Subject: Macro thoughts
Posted by mirek on Thu, 20 Nov 2008 10:16:11 GMT
View Forum Message <> Reply to Message

I am getting ready about adding support of macros to C++ parser.

There is one hard issue there:

When any macro changes, it is often required to reparse a lot of files (all that use it).

As we parse files quite often (at least, at each file switch), this would be slow.

I am therefore trying to find some other, less complex way how to do that.

The simple way is to do it at compile time (if any macros have changed) and at user request. The downside is that sometimes the code database would be 'out-of-sync' (contain wrong information w.r.t. macros).

Is that acceptable? Any better ideas?

Mirek

Subject: Re: Macro thoughts
Posted by mirek on Thu, 20 Nov 2008 11:04:34 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 20 November 2008 05:16
The simple way is to do it at compile time

Well, that is confusing term, I meant when you hit "Build"...

Mirek

Subject: Re: Macro thoughts
Posted by unodgs on Thu, 20 Nov 2008 14:57:17 GMT
View Forum Message <> Reply to Message

If you collect information about which file is affected by particular macro I think this scenario could be applied - for macros used by one or two files assist should work imediatelly, for rest before build process. Some kind of simple heuristics.

Subject: Re: Macro thoughts
Posted by mirek on Thu, 20 Nov 2008 17:29:56 GMT
View Forum Message <> Reply to Message

unodgs wrote on Thu, 20 November 2008 09:57If you collect information about which file is affected by particular macro I think this scenario could be applied - for macros used by one or two files assist should work imediatelly, for rest before build process. Some kind of simple heuristics.

Uhm, is it worth the trouble?

OTOH, not a bad idea.

Mirek

Subject: Re: Macro thoughts
Posted by unodgs on Thu, 20 Nov 2008 17:53:56 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 20 November 2008 12:29unodgs wrote on Thu, 20 November 2008 09:57If you collect information about which file is affected by particular macro I think this scenario could be applied - for macros used by one or two files assist should work imediatelly, for rest before build process. Some kind of simple heuristics.

Uhm, is it worth the trouble?

OTOH, not a bad idea.

Mirek
The only rason it's worth the trouble is a case when you edit macro used only in current file and you use this macro in this file so you can have correct intelisense.
I think for now you could parse files only before build. It should be enough. I didn't write any macro since one/two years..
I just gave you an idea because you were asking

Subject: Re: Macro thoughts
Posted by mirek on Thu, 20 Nov 2008 18:20:59 GMT
View Forum Message <> Reply to Message

unodgs wrote on Thu, 20 November 2008 12:53luzr wrote on Thu, 20 November 2008 12:29unodgs wrote on Thu, 20 November 2008 09:57If you collect information about which file is affected by particular macro I think this scenario could be applied - for macros used by one or two files assist should work imediatelly, for rest before build process. Some kind of simple heuristics.

Uhm, is it worth the trouble?

OTOH, not a bad idea.

Mirek
The only rason it's worth the trouble is a case when you edit macro used only in current file and you use this macro in this file so you can have correct intelisense.
I think for now you could parse files only before build. It should be enough. I didn't write any macro since one/two years..
I just gave you an idea because you were asking

The file edited would be reparsed anyway, it always is.

Well, I guess this should work

Mirek

Subject: Re: Macro thoughts
Posted by Mindtraveller on Sat, 22 Nov 2008 18:17:45 GMT
View Forum Message <> Reply to Message

Is there a backgrounf file reparsing?
The best IMO would be unodgs proposal as "quick try", the background processing to acquire more exact results.

Subject: Re: Macro thoughts
Posted by mirek on Sat, 22 Nov 2008 18:41:51 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Sat, 22 November 2008 13:17Is there a backgrounf file reparsing?
The best IMO would be unodgs proposal as "quick try", the background processing to acquire more exact results.

Background reparsing is though job.

Mirek

Subject: Re: Macro thoughts
Posted by Sender Ghost on Thu, 04 Dec 2008 12:44:23 GMT
View Forum Message <> Reply to Message

Hello, Mirek.

I think, you can see "mcpp - a portable C preprocessor" code, released under BSD-style-license. I'm already use it before Doxygen.

Subject: Re: Macro thoughts
Posted by mr_ped on Thu, 04 Dec 2008 13:02:20 GMT
View Forum Message <> Reply to Message

Sender Ghost wrote on Thu, 04 December 2008 13:44Hello, Mirek.

I think, you can see "mcpp - a portable C preprocessor" code, released under BSD-style-license. I'm already use it before Doxygen.

How fast it is? Can it improve compilation time of UPP, if it will be used for preprocessing and let gcc/MSC just to compile?
Anybody with enough free time on hands to try?

Subject: Re: Macro thoughts
Posted by mirek on Thu, 04 Dec 2008 14:56:55 GMT
View Forum Message <> Reply to Message

Sender Ghost wrote on Thu, 04 December 2008 07:44Hello, Mirek.

I think, you can see "mcpp - a portable C preprocessor" code, released under BSD-style-license. I'm already use it before Doxygen.

It would be interesting if I would plan to do C++ compiler.

For our purpose, we need something different.

In fact, parsing is not a problem with macros. The problem is the whole structure. We cannot preprocess files the same way as C++ compiler does - that would be too slow for A++.

Mirek

Subject: Re: Macro thoughts
Posted by Sender Ghost on Thu, 04 Dec 2008 16:58:55 GMT
View Forum Message <> Reply to Message

luzr
It would be interesting if I would plan to do C++ compiler.

For our purpose, we need something different.

In fact, parsing is not a problem with macros. The problem is the whole structure. We cannot preprocess files the same way as C++ compiler does - that would be too slow for A++.

Mirek

Ok, no problem here. Just to note.
I suggest you some modes for preprocessor part of A++:
- Current mode: Current OS, compiler and configuration.
- Free mode: All OS'es, compilers (and configuration).
mr_ped
How fast it is? Can it improve compilation time of UPP, if it will be used for preprocessing and let gcc/MSC just to compile?
Anybody with enough free time on hands to try?

Would be slow. It's not my point. I thinking Mirek have some troubles with preprocessing. He haven't.

Subject: Re: Macro thoughts
Posted by mirek on Thu, 04 Dec 2008 17:34:01 GMT

View Forum Message <> Reply to Message

Sender Ghost wrote on Thu, 04 December 2008 11:58luzr
It would be interesting if I would plan to do C++ compiler.

For our purpose, we need something different.

In fact, parsing is not a problem with macros. The problem is the whole structure. We cannot preprocess files the same way as C++ compiler does - that would be too slow for A++.

Mirek

Ok, no problem here. Just to note.
I suggest you some modes for preprocessor part of A++:
- Current mode: Current OS, compiler and configuration.
- Free mode: All OS'es, compilers (and configuration).

Actually, it will be the second one:)

Mirek