Exiv2 - Feature #686

LLVM clang: error: default initialization of an object of const type 'class Exiv2::IptcData const' requires a user-provided default constructor

04 Mar 2010 07:46 - Nikolai Saoukh

Status:	Closed	**Start date:**	04 Mar 2010
Priority:	Low	**Due date:**	
Assignee:		**% Done:**	100%
Category:	miscellaneous	**Estimated time:**	0.00 hour
Target version:	0.20		

Description

Recent clang issues the error message from subject. All the same messages are in the attachment. I am not sure it is an error.

Associated revisions

Revision 2047 - 04 Apr 2010 08:57 - Andreas Huggel

#686: Removed const-qualifier to make things comply with the C++ standard.

History

#1 - 04 Mar 2010 20:13 - Andreas Huggel

Does the following code cause the same error (it works fine with g++)?

```
struct A {
    A() {}
};

class B {
    A a_;
};

int main()
{
    const B b;
}
```

#2 - 04 Mar 2010 21:37 - Nikolai Saoukh

Yes, the same error message for 'const B b;'.

#3 - 04 Mar 2010 22:43 - Andreas Huggel

I don't see anything wrong with that code, but I'm admittedly not particularly fluent with the standard. It would be interesting to hear what the people at LLVM/clang have to say about this one. Pls share a pointer here if you ask them.

#4 - 04 Mar 2010 22:46 - Andreas Huggel

"Fixing" the Exiv2 code should be simple: just remove the 'const'-qualifier in the lines that cause the error.

#5 - 04 Mar 2010 22:55 - Nikolai Saoukh

I will file bug report for clang (still moving target). There are some clang crashes with your code too.

Stay tuned ;-)

#6 - 02 Apr 2010 22:49 - Andreas Huggel

Did you get any feedback from the clang developers?

#7 - 04 Apr 2010 04:19 - Nikolai Saoukh

LLVM developer Chandler Carruth <chandlerc@gmail.com> replied:

> ccd.cpp:12:10: error: default initialization of an object of const type 'B
> const' requires a user-provided default constructor

Clang is correct: C++'03 [dcl.init] p9: If no initializer is specified for an
object, and the object is of (possibly cv-qualified) non-POD class type (or
array thereof), the object shall be default-initialized; if the object is of
const-qualified type, the underlying class type shall have a user-declared
default constructor.

The user declared default constructor for A makes A non-POD. The A member in B
makes B non-POD, and thus this rule kicks in, and the code provided is invalid.

You could re-open as a request for a GCC-compatibility setting to allow such
code, but I don't personally see a lot of value in it. Perhaps others do, so
feel free to re-open if you want to pursue that path.

#8 - 04 Apr 2010 05:10 - Andreas Huggel

Thanks for the update. Now I also found the clang bugreport: http://llvm.org/bugs/show_bug.cgi?id=6772

#9 - 04 Apr 2010 08:58 - Andreas Huggel

- Category set to miscellaneous

- Status changed from New to Resolved

- Target version set to 0.20

- % Done changed from 0 to 100

Removed const-qualifier.

#10 - 29 May 2010 10:43 - Andreas Huggel

- Status changed from Resolved to Closed

Files

const-init.txt	1.28 KB	04 Mar 2010	Nikolai Saoukh