

---

Subject: Strange build error only after changing  
Posted by [cbpporter](#) on Mon, 28 Jan 2008 23:32:44 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi!

I found a really strange build error. I have some sources on the net. I download them, compile, link, everything is OK. I add an inline function to the main .h, still compiles. I use it in one of the .cpp files, and I get:

```
ToolsDropPane.obj : error LNK2005: "public: static class Upp::Iml & __cdecl
CBPImages::Iml(void)" (?Iml@CBPIm
ages@@@SAAV0Upp@@@XZ) already defined in $blitz.obj
ToolsDropPane.obj : error LNK2005: "public: static class Upp::Image __cdecl
CBPImages::Get(int)" (?Get@CBPIma
ges@@@SA?AVImage@Upp@@@H@Z) already defined in $blitz.obj
ToolsDropPane.obj : error LNK2005: "public: static class Upp::Image __cdecl
CBPImages::Get(char const *)" (?G
et@CBPImages@@@SA?AVImage@Upp@@@PBD@Z) already defined in $blitz.obj
ToolsDropPane.obj : error LNK2005: "public: static int __cdecl CBPImages::Find(class Upp::String
const &)" (?
Find@CBPImages@@@SAHABVString@Upp@@@Z) already defined in $blitz.obj
ToolsDropPane.obj : error LNK2005: "public: static int __cdecl CBPImages::Find(char const *)"
(?Find@CBPImage
s@@@SAHPBD@Z) already defined in $blitz.obj
ToolsDropPane.obj : error LNK2005: "public: static void __cdecl CBPImages::Set(int,class
Upp::Image const &)"
(?Set@CBPImages@@@SAXHABVImage@Upp@@@Z) already defined in $blitz.obj
ToolsDropPane.obj : error LNK2005: "public: static void __cdecl CBPImages::Set(char const
*,class Upp::Image
const &)" (?Set@CBPImages@@@SAXPBDABVImage@Upp@@@Z) already defined in $blitz.obj
D:\Develop\upp12\out\MSC8.Debug_full.Gui\CBPMain.exe : fatal error LNK1169: one or more
multiply defined symb
ols found
```

Even after pressing undo to remove all changes, these errors persist. Rebuild, clean + rebuild, nothing helps. I've even done a treedif to see, and the files are absolutely identical to their downloaded state. Only happens with MSC8.

I can't post a test case, but do you have any idea how to fix this?

---

---

Subject: Re: Strange build error only after changing  
Posted by [mr\\_ped](#) on Tue, 29 Jan 2008 08:50:17 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Try to switch off blitz and/or "Build / Clean UPPOUT".

Depends which one you value more, the already built objects or usage of BLITZ. (I suggest to do both)

How did you do "clean", through Upp build menu? Maybe some blitz files remain anyway so Clean UPPOUT is safer option (that deletes absolutely everything from build directory).

To me it looks like some .cpp file is compiled both individually and inside the big "blitz" file, so all of it's instantiated variables do get multiple definitions during linking.  
It may be also problem with instantiating variables inside .h/.hpp files.

And finally I would check date/time of all source files, and current time on the machine, if they are bogus (future?) dates, it may cause havoc both to compiler itself and to blitz heuristic (at least so I think).

edit: And on multi-core CPU the multi threaded compilation may hit the problem with future dates too, there was some fix already between 2007.1 and current dev I think? Try to limit compilation to single thread (Setup/Environment/IDE/HYDRA 1 threads), if this helps, check the "changes" forum if you are using older UPP to see when there was some minor fix introduced.  
(But I don't think this is the cause, as this one always did work when the compilation was invoked second time after first failure.)

---

Subject: Re: Strange build error only after changing  
Posted by [cbpporter](#) on Tue, 29 Jan 2008 09:42:27 GMT  
[View Forum Message](#) <> [Reply to Message](#)

I tried everything that you suggested. Set all dates for today, done repeated cleans (normal + UPPOUT) followed by builds, with and without BLITZ, with different values to HYDRA heads, and still no luck. Tested on 2008.1beta and 7.12dev and reproduced on another computer.

At least MINGW works until I can find the problem.

How do you usually set up a complex project structure, with interface split into multiple files/classes, so that it optimally uses BLITZ?

---

Subject: Re: Strange build error only after changing  
Posted by [mr\\_ped](#) on Tue, 29 Jan 2008 11:56:21 GMT  
[View Forum Message](#) <> [Reply to Message](#)

You can't get "\$blitz.obj" with BLITZ off (after clean UPPOUT).  
So you must get different problem in such case.  
Are you sure it's always the same problem?

If yes, use "verbose" ON and check out where is the \$blitz.obj found and what paths are used when MSCC is called. But it is sort of "impossible", so watch out what is really going on.

I can't help with complex project structure (didn't had any "complex" enough, I think TheIDE is the most complex UPP project available with sources right now.)

---

---

Subject: Re: Strange build error only after changing  
Posted by [mirek](#) on Tue, 29 Jan 2008 12:04:38 GMT

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

---

cbpporter wrote on Tue, 29 January 2008 04:42I tried everything that you suggested. Set all dates for today, done repeated cleans (normal + UPPOUT) followed by builds, with and without BLITZ, with different values to HYDRA heads, and still no luck. Tested on 2008.1beta and 7.12dev and reproduced on another computer.

IMO, you have wrong #includes for .iml.

You should include iml\_header.h in header file and iml\_source.h in .cpp.

Quote:

How do you usually set up a complex project structure, with interface split into multiple files/classes, so that it optimally uses BLITZ?

For BLITZ, the best is single header per package included in all .cpp files.

Mirek

---

---

Subject: Re: Strange build error only after changing  
Posted by [cbpporter](#) on Tue, 29 Jan 2008 13:35:52 GMT

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

---

luzr wrote on Tue, 29 January 2008 14:04

IMO, you have wrong #includes for .iml.

You should include iml\_header.h in header file and iml\_source.h in .cpp.

Thank you!

That solved the problem. I even created a separate .cpp only for <Draw/iml\_sources.h>. Since when do you need to include separate files into the .h and .cpp? I never had any problems with just using <Draw/iml.h> since I started using U++.

Quote:

For BLITZ, the best is single header per package included in all .cpp files.

I think it's quite ironic that in order to increase your compilation speed you need to do the exact opposite of what you learned about C++: if you minimize your .h dependencies, your compilation

speed could improve.

Just a question: is BLITZ partially responsible for the big binary size? It would seem that it makes it rather difficult to generate small obj files.

---

Subject: Re: Strange build error only after changing  
Posted by [mirek](#) on Tue, 29 Jan 2008 15:11:50 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

cbpporter wrote on Tue, 29 January 2008 08:35luzr wrote on Tue, 29 January 2008 14:04  
IMO, you have wrong #includes for .iml.

You should include iml\_header.h in header file and iml\_source.h in .cpp.

Thank you!

That solved the problem. I even created a separate .cpp only for <Draw/iml\_sources.h>. Since when do you need to include separate files into the .h and .cpp? I never had any problems with just using <Draw/iml.h> since I started using U++.

iml.h is just doing both \_header and \_source. So you can really use it for a single .cpp file and better put it into .cpp...

Think about \_header as variable declaration ("extern") and \_source as definition.

Quote:

I think it's quite ironic that in order to increase your compilation speed you need to do the exact opposite of what you learned about C++: if you minimize your .h dependencies, your compilation speed could improve.

Actually, it really does not matter that much.... (BLITZ will work with "sparse headers" model quite well too). But it is the most straightforward way.

Quote:

Just a question: is BLITZ partially responsible for the big binary size? It would seem that it makes it rather difficult to generate small obj files.

Yes. That is why for release mode, BLITZ is not recommended.

Actually, I have started to consider inverse-BLITZ technique for release mode - one that would split .cpp files by functions

(In theory, this should work with GCC and Linux with -ffunction-sections. In practice, it does not).

