
Subject: Disable Assits++ scan on certain files
Posted by [Didier](#) on Fri, 07 Dec 2018 18:42:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

When using Boost::spirit, I have an issue with assist++ scanner : it get's stuck in Boost headers.
So if I want to be able to use Assis++, I have to comment Boost header #include, launch rescan of all files, uncomment the boost headers #include

As BLITZ can be disabled on certain files using #pragma BLITZ_PROHIBIT

Maybe Assist++ scanning could be disable using something like #pragma ASSIST_SCAN_PROHIBIT

Subject: Re: Disable Assits++ scan on certain files
Posted by [Klugier](#) on Sat, 08 Dec 2018 17:40:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I found missing documentation page that speaks about your problem. The assist parsing can be turn off by following macros (part of missing documentation):

5. Assist++ C++ parser directives

Assist supports simple directives that are passed to it via `//'` comments that can be used in cases that original code is confusing:

`//$-`

Code past this directive will not be passed to parser.

`//$+`

Code past this directive will be passed to parser (stops `//$-`).

`//$ code`

example: `//$ void Foo(Bar&)`

Code will be passed to parser (adds code that is not part of compiled sources)

The documentation page will be available tomorrow and it will be bundled with TheIDE. I hope my message will help you to deal with your problem.

Sincerely,
Klugier

Subject: Re: Disable Assits++ scan on certain files
Posted by [Didier](#) on Sat, 08 Dec 2018 18:11:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Klugier,

I was filled with hope when I saw you're answer thinking i've ignored this for so long... but it doesn't work :cry: :cry: :cry: :cry: :cry: :cry:
I even tried the other way around :?

```
//$-  
#include <boost/config/warning_disable.hpp>  
#include <boost/spirit/include/qi.hpp>  
#include <boost/spirit/include/phoenix_core.hpp>  
#include <boost/spirit/include/phoenix_operator.hpp>  
#include <boost/spirit/include/phoenix_fusion.hpp>  
#include <boost/spirit/include/phoenix_stl.hpp>  
#include <boost/spirit/include/phoenix_object.hpp>  
#include <boost/fusion/include/adapt_struct.hpp>  
#include <boost/spirit/include/phoenix_function.hpp>  
#include <boost/spirit/include/qi_expect.hpp>  
//$+
```

Subject: Re: Disable Assits++ scan on certain files
Posted by [Didier](#) on Sat, 08 Dec 2018 18:12:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks anyway :)

Subject: Re: Disable Assits++ scan on certain files
Posted by [Didier](#) on Tue, 26 Feb 2019 21:13:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I took some time to find out why it doesn't work in my case, and I think I found a candidate
//\$+ and //\$- work when parsing the code of a file
But the list of files is extracted from the package files and file #include without taking into
account the //\$+ and //\$-
The function that does this extract seems to be (Is this correct ??) : ide/Browser/Base.cpp
BaseInfoSync(Progress& pi)

So the following code does not behave as I expected : I thought the includes (and the included
files) would not be processed by assist++

```
//$-  
#include <boost/config/warning_disable.hpp>
```

```
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/phoenix_bind.hpp>
#include <boost/fusion/include/adapt_struct.hpp>
#include <boost/variant/recursive_variant.hpp>
#include <boost/foreach.hpp>
// $+
```

Maybe the file should be preprocessed by :
 CppBase/Pre.cpp SrcFile PreProcess(Stream& in, Parser& parser)
 Before extracting the list of files to be processed by Assist++

Subject: Re: Disable Assits++ scan on certain files
 Posted by [Didier](#) on Sat, 31 Aug 2019 22:20:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well I finally found some time to look into this problem and actually found something that works.

Code is just a first quick try, but before going further I would like to share it first and have some return on it.

* in uppsrc/CppBase/CppBase.h

```
const char *SkipString(const char *s);

void RemoveComments(String& l, bool& incomment);
void RemoveDirectives(String& l, bool& inDirective); // ===== ADDED THIS LINE
=====
```

```
String GetStdDefs();
```

* in uppsrc/CppBase/ppFile.cpp
 Added the following function:

```
void RemoveDirectives(String& l, bool& inDirective)
{
  int q = -1;
  int w = -1;
  if(inDirective)
```

```

q = w = 0;
else {
    const char *s = l;
    while(*s) {
        if(*s == '\\')
            s = SkipString(s);
        else
            if(s[0] == '/' && s[1] == '/' && s[2] == '$' && s[3] == '-') {
                q = int(s - ~l);
                break;
            }
        else
            s++;
    }
    if(q >= 0)
        w = q + 4;
}
while(q >= 0) {
    int eq = l.Find("//$+", w);
    if(eq < 0) {
        inDirective = true;
        SetSpaces(l, q, l.GetCount() - q);
        return;
    }
    SetSpaces(l, q, eq + 4 - q);
    inDirective = false;
    q = l.Find("//$+");
    w = q + 4;
}
}

```

* Modified ppFile::parse() method

```

void PPFile::Parse(Stream& in)
{
    LTIMING("PPFile::Parse");
    for(int i = 0; i < ppmacro.GetCount(); i++)
        sAllMacros.Unlink(ppmacro[i]);
    ppmacro.Clear();
    item.Clear();
    includes.Clear();
    bool was_using = false;
    bool was_namespace = false;
    int level = 0;
    bool incomment = false;
    bool inDirective = false;    ===== line added =====
    Vector<int> namespace_block;
    bool next_segment = true;

```

```
Index<int> local_segments;
keywords.Clear();
int linei = 0;
Md5Stream md5;
while(!in.IsEof()) {
    String l = in.GetLine();
    while(*l.Last() == '\\' && !in.IsEof()) {
        l.Trim(l.GetLength() - 1);
        l.Cat(in.GetLine());
    }
    RemoveDirectives(l, inDirective);    ===== line added =====
    RemoveComments(l, incomment);
    try {
        CParser p(l);
```

Subject: Re: Disable Assits++ scan on certain files
Posted by [Didier](#) on Sat, 31 Aug 2019 22:21:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Assist++ doesn't get stuck while scanning c++/boost header files 8)

Subject: Re: Disable Assits++ scan on certain files
Posted by [mirek](#) on Fri, 06 Sep 2019 09:03:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

I have now fixed the original issue (parser does not get stuck there anymore) and also implement a little bit more polished version of directives to remove #includes.

Mirek

Subject: Re: Disable Assits++ scan on certain files
Posted by [Didier](#) on Fri, 06 Sep 2019 10:44:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks, great,
I'll try it this afternoon

Subject: Re: Disable Assits++ scan on certain files

Posted by [Didier](#) on Fri, 06 Sep 2019 19:39:55 GMT

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

Well I checked latest SVN version :

- * Without `//$- //$+` : scanner still gets stuck in boost headers
- * With `//$- //$+` : everything is OK

So for me it's all good since I don't need to scan the boost headers.

Thank's for makng the changes to ide

Subject: Re: Disable Assits++ scan on certain files

Posted by [mirek](#) on Mon, 09 Sep 2019 06:16:41 GMT

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

Didier wrote on Fri, 06 September 2019 21:39Well I checked latest SVN version :

- * Without `//$- //$+` : scanner still gets stuck in boost headers
- * With `//$- //$+` : everything is OK

So for me it's all good since I don't need to scan the boost headers.

We really need that to work... Where does it get stuck? (Perhaps send screenshot of progress....)

What is the version of boost?

Subject: Re: Disable Assits++ scan on certain files

Posted by [Didier](#) on Tue, 10 Sep 2019 18:06:07 GMT

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

Hello Mirek,

I will provide all debug info tonight, when I first took a look at it, it got stuck in an infinite loop.

It always get's stuck on the same file

Subject: Re: Disable Assits++ scan on certain files

Posted by [mirek](#) on Tue, 10 Sep 2019 21:04:22 GMT

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

Didier wrote on Tue, 10 September 2019 20:06Hello Mirek,

I will provide all debug info tonight, when I first took a look at it, it got stuck in an infinite loop.

It always get's stuck on the same file

Please check and report the svn version of theide as an infinite loop is exactly what I have fixed...

Mirek

Subject: Re: Disable Assits++ scan on certain files
Posted by [Didier](#) on Tue, 10 Sep 2019 21:54:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

I just checked with latest svn version (13596) and it works now !!! :)

Many Many Thank's, this was very annoying

My previous test, which did not work, was made with svn 13591

The boost version used is : 1.63.0

I am on linux (fedora) and compile with CLANG

Linux version 4.13.16-100.fc25.x86_64 (mockbuild@bkernel01.phx2.fedoraproject.org) (gcc
version 6.4.1 20170727 (Red Hat 6.4.1-1) (GCC)) #1 SMP Mon
Nov 27 19:52:46 UTC 2017

clang version 3.9.1 (tags/RELEASE_391/final)
Target: x86_64-unknown-linux-gnu
Thread model: posix

Subject: Re: Disable Assits++ scan on certain files
Posted by [mirek](#) on Wed, 11 Sep 2019 07:38:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Didier wrote on Tue, 10 September 2019 23:54I just checked with latest svn version (13596) and it works now !!! :)

Many Many Thank's, this was very annoying

My previous test, which did not work, was made with svn 13591

Just as I thought... :) Thanks for checking again!

Mirek
