Subject: #include "_uses_.h"
Posted by mirek on Mon, 18 Aug 2008 18:30:16 GMT
View Forum Message <> Reply to Message

I have got a strange idea.

Maybe, after editing in package organizer, we could create _uses_.h file in each package directory and automatically generate a list of include of used packages there.

That way you would not have to add individual #includes after adding a package to the project, just keep single #include in main .h.

Anyway, is not this going too far?

Mirek

Subject: Re: #include "_uses_.h"
Posted by cbpporter on Mon, 18 Aug 2008 18:59:48 GMT
View Forum Message <> Reply to Message

It's not going too far, but not quite necessary on the other hand. With such a modification you'd have almost perfectly mimicked a Java like package system. You won't stop until you turn C++ into something it's not .

Another idea, completely unrelated to you proposition: how hard would it be to create fake packages which include just some .h files that are parsed for Assist++ purposes, but are otherwise completely excluded from build. This is so that interacting with non U++ libraries becomes easier. Right now I have to create a package based on the library, it has to compile and link, and before deployment I have to replace it with "official" library.

And one more, for which I was going to start a new thread, but I'll just post here: how do you feel about having some autogenerated C interfaces for some GUI classes to allow cross language bindings? The only real problem I see is with callbacks.

Subject: Re: #include "_uses_.h"
Posted by mirek on Mon, 18 Aug 2008 22:28:32 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 18 August 2008 14:59
Another idea, completely unrelated to you proposition: how hard would it be to create fake packages which include just some .h files that are parsed for Assist++ purposes, but are otherwise completely excluded from build. This is so that interacting with non U++ libraries becomes easier. Right now I have to create a package based on the library, it has to compile and link, and before deployment I have to replace it with "official" library.

Eh, actually, would not it be easier to support non-U++ headers directly?

At some moment in the past, I was experimenting with this. I have abandoned the idea because both STL headers and Win32 are riddled with macros, so it has not produced any good results.

But when macros are covered....

Quote:
And one more, for which I was going to start a new thread, but I'll just post here: how do you feel about having some autogenerated C interfaces for some GUI classes to allow cross language bindings?

"uneasy" ?

Mirek

Subject: Re: #include "_uses_.h"
Posted by mr_ped on Tue, 19 Aug 2008 07:28:09 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 18 August 2008 20:30Maybe, after editing in package organizer, we could create _uses_.h file in each package directory and automatically generate a list of include of used packages there.

That way you would not have to add individual #includes after adding a package to the project, just keep single #include in main .h.

Maybe it's too early in the morning for me (well, not maybe, it IS), but I don't get it.
If the package contains API.h (public API of package), internal.h (internal classes which are not needed+wanted in public API), you will include this package into different one, what will get included?

I had very similar idea recently in terms of result (I mean the "successful compilation with minimal effort" result, not "one line include" one), and I was thinking more about IDE detecting usage of undefined classes/macros/other in .cpp files (in .h not auto detection, as it would clash sometimes with forward defined pointer types), and searching trough used packages/include path/any packages *.h files to find a definition and auto-add (or suggest) the necessary #include lines.
(in ideal world even checking from time to time for unneeded includes and suggesting their removal)

This way just the needed .h would be included, yet the programmer would not need to write any "include" lines (I believe it's just wasted time of programmer, and I believe it *can* be solved by CPU, so this idea is IMHO feasible, but maybe I'm overlooking some complex catch).

With BLITZ enabled including just single file will make the source easier to read (although it will hide some information ... I rather prefer cleanness by simplicity, not by obscurity) and the speed of compilation would be still good, but with BLITZ off including just used .h files can have tremendous impact on compilation speed.

Anyway, if you go by the "_used_.h" route, it will pollute the source directory again like "init" files? *frowned look*

Subject: Re: #include "_uses_.h"
Posted by cbpporter on Tue, 19 Aug 2008 08:48:38 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 19 August 2008 01:28
Eh, actually, would not it be easier to support non-U++ headers directly?

At some moment in the past, I was experimenting with this. I have abandoned the idea because both STL headers and Win32 are riddled with macros, so it has not produced any good results.

But when macros are covered....

It would be easier and more straightforward. If you think that it will be technically possible to get header parsing working at this moment that is. The headers I'm using are not only riddled with macros, but also a lot of freaky constructs. Can't alter them though. GPL.

Quote:
"uneasy" ?

. Care to elaborate?

BTW, mr_ped does have a point. On the other hand, I don't think that we'll ever have "minimal .h dependency" approach to code layout, so I doesn't really matter IMO.

Subject: Re: #include "_uses_.h"
Posted by mirek on Tue, 19 Aug 2008 08:57:01 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Tue, 19 August 2008 03:28luzr wrote on Mon, 18 August 2008 20:30Maybe, after editing in package organizer, we could create _uses_.h file in each package directory and automatically generate a list of include of used packages there.

That way you would not have to add individual #includes after adding a package to the project, just keep single #include in main .h.

Maybe it's too early in the morning for me (well, not maybe, it IS), but I don't get it.

Simply including the very first header would work in most cases.

We can add some more logic about this of course.... Plus, using _uses_ is not mandatory after all...

Quote:
This way just the needed .h would be included, yet the programmer would not need to write any "include" lines (I believe it's just wasted time of programmer, and I believe it *can* be solved by CPU, so this idea is IMHO feasible, but maybe I'm overlooking some complex catch).

Well, not this is "going too far" for me

Quote:
With BLITZ enabled including just single file will make the source easier to read (although it will hide some information ... I rather prefer cleanness by simplicity, not by obscurity) and the speed of compilation would be still good, but with BLITZ off including just used .h files can have tremendous impact on compilation speed.

Technically, that is correct. But the current state of U++ sources is exactly the same... IMO it is OK to depend on BLITZ during development and with eventual source form deployment (without theide) the time to build the stuff does not matter that much.

Quote:
Anyway, if you go by the "_used_.h" route, it will pollute the source directory again like "init" files? *frowned look*

Yep. :-\

Mirek

Subject: Re: #include "_uses_.h"
Posted by mirek on Tue, 19 Aug 2008 10:00:50 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Tue, 19 August 2008 04:48
Quote:
"uneasy" ?

. Care to elaborate?

Well, really I have not problem with other language bindings.

However, I think U++ is very C++. I am not quite sure what will remain if you create bindings for other languages....

Eg, I think destructors are THE foundation of U++ architecture. Without destructors, everything will have to be complete different.

IMO, of course...

Mirek
