

```
daniel@macbook ~ ~/git/presentations/How_to_read_template_error_messages master g++ -c sample_stream.cpp -Wall -std=c++17 -ftemplate-backtrace-limit=0
sample_stream.cpp:20:11: error: invalid operands to binary expression ('std::ostream' and 'const std::vector<std::string>')
    s << arg;
    ^ ~~~
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<0, std::vector<std::string>, std::ostream>::operator()' requested here
    return indented_printer<level - 1, Arg, Stream>()(s, arg);
    ^
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<1, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<2, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<3, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<4, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<5, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<6, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<7, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<8, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<9, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<10, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<11, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<12, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<13, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<14, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<15, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<16, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<17, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<18, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:11:16: note: in instantiation of member function 'indented_printer<19, std::vector<std::string>, std::ostream>::operator()' requested here
sample_stream.cpp:27:5: note: in instantiation of member function 'indented_printer<20, std::vector<std::string>, std::ostream>::operator()' requested here
    indented_printer<level, Arg, Stream>()(s, arg);
    ^
sample_stream.cpp:33:5: note: in instantiation of function template specialization 'printIndented<20, std::vector<std::string>, std::ostream>' requested here
    printIndented<20>(std::cout, v);
    ^
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/c++/v1/cstdint:125:3: note: candidate function template not viable: no known conversion from 'std::ostream' to
```

# How to read template error messages

---

C++ User Group Aachen, 2024-04-10  
Daniel Evers

# Templates are nice ...

---

- Generic programming
- Code reuse (functions, classes, ...)
- Very powerful
- Compile-time magic

```
#include <iostream>
#include <string>
#include <vector>

template<int level, typename Arg, typename Stream>
struct indented_printer
{
    void operator()(Stream& s, const Arg& arg)
    {
        s << ' ';
        return indented_printer<level - 1, Arg, Stream>()(s, arg);
    }
};

template<typename Arg, typename Stream>
struct indented_printer<0, Arg, Stream>
{
    void operator()(Stream& s, const Arg& arg)
    {
        s << arg;
    }
};

template<int level, typename Arg, typename Stream>
void printIndented(Stream& s, const Arg& arg)
{
    indented_printer<level, Arg, Stream>()(s, arg);
}
```

```
#include <iostream>
#include <string>
#include <vector>

template<int level, typename Arg, typename Stream>
struct indented_printer
{
    void operator()(Stream& s, const Arg& arg)
    {
        s << ' ';
        return indented_printer<level - 1, Arg, Stream>()(s, arg);
    }
};

template<typename Arg, typename Stream>
struct indented_printer<0, Arg, Stream>
{
    void operator()(Stream& s, const Arg& arg)
    {
        s << arg;
    }
};

template<int level, typename Arg, typename Stream>
void printIndented(Stream& s, const Arg& arg)
{
    indented_printer<level, Arg, Stream>()(s, arg);
}

int main()
{
    std::vector<std::string> v;
    printIndented<20>(std::cout, v);
    return 0;
}
```

# Compiler error messages

---

**GCC 4.1:** 13 KB, 41 lines

**GCC 13.2:** 24 KB, 236 lines

**MSVC 19.38:** 85 KB, 1785 lines

-> sample\_stream.cpp\_gcc4.log  
sample\_stream.cpp\_gcc13.log  
sample\_stream.cpp\_msvc1938.log

# Quantity vs. Quality

---

- **GCC 4: shorter, but less useful than GCC 13**
- Longer “prolog” listing all instantiations
- My code is in line 24 :-/
- Shows mismatching operators, but not why they don’t match

# Quantity vs. Quality

---

- MSVC:**
- Doesn't show my code at all, only the line number**
- Very verbose (but helpful)**

# Quantity vs. Quality

---

- What's generally bad:
  - Loooong path names
  - Loooong template names

Can we make this more  
readable?

# GCC options

---

- -fno-pretty-templates
  - Omit default template arguments for class specializations
- -ftemplate-backtrace-limit=n
  - Maximum number of template instantiation notes for a single warning/error
- -fdiagnostics-show-template-tree
  - Print a tree-like structure comparing mismatching template types
  - Doesn't make a difference in our example :-/

# MSVC options

---

???



[home](#) | [talks](#) | [projects](#) | [tutorials](#) | [stuff](#)

## about me

Hello! My name is Vittorio.

I'm a modern C++ enthusiast who loves to share his knowledge by creating video tutorials and participating to conferences.

I have a BS in Computer Science from the *University of Messina*. I write libraries, applications and games.

Check out my GitHub page and feel free to contact me if you're interested in my projects.

Please consider donating if you enjoy my work.

amomilla: C++ error simplification script

8 september 2016

[c++14](#) [python](#) [script](#) [error](#) [templates](#) [article](#)

I've recently released a script on my GitHub page that simplifies C++ compiler error messages: [camomilla](#).

# What does it do?

`amomilla` uses simple text transformations to make `gcc` and `clang` errors smaller and easier to read. During the development of `ecst`, a compile-time *Entity-Component-System* C++14 library developed for [my BCS thesis](#), I encountered a lot of huge undecipherable errors that sometimes completely filled my terminal buffer. Here's an example:

**Before:**

- Before:

g++: before camomilla

- After:

```
[vittorio@orion:~/workspace/ecst$ example]$ cer ./pres_code.cpp |> canonville -d8  
False  
/pres_code.cpp: In instantiation of 'example::s::keep_in_bounds::process(Omnia) [with EntityID = ecst::entity::Id; TData = ecst::content::system::data::Proxy::MultiType]':  
/home/vittoriooromo/00Workspace/ecst/include/ecst/include/system/instance/instance_sub.h:17: required from 'void ecst::content::system::instances<T>::for_entities(Omnia::ResourceSet, Omnia::ResourceSet, TRule)' [with TRule = example::s::keep_in_bounds::process(Omnia)]  
/home/vittoriooromo/00Workspace/ecst/include/ecst/include/system/data/impl.h:13: required from 'TSet<Omnia::ResourceSet> TResourceSignature::listResourceSets<T>(long unsigned int)'  
/home/vittoriooromo/00Workspace/ecst/include/ecst/include/system/data/impl.h:13: required from 'void ecst::content::system::data::Proxy::MultiType::for_entities(Omnia)' [with TRule = example::s::keep_in_bounds::process(Omnia)] [with TRule = ecst::content::system::instances<T>]  
/pres_code.cpp:24:17: required from 'void ecst::content::system::data::Proxy::MultiType::process(Omnia)' [with TRule = ecst::content::system::data::Proxy::MultiType]  
/pres_code.cpp:87:20: required from 'example::update_cts() [Context, TContext]' [with EntityID = ecst::entity::Id; TContent = ecst::content::system::data::Proxy::MultiType; auto151 = ecst::content::impl::step::Proxy<T>; TContent = ecst::content::impl::detect<T>; TResourceTarget = ecst::content::impl::detect<T>]  
/home/vittoriooromo/00Workspace/ecst/include/ecst/include/system/executor_adapter.h:101:1: predicate_header/predicate_header.inl:9:38: required from 'ecst::system::executor_adapter::impl::predicate_header<T>' [with auto152 = ecst::content::system::data::Proxy::MultiType; auto153 = long unsigned int; keep_in_bounds::auto154 = example::s::keep_in_bounds::auto154 = ecst::content::system::data::Proxy::MultiType; auto155 = ecst::content::system::data::Proxy::MultiType; auto156 = ecst::content::impl::step::Proxy<T>; T06Content = ecst::content::impl::detect<T>; TResourceTarget = sf::RenderableHandle<sf::Drawable>]  
/home/vittoriooromo/00Workspace/ecst/include/ecst/include/system/executor_adapter.inl:39: required from 'TSystemFlags = (ecst::tag::system::impl::log_impr<T>, ecst::tag::system::impl::log_impr<T>, ecst::tag::system::impl::log_impr<T>);'  
[skipping 50 instantiation contexts, use -fno-implicit-backtrace-limit= to disable]  
./util/pres_game_app.hpp:200:23: required from 'void example::game::app::TGame::initO [with TContent = ecst::content::impl::detect<T>]  
./util/pres_game_app.hpp:216:17: required from 'example::game::app::TGame::app(sf::RenderWindow &TContent)' [with TContent = ecst::content::impl::detect<T>]  
./util/.../201Template/app_number.hpp:18:28: required from 'example::201Template::app::runner<Ts>::app_number(const string, size_t, size_t, Ts...)<Ts>' [with Ts = {ecst::content::impl::detect<T>; T = example::game::app<T>}]  
./util/.../201Template/app_number.hpp:36:59: required from 'struct example::201Template::app::runner<Ts>::app_number(const string, size_t, size_t, Ts...)' [with Ts = {ecst::content::impl::detect<T>; T = example::game::app<T>; __ord1::string = __ord1::basic_string<T>; size_t = long unsigned int}];  
./util/.../201Template/app_number.hpp:36:25: required from 'example::201Template::app::runner<Ts>::app_number(const string, size_t, size_t, Ts...)' [with Ts = {ecst::content::impl::detect<T>; T = example::game::app<T>; __ord1::string = __ord1::basic_string<T>; size_t = long unsigned int}'  
./util/pres_game_app.hpp:229:99: required from 'void example::trans::initTrans(TContent)' [with TContent = ecst::content::impl::detect<T>]  
./pres_code.cpp:936:24: required from here  
./pres_code.cpp:274:31: error: 'class sf::Vector2f' has no member named 'y'; did you mean 'g'?  
    v.y = -t;  
In file included from /home/vittoriooromo/00Workspace/vn/_core/include/vn/core/assert/assert.hpp:16:8,  
from /home/vittoriooromo/00Workspace/vn/_core/include/vn/core/assert.hpp:11,  
from /home/vittoriooromo/00Workspace/ecst/include/ecst/include/vn/core/assert.hpp,  
from /home/vittoriooromo/00Workspace/ecst/include/ecst/include/vn/core/assert.hpp,  
from /home/vittoriooromo/00Workspace/ecst/include/ecst/include/vn/core/assert.hpp,  
from /home/vittoriooromo/00Workspace/ecst/include/ecst/include/vn/core/assert.hpp,  
from /home/vittoriooromo/00Workspace/ecst/include/ecst.hpp:12,  
from ./util/111s/dependencies.hpp:14,  
from ./pres_code.cpp:7.  
/home/vittoriooromo/00Workspace/vn/_core/include/vn/core/assert/impl/assert.inl:21:10: warning: 'void vn::core::impl::assert::fire(const char*, const TRule, const TRules)' [with TRule = ecst::content::impl::instance<base>; TRules = multiset<T>] used but never defined  
    void fire(const char* charkode, const char* charkline, const char* charkfile,
```

## *g++: after camomilla*

[https://vittoriromeo.info/index/blog/2016\\_camo...](https://vittoriromeo.info/index/blog/2016_camo...)

<https://github.com/vittorioromeo/camomilla>

# Camomilla: features

---

## □ Template typename collapsing

```
echo "metavector<metatype<metawhatever<int>>::method()" | camomilla -d0
# outputs
metavector<?>::method()
```

```
echo "metavector<metatype<metawhatever<int>>::method()" | camomilla -d1
# outputs
metavector<metatype<?>>::method()
```

```
echo "metavector<metatype<metawhatever<int>>::method()" | camomilla -d2
# outputs
metavector<metatype<metawhatever<?>>>::method()
```

```
echo "metavector<metatype<metawhatever<int>>::method()" | camomilla -d3
# outputs
metavector<metatype<metawhatever<int>>>::method()
```

# Camomilla: features

---

## □ Namespace replacement regexes

```
echo "std::vector<std::pair<std::int16_t, std::int32_t>>" | camomilla --depth=100
# outputs
vector<pair<int16_t, int32_t>>

echo "boost::hana::tuple<boost::hana::tuple<boost::hana::int_c<10>, boost::hana::int_c<15>>>"
| camomilla -d100
# outputs
bh::tuple<bh::tuple<bh::int_c<10>, bh::int_c<15>>>
```

# Camomilla: features

---

## □ Generic replacement regexes

```
echo "std::forward<decltype(std::tuple<unsigned long long, std::size_t, int>)(x)"  
    | camomilla -d100  
# outputs  
fwd<decltype(tuple<ulong long, sz_t, int>)(x)
```

## Examples

1. Plain log

2. `camomilla -d 0`

3. `camomilla -d 2`

4. `camomilla.json`

# General strategy

---

- Post-process with camomilla (if necessary)
- GCC/clang:
  - Identify the problematic line in “your code” by searching for “required from here”
  - See above for template instantiations
  - See below for additional infos (attempted conversions)

# General strategy

---

- Post-process with camomilla (if necessary)
- MSVC:
  - Starts with the problematic code line (probably yours)
  - Or search for “note: see reference to function template instantiation ... being compiled”
  - Lists all attempted conversions
  - “the template instantiation context (the oldest one first) is”