Compilers Have Defects Too

(even C++ compilers)

Andrew Paxie

21 August 2018

(revised 23 August 2018)

"Ex Ignorantia Ad Sapientiam; Ex Luce Ad Tenebras"

Trompeloeil

A header only C++ mocking framework

https://github.com/rollbear/trompeloeil

Björn Fahller
(Code Owner)

mailto:bjorn@fahller.se

https://github.com/rollbear

https://playfulprogramming.blogspot.com

Andrew Paxie
(Contributor)

mailto:cpp.scribe@gmail.com

https://github.com/AndrewPaxie

https://blog.andrew.paxie.org

Overview

Supported toolchains:

GCC:	4.8.4 through latest
Clang:	3.5 through latest
MSVC:	VS 2015 Release 3, VS 2017 15.8.0

Modes, where supported:

`-std=c++11`, `-std=c++14`, `-std=c++17`, `-std=c++2a`

C++ Standard Libraries:

`libstdc++-v3`, `libc++`, MSVC C++ Standard Library

Overview

About twenty compiler defects have been reported, so far.

The nature of the defects (all for well-formed code):

- Spurious warning.
- Misleading error or warning message.
- Error on well-formed code.
- Internal Compiler Error (ICE) during compilation.
- Missing language feature from compiler.
- Missing entity from standard library implementation.
- "Bad build" of the toolchain.

Issue 1 - C2066 with type alias declaration in class template

Error on well-formed code

Compiler:	VS 2017 15.7.1
Flags:	`/permissive-`

```
// Must be a class tempate, not a class.
template <typename T>
struct S
{
  using sig = void(int);
};
```

```
error C2066:  cast to function type is illegal
```

Issue 1 - C2066 with type alias declaration in class template

Use `identity` to wrap the function type

```
template <typename T>
struct identity
{
  using type = T;
};

template <typename T>
struct S
{
  using sig = typename identity<void(int)>::type;
};
```

Status: Under investigation

Issue 2 - base class has incomplete type

Error on well-formed code

Compiler: Clang 6.0
Flags: `-std=c++17` or `-std=c++2a`

```
#include <type_traits>

struct S {
  template <typename V,
    typename = std::enable_if_t<
      std::is_copy_constructible<V>::value>>
  operator V&() const;
};
static_assert(std::is_constructible<S,const S&>{});
```

```
include/c++/7.3.0/type_traits:1142:14:
error:  base class has incomplete type
: public is_constructible<_Tp, const _Tp&>
```

Issue 2 - base class has incomplete type

Explicitly add another `static_assert`:

```
#include <type_traits>

struct S {
  template <typename V,
    typename = std::enable_if_t<
      std::is_copy_constructible<V>::value>>
  operator V&() const;
};
static_assert(std::is_copy_constructible<S>{});
static_assert(std::is_constructible<S,const S&>{});
```

Status: Unconfirmed

Issue 3 - Ubuntu 17.10 compilers and libraries broken

Bad build of the toolchain

Libraries broken on Ubuntu 17.10 (Artful Aardvark):

`libstdc++-v3`	for GCC 4.8.5
`libstdc++-v3`	for GCC 5.5.0
`libc++`	for Clang 3.9.1
`libc6`	for g++-4.8, g++-5

GCC 4.8.5, GCC 5.5.0:	`std::to_string()` not defined
libc++:	Error including `/usr/include/xlocale`
G++ 4.8.5, G++ 5.5.0:	`signbit()` macro fails to compile

Issue 3 - Ubuntu 17.10 compilers and libraries broken

- Hack files in-place until it all works.
  (symlink `xlocale`, `patch <math.h>`)
  `Provide macros on the compiler command line (_GLIBCXX_USE_C99).`
- `Compile GCC 4.8.5, 5.5.0 from source, or use copies from Compiler Explorer.`
- `Use GCC 6.x, GCC 7.x, Clang 4.0, Clang 5.0.`

Status: Fixed in Ubuntu 18.04 LTS (Bionic Beaver)

Issue 4 - ICE with decltype

Internal Compiler Error

Compiler: VS 2017 RC

```
template <typename... U>
using param_t = decltype(
  std::make_tuple(std::declval<U>()...));
```

```
fatal error C1001:
An internal error has occurred in the compiler.
(compiler file ’msc1.cpp’, line 1469)
```

Issue 4 - ICE with decltype

Introduce a helper metafunction

```
template <typename... U>
struct param_helper {
  using type = decltype(
    std::make_tuple(std::declval<U>()...));
};

template <typename... U>
using param_t = typename param_helper<U...>::type;
```

Status: Under investigation

Issue 5 - MSVC traditional preprocessor

Missing language feature from compiler

Compiler: VS 2015 Update 3 through VS 2017 15.7.x

MSVC "traditional" preprocessor has to be worked around

Status: Fixed in VS 2017 15.8.0 (`/experimental:preprocessor`)

Issue 6 - <regex> not fully implemented

Missing entity from standard library implementation

Compiler: GCC 4.8.x

Yet the API is defined and successfully links.

As Jonathan Wakely said, on 23 June 2014,

> **sigh* <regex> is not implemented prior to GCC 4.9.0,*
> *I thought the whole world was aware of that by now.*

Status: Fixed in GCC 4.9.0

Issue 7 - Unreachable code warning

Spurious warning

Compiler:	VS 2017
Build:	Release

```
template <typename T>
R operator()(T& p)
{
  h(p);         // Always throws
  return R(); // C4702 here
}
```

```
C4702:  unreachable code
```

Issue 7 - Unreachable code warning

Insert an unnecessary try/catch block:

```
template <typename T>
R operator()(T& p)
{
  try
  {
    h(p);     // Always throws
  }
  catch (...)
  {
    throw;
  }
  return R();
}
```

Status: Unknown, but seems to be a Release build feature.

Issue 8 - Ambiguous type name unhelpful warning

Misleading error or warning message

Compiler: GCC 7.x

```
namespace A { class Foo {}; }
namespace B { class Foo {}; }

using namespace A;
using namespace B;

template <typename A> using t = T;

using type = t<Foo>;
```

```
error: template argument 1 is invalid
```

Issue 8 - Ambiguous type name unhelpful warning

Use multiple compilers when error messages with one are unhelpful. E.g. Clang 3.9 is more informative in this context.

```
error:  reference to ’Foo’ is ambiguous

note:  candidate found by name lookup is ’A::Foo’
    class Foo {};

note:  candidate found by name lookup is ’B::Foo’
    class Foo {};
```

Status: Confirmed (not Fixed or Closed)

Conclusion

> *You know what they say, if you're not filing compiler bug reports, you're not doing C++ properly ;-)*
>
> – Björn Fahller

Conclusion

Test Relentlessly

Help yourself and others by reporting the defects you find.
They soon become easier to prepare.

Do it now

https://gcc.gnu.org/bugzilla

https://bugs.llvm.org

https://developercommunity.visualstudio.com/spaces/62/index.html

References - Overview

cppreference.com, ”feature test macros,” last modified 14 August 2018.

Available: https://en.cppreference.com/w/User:D41D8CD98F/feature_testing_macros

Accessed: 19 August 2018

cppreference.com, ”Feature Test Recommendations,” last modified 12 August 2018.

Available: https://en.cppreference.com/w/cpp/experimental/feature_test

Accessed: 19 August 2018

Clark Nelson, ”Feature Testing Recommendations in C++,” P0096R5, 9 October 2017.

Available: https://wg21.link/p0096

Accessed: 19 August 2018

References - Issue 1

Andrew Paxie, "Issue #88 - Error C2066 with VS 2017 15.7.1," 20 May 2018.
Available: https://github.com/rollbear/trompeloeil/issues/88
Accessed: 20 August 2018

Andrew Paxie, "C2066 with type alias declaration in class template," 19 May 2018.
Available: https://developercommunity.visualstudio.com/content/problem/256712/c2066-with-type-alias-declaration-in-class-templat.html
Accessed: 17 August 2018

Timur Doumler, "The identity metafunction," P0887R1, 18 March 2018.
Available: http://wg21.link/p0887
Accessed: 17 August 2018

Andrew Paxie, "G++ 4.8.x limitations", in "Backward compatibility with earlier versions of C++".
Available: https://github.com/rollbear/trompeloeil/blob/master/docs/Backward.md#gxx48x_limitations
Accessed: 17 August 2018

Microsoft Corporation, "/permissive- (Standards conformance)," last modified 21 June 2018.

Available: https://docs.microsoft.com/en-us/cpp/build/reference/permissive-standards-conformance

Accessed: 23 August 2018

References - Issue 2

Andrew Paxie, "Issue #94 - compiling_tests fails with clang++-6.0 -std=c++17," 28 June 2018.

Available: https://github.com/rollbear/trompeloeil/issues/94

Accessed: 20 August 2018

LLVM Bugzilla, "Bug 38010 - ICE on SFINAEd conversion operator," 1 July 2018.

Available: https://bugs.llvm.org/show_bug.cgi?id=38010

LLVM Bugzilla, "Bug 38033 - Templated conversion operator fails construction test unless std::is_copy_constructible is tested," 3 July 2018.

Available: https://bugs.llvm.org/show_bug.cgi?id=38033

References - Issue 3

Andrew Paxie, "Issue #63 - Things to know about compiling Trompeloeil on Ubuntu 17.10," 22 October 2017.

Available: https://github.com/rollbear/trompeloeil/issues/63

Andrew Paxie, "libstdc++-4.8-dev:amd64 4.8.5-4ubuntu6 configured without _GLIBCXX_USE_C99," 21 October 2017.

Available: https://bugs.launchpad.net/ubuntu/+source/gcc-4.8/+bug/1725847

Andrew Paxie, "libstdc++-5-dev:amd64 5.5.0-1ubuntu1 configured without _GLIBCXX_USE_C99," 21 October 2017.

Available: https://bugs.launchpad.net/ubuntu/+source/gcc-5/+bug/1725848

Andrew Paxie, "libc++-dev:amd64 3.9.1-3 expects /usr/include/xlocale.h to be installed," 21 October 2017.

Available: https://bugs.launchpad.net/ubuntu/+source/libc++/+bug/1725858

Andrew Paxie, "libc6-dev:amd64 (2.26-0ubuntu2) has unusable signbit for C++ programs," 22 October 2017.

Available: https://bugs.launchpad.net/ubuntu/+source/glibc/+bug/1725869

References - Issue 4

Björn Fahller, ”Issue #29 - VS 2017RC compilation errors,” 28 January 2017.

Available: https://github.com/rollbear/trompeloeil/issues/29

mlimber, ”Internal error in the compiler when building heavily C++14-targeted mocking framework Trompeloeil,” 8 March 2017.

Available: https://developercommunity.visualstudio.com/content/problem/25323/internal-error-in-the-compiler-when-building-heavi.html

References - Issue 5

Simon Kagstrom, "Issue #1 - Lower requirements to C++11?" 31 December 2014.

Available: https://github.com/rollbear/trompeloeil/issues/1

Bat-Ulzii Luvsanbat, "MSVC Preprocessor Progress towards Conformance," July 6, 2018.

Available: https://
//blogs.msdn.microsoft.com/vcblog/2018/07/06/msvc-preprocessor-progress-towards-conformance/

References - Issue 6

Simon Kagstrom, "Issue #1 - Lower requirements to C++11?" 31 December 2014.

Available: https://github.com/rollbear/trompeloeil/issues/1

GCC Bugzilla, "Bug 61582 - C++11 regex memory corruption," 23 June 2014.

Available: https://gcc.gnu.org/bugzilla/show_bug.cgi?id=61582

Accessed: 9 November 2017

References - Issue 7

JohnboyJovi, "Issue #69 - "Warning in Visual Studio Release build when mocking return struct," 14 December 2017.

Available: https://github.com/rollbear/trompeloeil/issues/69

References - Issue 8

mattgodbolt, ”Issue #25 - Ambiguous type name leads to unusual compiling error,” 12 January 2017.

Available: https://github.com/rollbear/trompeloeil/issues/25

GCC Bugzilla, ”Bug 79070 - Unhelpful error message for ambiguous type in template parameter,” 12 January 2017.

Available: https://gcc.gnu.org/bugzilla/show_bug.cgi?id=79070

Accessed: 19 August 2018

Trompeloeil

https://github.com/rollbear/trompeloeil

Björn Fahller
(Code Owner)

mailto:bjorn@fahller.se

https://github.com/rollbear

https://playfulprogramming.blogspot.com

Andrew Paxie
(Contributor)

mailto:cpp.scribe@gmail.com

https://github.com/AndrewPaxie

https://blog.andrew.paxie.org