
    return a * 2;
```

↑ Priority	Kind	Completion
34	Method	operator[](int n)
34	Method	GetLength()
34	Method	GetTypo()
34	Method	operator=(const String &)
34	Destructor	~String()
35	Field declaration	length
35	Field declaration	buffer
75	Class	String::

Subject: Re: C++ Parser

Posted by [mirek](#) on Tue, 25 Oct 2011 15:36:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

unodgs wrote on Tue, 25 October 2011 08:27 I did some further work on integration of clang and code editor. Here's the result

When I fix some bugs I will upload binaries here so everybody could see how it works.

I'm also trying to parse all the upp project functions/methods. When it's ready I'll publish the results about accuracy and speed of parsing. Anyway for now clang seems to be quite promising.

I believe that the real trouble is not C++ parser alone, working on single file, but how to make all files work together (with all macros and headers and such).

E.g. for Alt+J (jump to definition) to work, you need to keep track of all files...

Is this somehow solved in clang?

Mirek

Subject: Re: C++ Parser

Posted by [fudadmin](#) on Tue, 25 Oct 2011 15:46:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have been also working on the ide upp Cpp code parsing and browsing improvement alternative and also using GridCtrl . I tried to use clang for compiling upp on mac but it didnt work. Are you able to compile upp with clang?

File Attachments

1) [Screen shot 2011-10-25 at 16.25.16.png](#), downloaded 643 times

- ide
- Comm...
- Core
- LayDes
- IconD...
- Build...
- Debu...
- Browser
- CodeE...
- CtrlLib
- Web
- bz2
- HexView
- BlueBar
- astyle
- usvn
- TextDif...
- TabBar
- SrcU...
- Esc
- Core
- z
- RichEdit
- IconDes
- gif
- jpg

- Draw
- binobj
- coff
- ndisasm
- CppBase
- PdfDraw
- Topic
- CtrlC...
- bmp
- RichText
- png
- <prj-a...
- <ide-a...
- <temp...
- <meta>

ndisasm.h x Browser.h x CodeBrowser.cpp x ide.h x Topic.lay x Browser.l

```
(Search (F12))
_Browser_Browser_h
LAYOUTFILE
IMAGECLASS
IMAGEFILE
WITHBODY
BrowserFont
CppNestingInfo
CppitemInfo
ItemTextPart
BrowserFileInfo
AdditionalKinds
BrowserQuery
QueryDlg
CppitemInfoGridDisplay
CppitemInfoDisplay
CppNestingInfoDisplay
CppNestingInfoGridDisplay
ItemList
Nesting
Browser
LAYOUTFILE
ReferenceDlg
TopicEditorIde
CLASSNAME
```

```

224 void Clear();
225
226 ItemList();
227 };
228
229 class Nesting : public GridCtrl {
230 public:
231 int FindSetCursor(const Value& val);
232 // void GoToNest(const );
233 Value GetKey() const;
234 void ScCursor(int a);
235 void Clear();
236 void Sort();
237 typedef Nesting CLASSNAME;
238 Nesting();
239 };
240
241 void SetFilter1(EditString *e, const char
242
243 class Browser : public StaticRect {
244 public:
245 void Serialize(Stream& s);
246 void SerializeWspc(Stream& s);
247
248 Nesting nesting;
249 // GridCtrl nesting;
250
251 ItemList item;
252 Splitter split;
253 GridFilter filter;
254
255 Splitter split2;
256 // Array<EditField> itf; //
257 // void OnFilterChange(Id id);

```

- # Browser.h
- Browser
- Lay.cpp
- Base.cpp
- Util.cpp
- Item.cpp
- ItemDisplay.cpp
- ItemGridDisplay.cpp
- ItemList.cpp
- Query.cpp
- Browser1.cpp
- Browser.cpp
- Reference.cpp
- Browser.lay
- ReferenceDlg.lay
- Browser.iml
- Topic
- arisnotes.txt
- IdeTopic.cpp
- Topic1.cpp

Nesting:1716
meconsorcioGroup
::TopicCtrl
::TopicCtrl::CLASSNAME
::IdeCalc
::RightTabs
::RightTabs::Tab
::EditorTabBar
::EditorTabBar::CLASSNAME
::AssistEditor
::AssistEditor::IndexItem
::AssistEditor::ParamInfo
::AssistEditor::CLASSNAME
::FindInFilesDlg
::FindInFilesDlg::CLASSNAME

CppInfo:30411	Access
bool Nesting(const CppNesting& n, const Strin	0
void TypeOf(const String& id, Vector<Stri	0
Vector<String> Operator(const char*ope	0
Vector<String> TypeOf(const Vector<Stri	0
String RemoveDefPar(const char*s)	0
String MakeDefinition(const String& cls,	0
void DCopy()	0
void Virtuals()	0
void Thisbacks()	0

All
 CppInfo
 Access

clear->

CppInfo
 Acc

Subject: Re: C++ Parser

Posted by [Novo](#) on Tue, 25 Oct 2011 15:53:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Tue, 25 October 2011 11:36

I believe that the real trouble is not C++ parser alone, working on single file, but how to make all files work together (with all macros and headers and such).

AFAIK, clang is a real compiler, so, it has to parse all included files. The difference with gcc is that internal data structure is more high level and it stores more info for each parsed element. As a result you can use it not just to jump/navigate among files but also to regenerate file back (or to do some refactoring and generate files after that). It is a real parser an you have real type info about each node. The only question is performance. But XCode is based on clang and seems to work fine.

Subject: Re: C++ Parser

Posted by [unodgs](#) on Tue, 25 Oct 2011 16:48:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:I believe that the real trouble is not C++ parser alone, working on single file, but how to make all files work together (with all macros and headers and such).

I had also `#include <Core/Core.h>` in the sample file and the speed was more than acceptable (especially the second time when files were cached by the system I guess). Besides clang supports precompiled headers what should make parsing fast enough even if lots of headers and macros are involved.

Integrating clang with upp is also very easy. It's only one file `libclang.dll` (4.7 MB) that has to be distributed with the rest of the framework.

Anyway I'll try to finish the demo and add the whole upp library with windows headers and see what the speed will be.

Subject: Re: C++ Parser

Posted by [unodgs](#) on Tue, 25 Oct 2011 16:52:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

fudadmin wrote on Tue, 25 October 2011 11:46I have been also working on the ide upp Cpp code parsing and browsing improvement alternative and also using `GridCtrl` .

That's a great news! Could you share your sources?

Quote:I tried to use clang for compiling upp on mac but it didnt work. Are you able to compile upp with clang?

On win64 clang exploded during compilation one of the upp's package (I used svn version not a stable one). But I'll keep on trying
