

Re: Is it standard and practical to use long long types?

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.cpp/2004-04/2849.html

From: Ioannis Vranos (ivr_at_guesswh.at.emails.ru)

Date: 04/15/04

Date: Thu, 15 Apr 2004 16:33:05 +0300

"Dan Pop" <Dan.Pop@cern.ch> wrote in message
news:c5m02f\$5e2\$1@sunnews.cern.ch...

- >
- > *Unfortunately, this is not a matter of opinion. Here are a few hard facts*
- > *about your implementation:*
- >
- > *1. Inconsistent documentation. MANUAL.DOC says that -ANSI is the*
- > *right option for putting the compiler in conforming mode. The online*
- > *help says that you need -ansic for this purpose. Only the latter*
- > *is actually recognised by the compiler, but the result is not a*
- > *conforming compiler.*
- >
- > *2. I couldn't find the implementation's document of conformance (maybe*
- > *I didn't try hard enough).*
- >
- > *An implementation shall be accompanied by a document that defines*
- > *^^^^^^*
- > *all implementation-defined and locale-specific characteristics*
- > *^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^*
- > *and all extensions.*

The above two are personally unimportant to me.

- > *3. Dirty headers. Non-standard functions are declared even when the*
- > *compiler is invoked with the extensions disabled, breaking correct*
- > *C programs. See the example below.*
- >
- > *4. Bogus/idiotic warnings when all warnings are enabled. See the example*
- > *below.*
- >
- > *5. Badly needed warnings not produced even when all warnings are enabled:*
- >
- > *T:\lcc>type test.c*
- > *#include <stdio.h>*
- >

comp.lang.c++: Re: Is it standard and practical to use long long types?

```
> int fileno;
>
> int main()
> {
> printf("hello world %d\n", "bar");
> }
>
> T:\lcc>lcc -ansic -A test.c
> Error test.c: 3 redeclaration of 'fileno' previously declared at
h:\lcc\include
> \stdio.h 149
> Warning test.c: 6 old-style function definition for 'main'
> Warning test.c: 6 missing prototype for 'main'
> Warning test.c: 6 'int main()' is a non-ANSI definition
> 1 errors, 3 warnings
>
> Let's look at each diagnostic:
>
> Error test.c: 3 redeclaration of 'fileno' previously declared at
h:\lcc\include\stdio.h 149
>
> stdio.h has no business to declare an identifier "fileno" when the
> compiler is invoked in conforming mode. Chapter and verse available.
>
> Warning test.c: 6 old-style function definition for 'main'
>
> This one is OK. Except for the fact that the line number is wrong: main
> id defined on line 5.
>
> Warning test.c: 6 missing prototype for 'main'
>
> I can't see any call to main() in my program, so why should I provide
> a prototype for it? And I've been already chastised for using an
> old-style definition for it, right?
>
> Warning test.c: 6 'int main()' is a non-ANSI definition
>
> That's BULLSHIT, Jacob. 'int main()' *is* an ANSI definition. It's even
> a C99-conforming definition for the main function. If you don't
> believe me, ask in comp.std.c.
```

The above are indeed serious but have you to be insulting? Are you at war or something? There is no need to hurt our feelings.

```
> Now, for the missing warnings:
>
> 1. The printf call is obviously wrong, yet the compiler has no objection.
```

Well it could give a warning but i do not think it could be signaled as error. However your 3,4,5 are indeed serious.

Re: Is it standard and practical to use long long types?

comp.lang.c++. Re: Is it standard and practical to use long long types?

- > 2. *main()* is defined as returning int, but it doesn't return anything at all. Methinks a warning is badly needed.

Nope. Valid C99 behaviour.

- > *Imagine that YOU were discovering all these things when evaluating someone else's work. What would you think?*

Thinking is one thing, criticising (and better suggesting) is another thing. Insulting is a third thing! This guy is trying to create a useful thing for free, i think he should be encouraged to continue with the right direction. You also are trying to do a good thing here by making useful suggestions, but you are doing it in the wrong way!

- > *For reference, this what I get from gcc, when invoked in conforming mode and with additional warnings enabled:*
- >
- > *fangorn:/tmp/lcc 387> gcc -ansi -pedantic -Wall test.c*
- > *test.c: In function `main':*
- > *test.c:7: warning: int format, pointer arg (arg 2)*
- > *test.c:8: warning: control reaches end of non-void function*

Yours must be old. Mine:

```
C:\c>\mingw\bin\gcc -std=c99 -pedantic-errors -O3 -Wall temp.c -o temp
temp.c: In function `main':
temp.c:7: warning: int format, pointer arg (arg 2)
```

```
C:\c>
```

- > *No bullshit and only the really objectionable "features" of my program are reported. Do you understand now why your compiler looks like a bad joke in the eyes of a competent professional, who expects high quality tools and not toys, even when they are free?*

Behaviour is **very important** if someone does not want to live alone in another planet.

- > *>No institution has supported lcc-win32. It is*
- > *>a user supported project. There is no GNU,*
- > *>nor Stallman, nor Microsoft.*
- >
- > *This is a lame excuse for not doing the right thing. It doesn't take more*
- > *effort to get things right, you just have to think more seriously about*
- > *what you're doing. I refuse to believe that fixing all the issues I've*
- > *mentioned above takes herculean efforts.*

Again nice effort in the wrong way.

- > *>Not even a simple sentence like*
- > *>"lcc-win32 supports long long" will go unnoticed.*

Re: Is it standard and practical to use long long types?

comp.lang.c++: Re: Is it standard and practical to use long long types?

>
> *You have a very strange attitude toward bug reports. Yes, it was meant
> as a bug report, because it is not done the right way: -ansic should put
> the compiler in the only conforming mode it can current support (C89)
> and the C99 support should be available *only* when the compiler is
> invoked with extensions enabled. You can also add an additional switch,
> say -c99, that enables all the C99 features currently supported and
> disables other extensions. Again, you can use gcc as an example of
> how to get it right: gcc -std=c89 vs gcc -std=c99. Don't be afraid to
> look at what other people working on similar projects are doing.*

GCC does not fully support C99, yet ansi invokes it's C99 spirit.

> *On the contrary, you're adopting a paranoid attitude when people report
> them to you, as proved by this very subthread...*

If someone suggested something to you and at the meantime was calling you d**khead, wouldn't you get angry? I think that you don't understand that others get your words more seriously ad you are missing the fact that the same words from a completely stranger is a heavy insult than when from a close friend.

Ioannis Vranos