

---

Subject: BUG?: .cpp local typedefs mess up /w BLITZ\$  
Posted by [kohait00](#) on Sun, 06 Feb 2011 09:16:15 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

hi guys

i'm getting compilation errors when using blitz enabled in a szenario like the one in attachment.

i always thought cpp local typedefs were what they are.  
blitz seems to mess them up intrenally.

compile w/ blitz -> error (i.e. msc9, shared)  
compile w/o blitz -> ok..

same for GCC

any idea?

---

### File Attachments

1) [TypedefTest.rar](#), downloaded 374 times

---

---

Subject: Re: BUG?: .cpp local typedefs mess up /w BLITZ\$  
Posted by [dolik.rce](#) on Sun, 06 Feb 2011 13:48:57 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

kohait00 wrote on Sun, 06 February 2011 10:16hi guys

i'm getting compilation errors when using blitz enabled in a szenario like the one in attachment.

i always thought cpp local typedefs were what they are.  
blitz seems to mess them up intrenally.

compile w/ blitz -> error (i.e. msc9, shared)  
compile w/o blitz -> ok..

same for GCC

any idea?  
Hi kohati,

You are right, typedef should work only in scope where it is defined, that is single file if it is at global level. The problem is how the files are combined when using BLITZ. Since typedef can't be "undone", it is impossible for BLITZ to solve such situation.

There are two simple solutions you can try (simple==not requiring to rewrite half of your program

- )
- 1) Use macros for the same effect: "typedef int \_MyType\_;" -> "#define \_MyType\_ int"
  - 2) Switch BLITZ off for the files where you use this by adding "#pragma BLITZ\_PROHIBIT"

Hope that helps,  
Honza

---

---

Subject: Re: BUG?: .cpp local typedefs mess up /w BLITZ\$  
Posted by [kohait00](#) on Sun, 06 Feb 2011 18:26:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Thats apity.this has to become a portable package.and the typedef itself is part of a define macro.so i have to rethink the modell.  
Thanks a lot it clears up where to go.

---