Subject: [SOLVED] CtrlLib/Ctrl.iml:41:1: error: expected unqualified-id before '{' token

Posted by fudadmin on Tue, 13 Mar 2012 18:34:53 GMT

View Forum Message <> Reply to Message

This error is killing me.
I want to move my test *.mm files from UNS and SkeletonCoc packages (which compiles fine), bet if I add any new *.mm files to the main testing package which contains *.cpp files, I got this error:
CtrlLib/Ctrl.iml:41:1: error: expected unqualified-id before '{' token
So, most likely, this is related to objc and Draw/iml_header.h. I tried many different combinations of includes and package structuring but... I guess I need to understand the intrinsics of iml_header.h

Combined Posibilities:
1 for objc and/or c++
2 include or exclude some header
3 or define or undef something?

iml_header.h:

```
class IMAGECLASS {
public:
#define IMAGE_BEGIN(n) I_##n,
#define IMAGE_ID(n) I_##n,
 enum {
#include IMAGEFILE
  COUNT
 };
#undef  IMAGE_BEGIN
#undef  IMAGE_ID
...
```

line 41:

```
IMAGE_ID(check)
```

my guess - IMAGE_ID becomes undefined at the wrong moment (when is included/compiles objc (where?)) for some reasons... what reasons?

Need some clues..

Subject: Re: CtrlLib/Ctrl.iml:41:1: error: expected unqualified-id before '{' token

Posted by mr_ped on Tue, 13 Mar 2012 21:47:47 GMT

View Forum Message <> Reply to Message

It's macro, so it may be redefined in mac headers? Did you check preprocessor output?

Subject: Re: CtrlLib/Ctrl.iml:41:1: error: expected unqualified-id before '{' token
Posted by fudadmin on Wed, 14 Mar 2012 00:30:39 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Tue, 13 March 2012 21:47It's macro, so it may be redefined in mac headers? Did you check preprocessor output?

SOLVED! (and maybe this is the only way)

Thanks, mr_ped. Your preprocessor idea directed me towards a victory against my forgetfullness . Btw, your idea had one more side-effect - to discover that theide has got gcc -E built-in! I was aware of gcc -E but never used it. So, the code generated output was big and didn't look suspicious to me. I didn't study it too much. But while exploring I started thinking about *.h files. And then the lightning struck!.. ... Started thinking about other packages.. XCode objc and..

I added one more *.h file to the package and included into TEST.mm -eureka!

So, if you have *.cpp and .mm files in your packages 1(one!) *.h file is not enough! And mostly *.mm and *.h go in pairs.

By accident I have been using that way all the time...
It would be good to understand what is inside the compiler thinking in this case..

Thanks a lot mr_ped once again!