
Subject: CPP file dependency
Posted by [raxvan](#) on Mon, 19 Sep 2011 17:30:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

If you have a C++ project and change one file how do you detect all the dependences to build the correct number of Cpp files affected by your change?

In what way is Theide doing this?

Thanks,
Raxvan.

Subject: Re: CPP file dependency
Posted by [dolik.rce](#) on Mon, 19 Sep 2011 19:28:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Raxvan

If you change a .cpp file, only this single file needs to be rebuild (there are some exceptions, when you use macros heavily, but that is ignored by theide). If you use BLITZ then you might need to recompile the entire "blitz batch" in case that the file you just changed was part of it in the last rebuild.

If you change an .h file, theide will rebuild all .c and .cpp files that include this file. Also files included recursively (i.e. through other .h files) are rebuild. Again, if the "blitz batch" contents change or if any of the files in it includes the changed .h file, then it will be rebuild.

I think these are the basic rules that are used in theide. Does that answer your question sufficiently?

Best regards,
Honza

Subject: Re: CPP file dependency
Posted by [mirek](#) on Mon, 19 Sep 2011 21:23:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

raxvan wrote on Mon, 19 September 2011 13:30
In what way is Theide doing this?

It scans through files to gather #include dependencies...

Mirek

Subject: Re: CPP file dependency
Posted by [raxvan](#) on Tue, 20 Sep 2011 10:04:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Mon, 19 September 2011 21:28Hi Raxvan

If you change a .cpp file, only this single file needs to be rebuild (there are some exceptions, when you use macros heavily, but that is ignored by theide). If you use BLITZ then you might need to recompile the entire "blitz batch" in case that the file you just changed was part of it in the last rebuild.

If you change an .h file, theide will rebuild all .c and .cpp files that include this file. Also files included recursively (i.e. through other .h files) are rebuild. Again, if the "blitz batch" contents change or if any of the files in it includes the changed .h file, then it will be rebuild.

I think these are the basic rules that are used in theide. Does that answer your question sufficiently?

Best regards,
Honza

Thanks for the reply, i was interested if an external tool is used or the dependency checker is build in house.

In theide, is it safe if i use macro magic like #include MY_FILE and MY_FILE is defined in some other place? How do you scan then for dependencies without implementing a full preprocessor? This is basically my initial problem. How to solve a dependency if "#include MY_FILE" is used.

Raxvan.

Subject: Re: CPP file dependency
Posted by [dolik.rce](#) on Tue, 20 Sep 2011 10:53:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

raxvan wrote on Tue, 20 September 2011 12:04In theide, is it safe if i use macro magic like #include MY_FILE and MY_FILE is defined in some other place? How do you scan then for dependencies without implementing a full preprocessor? This is basically my initial problem. How to solve a dependency if "#include MY_FILE" is used.

Nope, as of now TheIDE doesn't understand macros very well, so this is not really safe. On POSIX platforms you could use "touch" as custom build step to force update of the files with the #include, but that is not portable and in some cases it would result into rebuilding everything all the time...

If you know that the definition of MY_FILE will not change too often you can ignore it, but be sure to use "Clean package" after each such change. What exactly are you trying to achieve? Maybe we can try to find some other solution?

Honza

Subject: Re: CPP file dependency
Posted by [raxvan](#) on Tue, 20 Sep 2011 11:31:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

I need a tool to safely optimally construct the dependency tree. I found <http://code.google.com/p/include-what-you-use/> but i'm thinking it is too complex. An easier solution is to just simply scan the file for #include and for the macro hack just to ignore it.

Raxvan

Subject: Re: CPP file dependency
Posted by [dolik.rce](#) on Tue, 20 Sep 2011 12:47:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

If that is all you want then you might want to look at Doxygen, which can generate inclusion graphs if instructed to. Also writing simple script that uses graphviz should be simple - I wrote similar thing for Upp packages some time ago.

Also "g++ -H ..." outputs the entire include hierarchy - you could simply parse it and visualize in other tool, e.g. in already mentioned graphviz. Looking at -M*, -d* and -E options might be of some help as well...

Honza

Subject: Re: CPP file dependency
Posted by [mirek](#) on Thu, 22 Sep 2011 07:36:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Tue, 20 September 2011 06:53
raxvan wrote on Tue, 20 September 2011 12:04
In the IDE, is it safe if I use macro magic like #include MY_FILE and MY_FILE is defined in some other place? How do you scan then for dependencies without implementing a full preprocessor? This is basically my initial problem. How to solve a dependency if "#include MY_FILE" is used.

Nope, as of now TheIDE doesn't understand macros very well, so this is not really safe. On POSIX platforms you could use "touch" as custom build step to force update of the files with the #include, but that is not portable and in some cases it would result into rebuilding everything all the time...

Actually, the code used to check for dependencies accounts for macros quite well. Otherwise we could not use `#include LAYOUTFILE` etc...

Mirek

Subject: Re: CPP file dependency
Posted by [dolik.rce](#) on Thu, 22 Sep 2011 08:57:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Thu, 22 September 2011 09:36: Actually, the code used to check for dependencies accounts for macros quite well. Otherwise we could not use `#include LAYOUTFILE` etc...

Mirek
Oups, my mistake then Sorry for underestimating TheIDE.

Honza
