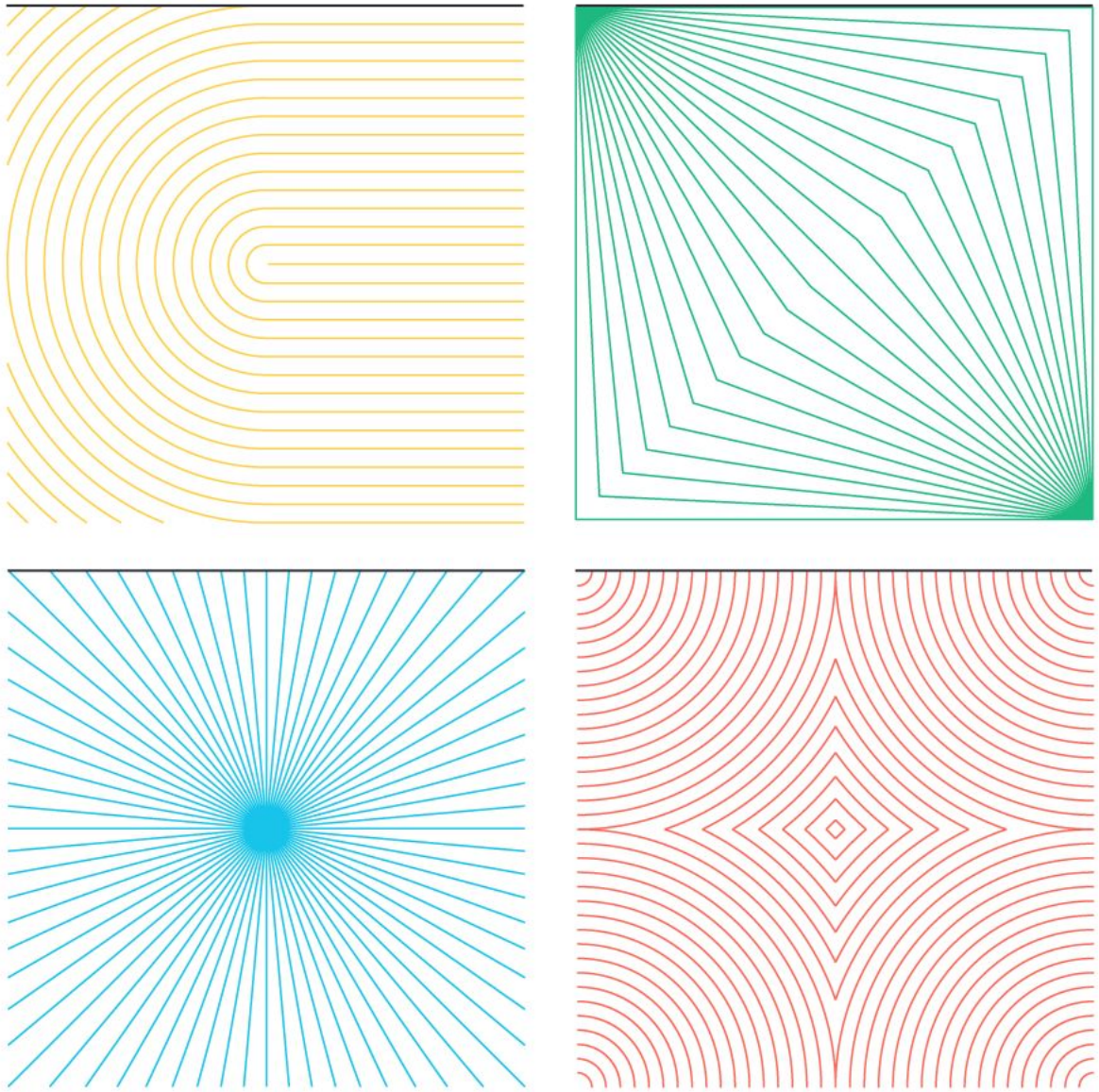
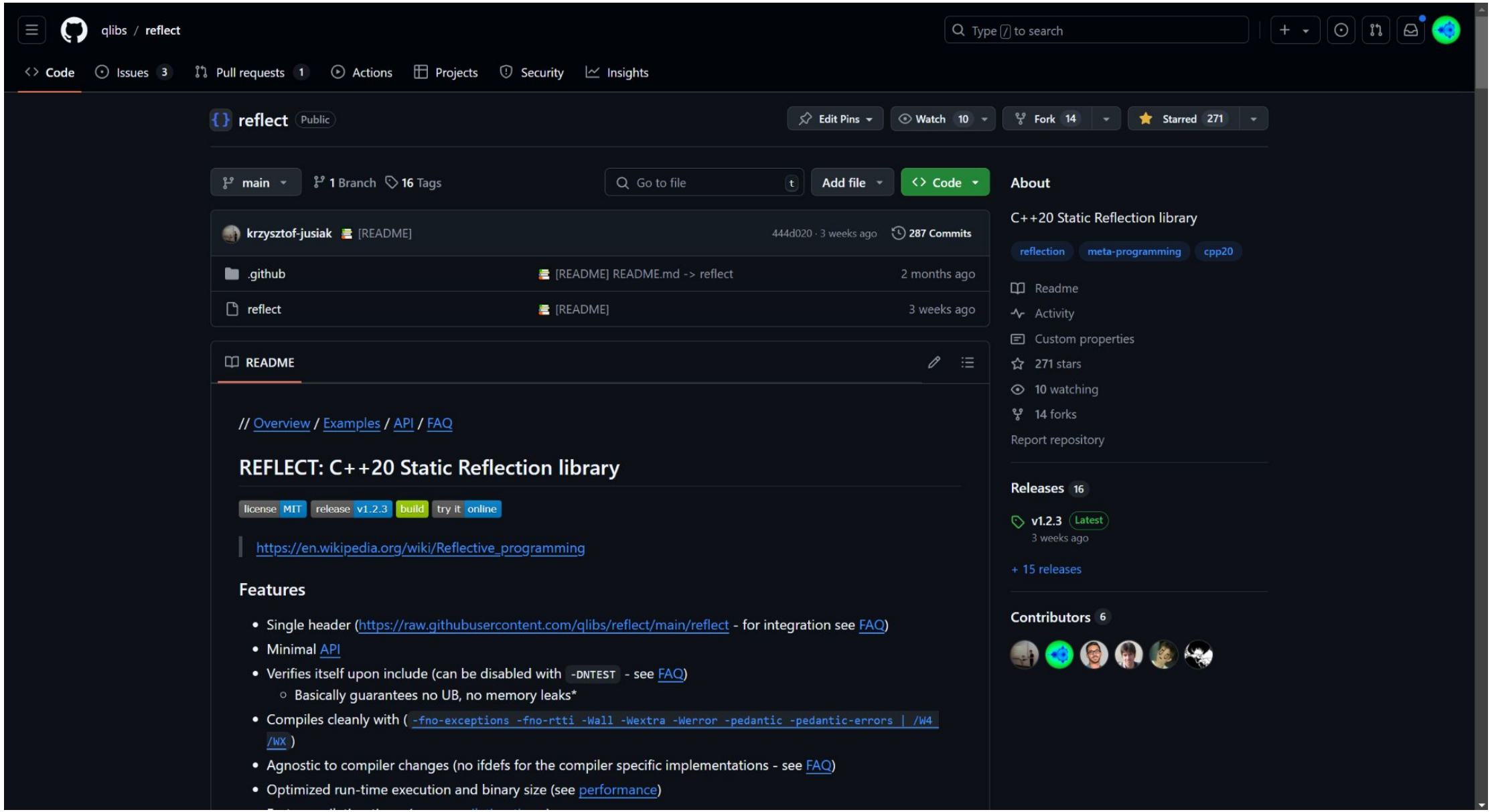
Customizing Compilation Error Messages

Formatting a Compile-Time String During Substitution Failure

Patrick Roberts, P.E.
Principal Software Engineer



What problem does it solve?


qlibs / reflect
Type / to search
Code
Issues 3
Pull requests 1
Actions
Projects
Security
Insights
reflect Public
Edit Pins
Watch 10
Fork 14
Starred 271
main
1 Branch
16 Tags
Go to file
t
Add file
Code
krzysztof-jusiak [README]
444d020 · 3 weeks ago
287 Commits
.github
[README] README.md -> reflect
2 months ago
reflect
[README]
3 weeks ago
README
// Overview / Examples / API / FAQ
REFLECT: C++20 Static Reflection library
license MIT
release v1.2.3
build
try it online
https://en.wikipedia.org/wiki/Reflective_programming
Features
Single header (https://raw.githubusercontent.com/qlibs/reflect/main/reflect - for integration see FAQ)
Minimal API
Verifies itself upon include (can be disabled with -DNTEST - see FAQ)
Basically guarantees no UB, no memory leaks*
Compiles cleanly with (-fno-exceptions -fno-rtti -Wall -Wextra -Werror -pedantic -pedantic-errors | /W4 /WX)
Agnostic to compiler changes (no ifdefs for the compiler specific implementations - see FAQ)
Optimized run-time execution and binary size (see performance)
About
C++20 Static Reflection library
reflection
meta-programming
cpp20
Readme
Activity
Custom properties
271 stars
10 watching
14 forks
Report repository
Releases 16
v1.2.3 Latest
3 weeks ago
+ 15 releases
Contributors 6

```cpp
#include <https://raw.githubusercontent.com/patrickroberts/reflect/d0b4f2a419f750c720005713c39a6e5d8a903b2e/reflect>

struct foo {
  char ch;
  float flt{26.2f};
};

static_assert(reflect::get<"flt">(foo{}) == 26.2f);

```

Output of x86-64 clang (trunk) (Compiler #1)

A Wrap lines Select all

```
Compiler returned: 0
```

```cpp
#include <https://raw.githubusercontent.com/patrickroberts/reflect/d0b4f2a419f750c720005713c39a6e5d8a903b2e/reflect>

struct foo {
  char ch;
  float flt{26.2f};
};

static_assert(reflect::get<"glt">(foo{}) == 26.2f);

```

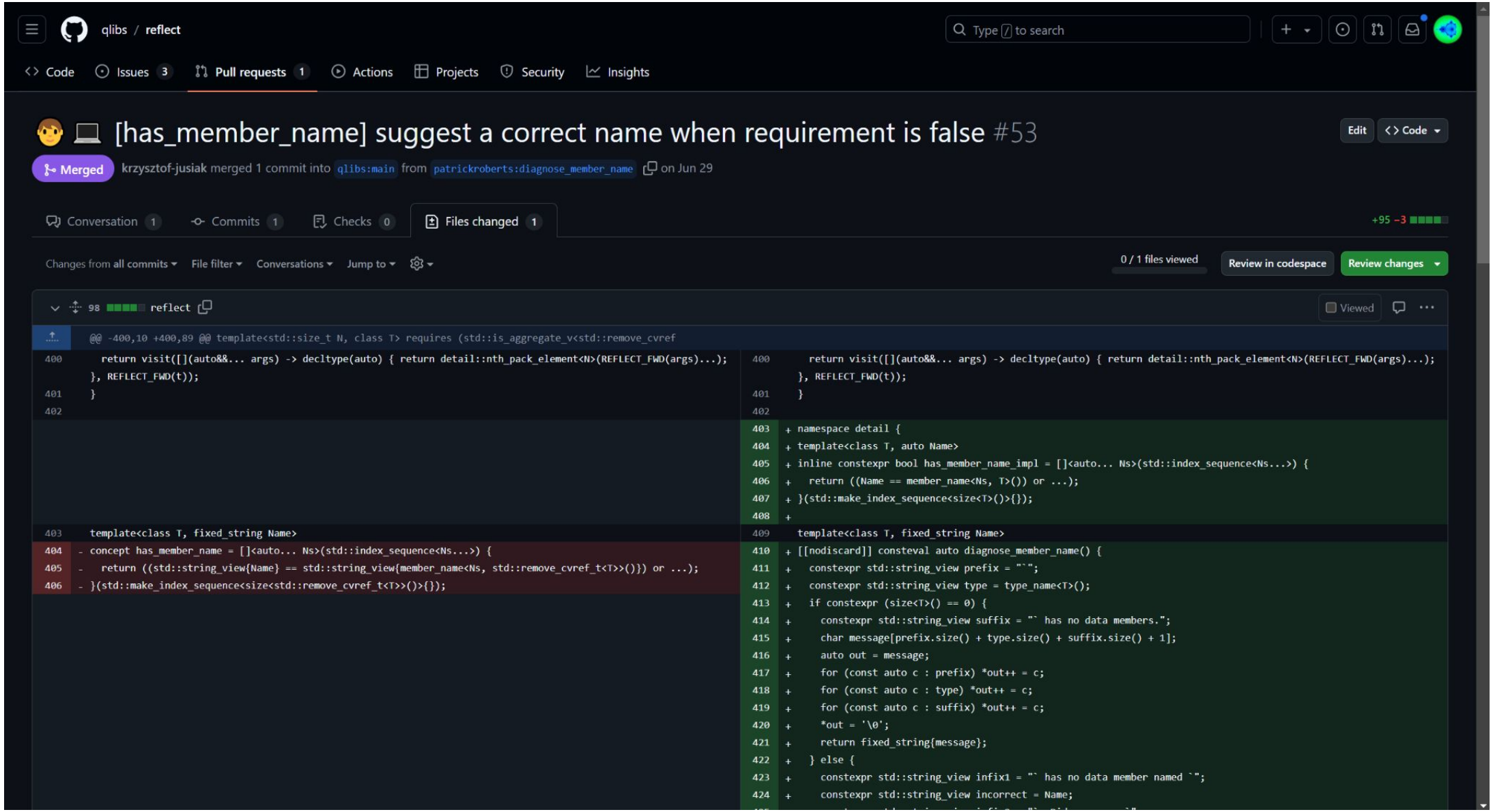
Output of x86-64 clang (trunk) (Compiler #1)

Wrap lines Select all

```
<source>:8:15: error: no matching function for call to 'get'
    8 | static_assert(reflect::get<"glt">(foo{}) == 26.2f);
      |               ^~~~~~~~~~~~~~~~~~
/app/raw.githubusercontent.com/patrickroberts/reflect/d0b4f2a419f750c720005713c39a6e5d8a903b2e/reflect:410:26: note: candidate template ignore
  410 | constexpr decltype(auto) get(T&& t) noexcept {
      |                          ^
/app/raw.githubusercontent.com/patrickroberts/reflect/d0b4f2a419f750c720005713c39a6e5d8a903b2e/reflect:409:61: note: because 'has_member_name<
  409 |   requires (std::is_aggregate_v<std::remove_cvref_t<T>> and has_member_name<std::remove_cvref_t<T>, Name>)
      |                                                             ^
/app/raw.githubusercontent.com/patrickroberts/reflect/d0b4f2a419f750c720005713c39a6e5d8a903b2e/reflect:404:27: note: because '[]<auto ...Ns>(s
    return ((std::string_view{fixed_string<char, 3UL>{"glt"}} == std::string_view{member_name<Ns, std::remove_cvref_t<foo> >()}) || ...);
}(std::make_index_sequence<size<std::remove_cvref_t<foo> >()>{})' evaluated to false
  404 | concept has_member_name = []<auto... Ns>(std::index_sequence<Ns...>) {
      |                           ^
/app/raw.githubusercontent.com/patrickroberts/reflect/d0b4f2a419f750c720005713c39a6e5d8a903b2e/reflect:399:40: note: candidate template ignore
  399 | [[nodiscard]] constexpr decltype(auto) get(T&& t) noexcept {
      |                                        ^
1 error generated.
Compiler returned: 1
```

Guidelines for a helpful error message

Where is the error? ✅
What is wrong? ✅
How can it be fixed? ❌


qlibs / reflect
Type / to search
Code
Issues 3
Pull requests 1
Actions
Projects
Security
Insights
[has_member_name] suggest a correct name when requirement is false #53
Edit
Code
Merged
krzysztof-jusiak merged 1 commit into qlibs:main from patrickroberts:diagnose_member_name on Jun 29
Conversation 1
Commits 1
Checks 0
Files changed 1
+95 −3
Changes from all commits
File filter
Conversations
Jump to
0 / 1 files viewed
Review in codespace
Review changes
98 reflect
Viewed
@@ -400,10 +400,89 @@ template<std::size_t N, class T> requires (std::is_aggregate_v<std::remove_cvref
return visit([](auto&&... args) -> decltype(auto) { return detail::nth_pack_element<N>(REFLECT_FWD(args)...);
}, REFLECT_FWD(t));
}
+ namespace detail {
+ template<class T, auto Name>
+ inline constexpr bool has_member_name_impl = []<auto... Ns>(std::index_sequence<Ns...>) {
+ return ((Name == member_name<Ns, T>()) or ...);
+ }(std::make_index_sequence<size<T>()>{});
+
template<class T, fixed_string Name>
- concept has_member_name = []<auto... Ns>(std::index_sequence<Ns...>) {
- return ((std::string_view{Name} == std::string_view{member_name<Ns, std::remove_cvref_t<T>>()}) or ...);
- }(std::make_index_sequence<size<std::remove_cvref_t<T>>()>{});
+ [[nodiscard]] consteval auto diagnose_member_name() {
+ constexpr std::string_view prefix = "`";
+ constexpr std::string_view type = type_name<T>();
+ if constexpr (size<T>() == 0) {
+ constexpr std::string_view suffix = "` has no data members.";
+ char message[prefix.size() + type.size() + suffix.size() + 1];
+ auto out = message;
+ for (const auto c : prefix) *out++ = c;
+ for (const auto c : type) *out++ = c;
+ for (const auto c : suffix) *out++ = c;
+ *out = '\0';
+ return fixed_string{message};
+ } else {
+ constexpr std::string_view infix1 = "` has no data member named `";
+ constexpr std::string_view incorrect = Name;

```cpp
#include <https://raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect>

struct foo {
  char ch;
  float flt{26.2f};
};

static_assert(reflect::get<"glt">(foo{}) == 26.2f);

```

Output of x86-64 clang (trunk) (Compiler #1)

A Wrap lines Select all

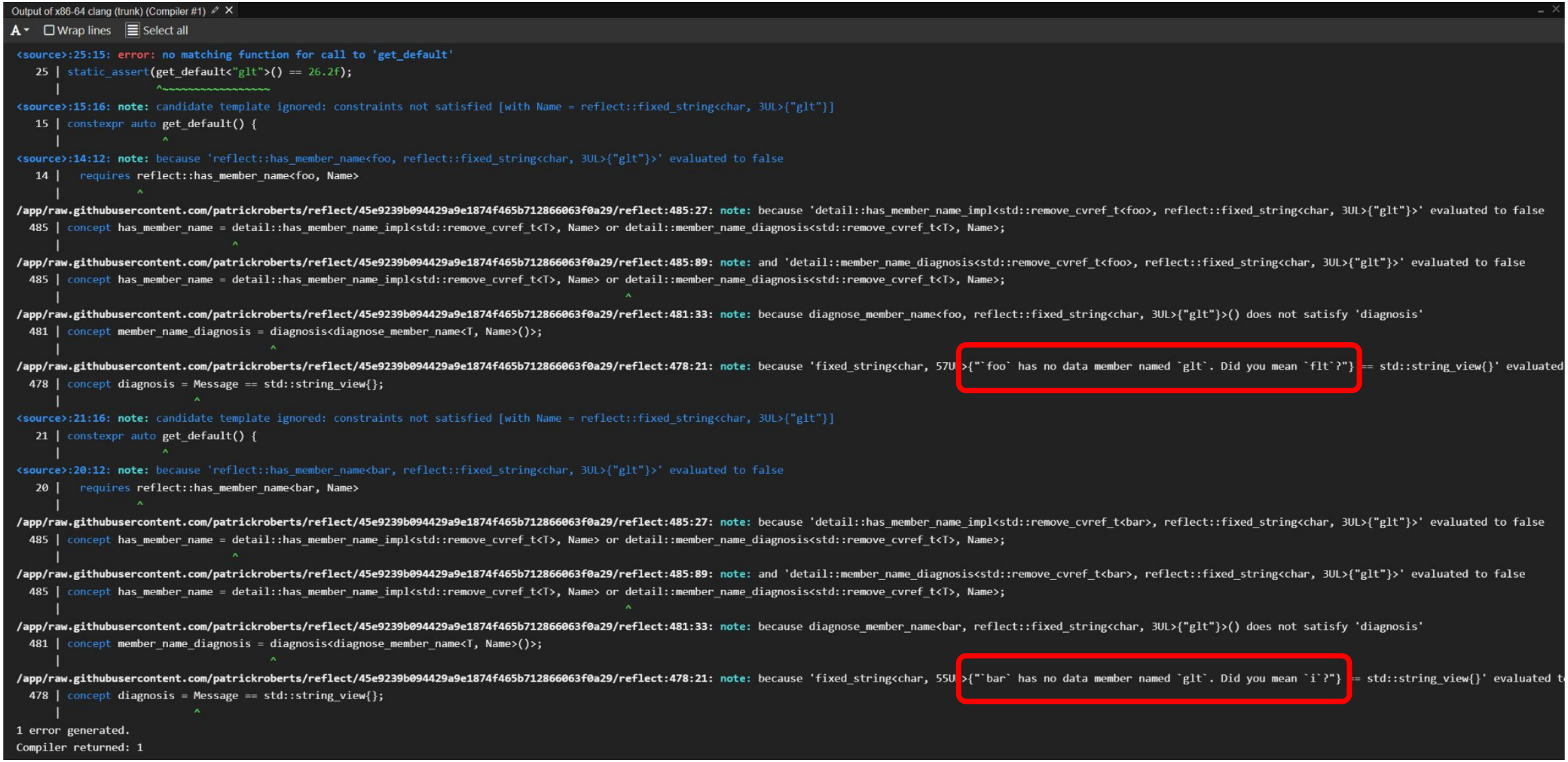
```
<source>:8:15: error: no matching function for call to 'get'
    8 | static_assert(reflect::get<"glt">(foo{}) == 26.2f);
      |               ^~~~~~~~~~~~~~~~~~
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:489:26: note: candidate template ignored: constraints not satisfied [with Name = fixed_string<char, 3UL>{"glt
  489 | constexpr decltype(auto) get(T&& t) noexcept {
      |                          ^
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:488:61: note: because 'has_member_name<std::remove_cvref_t<foo>, fixed_string<char, 3UL>{"glt"}>' evaluated t
  488 |   requires (std::is_aggregate_v<std::remove_cvref_t<T>> and has_member_name<std::remove_cvref_t<T>, Name>)
      |                                                             ^
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:485:27: note: because 'detail::has_member_name_impl<std::remove_cvref_t<foo>, fixed_string<char, 3UL>{"glt"}>
  485 | concept has_member_name = detail::has_member_name_impl<std::remove_cvref_t<T>, Name> or detail::member_name_diagnosis<std::remove_cvref_t<T>, Name>;
      |                           ^
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:485:89: note: and 'detail::member_name_diagnosis<std::remove_cvref_t<foo>, fixed_string<char, 3UL>{"glt"}>' e
  485 | concept has_member_name = detail::has_member_name_impl<std::remove_cvref_t<T>, Name> or detail::member_name_diagnosis<std::remove_cvref_t<T>, Name>;
      |                                                                                         ^
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:481:33: note: because diagnose_member_name<foo, fixed_string<char, 3UL>{"glt"}>() does not satisfy 'diagnosis
  481 | concept member_name_diagnosis = diagnosis<diagnose_member_name<T, Name>()>;
      |                                 ^
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:478:21: note: because 'fixed_string<char, 57UL>{"`foo` has no data member named `glt`. Did you mean `flt`?"}
  478 | concept diagnosis = Message == std::string_view{};
      |                     ^
```

So why not just `static_assert`?

https://wg21.link/p2741 “User-generated static_assert messages”

Answer: It does not result in substitution failure.

```cpp
#include <https://raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect>

struct foo {
  char ch;
  float flt{26.2f};
};

struct bar {
  int i{42};
  double dbl;
};

template <reflect::fixed_string Name>
  requires reflect::has_member_name<foo, Name>
constexpr auto get_default() {
  return reflect::get<Name>(foo{});
}

template <reflect::fixed_string Name>
  requires reflect::has_member_name<bar, Name>
constexpr auto get_default() {
  return reflect::get<Name>(bar{});
}

static_assert(get_default<"flt">() == 26.2f);

```


Output of x86-64 clang (trunk) (Compiler #1)
Wrap lines
Select all
<source>:25:15: error: no matching function for call to 'get_default'
25 | static_assert(get_default<"glt">() == 26.2f);
<source>:15:16: note: candidate template ignored: constraints not satisfied [with Name = reflect::fixed_string<char, 3UL>{"glt"}]
15 | constexpr auto get_default() {
<source>:14:12: note: because 'reflect::has_member_name<foo, reflect::fixed_string<char, 3UL>{"glt"}>' evaluated to false
14 | requires reflect::has_member_name<foo, Name>
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:485:27: note: because 'detail::has_member_name_impl<std::remove_cvref_t<foo>, reflect::fixed_string<char, 3UL>{"glt"}>' evaluated to false
485 | concept has_member_name = detail::has_member_name_impl<std::remove_cvref_t<T>, Name> or detail::member_name_diagnosis<std::remove_cvref_t<T>, Name>;
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:485:89: note: and 'detail::member_name_diagnosis<std::remove_cvref_t<foo>, reflect::fixed_string<char, 3UL>{"glt"}>' evaluated to false
485 | concept has_member_name = detail::has_member_name_impl<std::remove_cvref_t<T>, Name> or detail::member_name_diagnosis<std::remove_cvref_t<T>, Name>;
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:481:33: note: because diagnose_member_name<foo, reflect::fixed_string<char, 3UL>{"glt"}>() does not satisfy 'diagnosis'
481 | concept member_name_diagnosis = diagnosis<diagnose_member_name<T, Name>()>;
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:478:21: note: because 'fixed_string<char, 57U>{"`foo` has no data member named `glt`. Did you mean `flt`?"} == std::string_view{}' evaluated
478 | concept diagnosis = Message == std::string_view{};
<source>:21:16: note: candidate template ignored: constraints not satisfied [with Name = reflect::fixed_string<char, 3UL>{"glt"}]
21 | constexpr auto get_default() {
<source>:20:12: note: because 'reflect::has_member_name<bar, reflect::fixed_string<char, 3UL>{"glt"}>' evaluated to false
20 | requires reflect::has_member_name<bar, Name>
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:485:27: note: because 'detail::has_member_name_impl<std::remove_cvref_t<bar>, reflect::fixed_string<char, 3UL>{"glt"}>' evaluated to false
485 | concept has_member_name = detail::has_member_name_impl<std::remove_cvref_t<T>, Name> or detail::member_name_diagnosis<std::remove_cvref_t<T>, Name>;
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:485:89: note: and 'detail::member_name_diagnosis<std::remove_cvref_t<bar>, reflect::fixed_string<char, 3UL>{"glt"}>' evaluated to false
485 | concept has_member_name = detail::has_member_name_impl<std::remove_cvref_t<T>, Name> or detail::member_name_diagnosis<std::remove_cvref_t<T>, Name>;
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:481:33: note: because diagnose_member_name<bar, reflect::fixed_string<char, 3UL>{"glt"}>() does not satisfy 'diagnosis'
481 | concept member_name_diagnosis = diagnosis<diagnose_member_name<T, Name>()>;
/app/raw.githubusercontent.com/patrickroberts/reflect/45e9239b094429a9e1874f465b712866063f0a29/reflect:478:21: note: because 'fixed_string<char, 55U>{"`bar` has no data member named `glt`. Did you mean `i`?"} == std::string_view{}' evaluated t
478 | concept diagnosis = Message == std::string_view{};
1 error generated.
Compiler returned: 1

Thanks for watching

https://patrickroberts.dev

Special thanks to Kris Jusiak for writing **https://github.com/qlibs/reflect** (and for merging my pull request)