


Как обновить компилятор и не тронуться

Елена Степанова

 st_1ena

 elena.1.stepanova@nokia.com

Packet Core developer

Product owner

Nokia

План

- Почему это больно

План

- Почему это больно
- Какой профит

План

- Почему это больно
- Какой профит
- Что меняется

План

- Почему это больно
- Какой профит
- Что меняется
- Примеры

План

- Почему это больно
- Какой профит
- Что меняется
- Примеры
- Рекомендации

Почему это больно

- Обратная совместимость не работает

Почему это больно

- Обратная совместимость не работает
 - <https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines#S-modernizing>

Почему это больно

- Обратная совместимость не работает
 - <https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines#S-modernizing>
- Рекомендаций, подходящих всем, не существует

Почему это больно

- Обратная совместимость не работает
 - <https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines#S-modernizing>
- Рекомендаций, подходящих всем, не существует
 - об этом мало говорят

Почему это больно

- Обратная совместимость не работает
 - <https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines#S-modernizing>
- Рекомендаций, подходящих всем, не существует
 - об этом мало говорят
 - некоторые и не обновляют, и им нормально

Почему это больно

- Обратная совместимость не работает
 - <https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines#S-modernizing>
- Рекомендаций, подходящих всем, не существует
 - об этом мало говорят
 - некоторые и не обновляют, и им нормально
- Занимает непредсказуемое время

Какой профит

- Новые фичи языка и стандартной библиотеки

Какой профит

- Новые фичи языка и стандартной библиотеки
- Новые компиляторные диагностики

Какой профит

- Новые фичи языка и стандартной библиотеки
- Новые компиляторные диагностики
 - обнаружение багов

Какой профит

- Новые фичи языка и стандартной библиотеки
- Новые компиляторные диагностики
 - обнаружение багов
- Модернизация диалекта кода

Какой профит

- Новые фичи языка и стандартной библиотеки
- Новые компиляторные диагностики
 - обнаружение багов
- Модернизация диалекта кода
- Обновление практик

Какой профит

- Новые фичи языка и стандартной библиотеки
- Новые компиляторные диагностики
 - обнаружение багов
- Модернизация диалекта кода
- Обновление практик
- Обновление 3rd-party

Что меняется

- Интерфейсы

Что меняется

- Интерфейсы
- Поведение

Что меняется

- Интерфейсы
- Поведение
- Требования

Что меняется

- Интерфейсы
- Поведение
- Требования
- Диагностики

Что меняется

- Интерфейсы
- Поведение
- Требования
- Диагностики

<https://www.gnu.org/software/gcc/projects/cxx-status.html>

<https://gcc.gnu.org/onlinedocs/libstdc++/manual/status.html>

<https://clang.llvm.org/compatibility.html>

Что меняется

en.cppreference.com/w/cpp/memory/auto_ptr

std::auto_ptr

Defined in header `<memory>`

<code>template< class T > class auto_ptr;</code>	(1)	(deprecated in C++11) (removed in C++17)
<code>template<> class auto_ptr<void>;</code>	(2)	(deprecated in C++11) (removed in C++17)

en.cppreference.com/w/cpp/memory/allocator/allocate

std::allocator<T>::allocate

<code>pointer allocate(size_type n, const void * hint = 0);</code>		(until C++17)
<code>T* allocate(std::size_t n, const void * hint);</code>	(1)	(since C++17) (deprecated) (removed in C++20)
<code>T* allocate(std::size_t n);</code>	(2)	(since C++17) (until C++20)
<code>[[nodiscard]] constexpr T* allocate(std::size_t n);</code>		(since C++20)

en.cppreference.com/w/cpp/algorithm/swap

std::swap

Defined in header <code><algorithm></code>	(until C++11)
Defined in header <code><utility></code>	(since C++11)
Defined in header <code><string_view></code>	(since C++17)

Example

Run Share Exit GCC 9.2 (C++2a)

Powered by Coliru online compiler

```
1 #include <iostream>
2 #include <locale>
3 #include <cstdio>
4
5 int main()
6 {
7     int i, j;
8     float x, y;
9     char str1[10], str2[4];
10    wchar_t warr[2];
11    std::setlocale(LC_ALL, "en_US.utf8");
12
13    char input[] = u8"25 54.32E-1 Thompson 56789 0123 56水";
14    // parse as follows:
15    // %d: an integer
16    // %f: a floating-point value
17    // %9s: a string of at most 9 non-whitespace characters
18    // %2d: two-digit integer (digits 5 and 6)
19    // %f: a floating-point value (digits 7, 8, 9)
20    // %*d an integer which isn't stored anywhere
21    // ' ': all consecutive whitespace
22    // %3[0-9]: a string of at most 3 digits (digits 5 and 6)
23    // %2lc: two wide characters, using multibyte to wide conversion
24    int ret = std::sscanf(input, "%d%f%9s%2d%f*d %3[0-9]%2lc",
25                          &i, &x, str1, &j, &y, str2, warr);
26
```



Compiler messages:

```
main.cpp: In function 'int main()':
main.cpp:13:20: error: cannot initialize array of 'char' from a string literal with type array of 'char8_t'
13 |     char input[] = u8"25 54.32E-1 Thompson 56789 0123 56水";
    |                   ^~~~~~
```

master



eigen / Eigen / src / Core / util / **DisableStupidWarnings.h**

 **DisableStupidWarnings.h** 4.67 KB 



```
1  #ifndef EIGEN_WARNINGS_DISABLED
2  #define EIGEN_WARNINGS_DISABLED
3
4  #ifdef _MSC_VER
5      // 4100 - unreferenced formal parameter (occurred e.g. in aligned_allocator::destroy(pointer p))
6      // 4101 - unreferenced local variable
7      // 4181 - qualifier applied to reference type ignored
8      // 4211 - nonstandard extension used : redefined extern to static
9      // 4244 - 'argument' : conversion from 'type1' to 'type2', possible loss of data
10     // 4273 - QtAlignedMalloc, inconsistent DLL linkage
11     // 4324 - structure was padded due to declspec(align())
12     // 4503 - decorated name length exceeded, name was truncated
13     // 4512 - assignment operator could not be generated
14     // 4522 - 'class' : multiple assignment operators specified
15     // 4700 - uninitialized local variable 'xyz' used
16     // 4714 - function marked as __forceinline not inlined
17     // 4717 - 'function' : recursive on all control paths, function will cause runtime stack overflow
18     // 4800 - 'type' : forcing value to bool 'true' or 'false' (performance warning)
19     #ifndef EIGEN_PERMANENTLY_DISABLE_STUPID_WARNINGS
20         #pragma warning( push )
21     #endif
22     #pragma warning( disable : 4100 4101 4181 4211 4244 4273 4324 4503 4512 4522 4700 4714 4717 4800)
--
```

libeigen > eigen > Repository

master



eigen / Eigen / src / Core / util / DisableStupidWarnings.h

 **DisableStupidWarnings.h** 4.67 KB 

```
52
53 #elif defined __GNUC__
54
55 #if (!defined(EIGEN_PERMANENTLY_DISABLE_STUPID_WARNINGS)) && (__GNUC__ > 4 || (__GNUC__ == 4 && __GNUC_MINOR__ >= 6))
56     #pragma GCC diagnostic push
57 #endif
58 // g++ warns about local variables shadowing member functions, which is too strict
59 #pragma GCC diagnostic ignored "-Wshadow"
60 #if __GNUC__ == 4 && __GNUC_MINOR__ < 8
61     // Until g++-4.7 there are warnings when comparing unsigned int vs 0, even in templated functions:
62     #pragma GCC diagnostic ignored "-Wtype-limits"
63 #endif
64 #if __GNUC__ >= 6
65     #pragma GCC diagnostic ignored "-Wignored-attributes"
66 #endif
67 #if __GNUC__ == 7
68     // See: https://gcc.gnu.org/bugzilla/show_bug.cgi?id=89325
69     #pragma GCC diagnostic ignored "-Wattributes"
70 #endif
71 #endif
```

godbolt.org

Incognito

C++ source #1

```
1 #include <string_view>
2 #include <iostream>
3
4 int main()
5 {
6     std::string_view sv = nullptr;
7     std::cout << sv.size();
8     return 0;
9 }
```

#1 with x86-64 gcc 8.3

Wrap lines

Compiler returned: 0

x86-64 gcc 8.3 (Editor #1, Compiler #1) C++

x86-64 gcc 8.3 --std=c++17 -Wall

godbolt.org Incognito

C++ source #1 x #1 with x86-64 gcc 9.1 x

```
1 #include <string_view>
2 #include <iostream>
3
4 int main()
5 {
6     std::string_view sv = nullptr;
7     std::cout << sv.size();
8     return 0;
9 }
```

Wrap lines

```
<source>: In function 'int main()':
<source>:6:27: warning: null argument where non-
null required (argument 2) [-Wnonnull]
    6 |     std::string_view sv = nullptr;
      |                               ^~~~~~
Compiler returned: 0
```

x86-64 gcc 9.1 (Editor #1, Compiler #1) C++ x

x86-64 gcc 9.1 --std=c++17 -Wall

godbolt.org Incognito

C++ source #1 #1 with x86-64 gcc 8.3

```
1 #include <utility>
2
3 struct S {};
4
5 S getS()
6 {
7     S s;
8     return std::move(s);
9 }
```

Compiler returned: 0

x86-64 gcc 8.3 (Editor #1, Compiler #1) C++

x86-64 gcc 8.3 --std=c++17 -Wall

godbolt.org Incognito

C++ source #1 #1 with x86-64 gcc 9.1

```
1 #include <utility>
2
3 struct S {};
4
5 S getS()
6 {
7     S s;
8     return std::move(s);
9 }
```

<source>: In function 'S getS()':
<source>:8:21: warning: moving a local object in a return statement prevents copy elision [-Wpessimizing-move]
8 | return std::move(s);
 | ~~~~~^~
<source>:8:21: note: remove 'std::move' call
Compiler returned: 0

x86-64 gcc 9.1 (Editor #1, Compiler #1) C++
x86-64 gcc 9.1 --std=c++17 -Wall

godbolt.org

C++ source #1

Save/Load Add new... Vim CppInsights Quick-bench C++

```
1 #include <boost/mpl/size_t.hpp>
2
3 typedef boost::mpl::size_t < ~0 > param_type;
```

#1 with x86-64 gcc 8.3

Wrap lines

Compiler returned: 0

x86-64 gcc 8.3 (Editor #1, Compiler #1) C++

x86-64 gcc 8.3 --std=c++17 -Wall


```
C++ source #1 X  
A Save/Load + Add new... Vim CppInsights Quick-bench C++  
1 #include <boost/mpl/size_t.hpp>  
2  
3 typedef boost::mpl::size_t < ~0 > param_type;
```

```
#1 with x86-64 gcc 9.1 X  
A Wrap lines  
<source>:3:33: error: narrowing conversion of '-1' from 'int' to 'long unsigned int' [-  
Wnarrowing]  
3 | typedef boost::mpl::size_t < ~0 > param_type;  
  |                               ^  
Compiler returned: 1
```

C++ source #1

Save/Load Add new... Vim CppInsights Quick-bench

C++

```
1 #include <boost/regex.hpp>
2 #include <string>
3 #include <iostream>
4
5 int main()
6 {
7     std::string s = "Boost Libraries";
8     boost::regex expr{"\\w+\\s\\w+"};
9     std::cout << std::boolalpha << boost::regex_match(s, expr) << '\n';
10 }
```

#1 with x86-64 gcc 10.1

Wrap lines

Compiler returned: 0

x86-64 gcc 10.1 (Editor #1, Compiler #1) C++

x86-64 gcc 10.1



--std=c++17 -Wall

godbolt.org Incognito

C++ source #1

```
1 #include <boost/regex.hpp>
2 #include <string>
3 #include <iostream>
4
5 int main()
6 {
7     std::string s = "Boost Libraries";
8     boost::regex expr{"\\w+\\s\\w+"};
9     std::cout << std::boolalpha << boost::regex_match(s, expr) << '\n';
10 }
```

#1 with x86-64 gcc 10.1

Wrap lines

In file included from /celibs/boost_1_64_0/boost/regex/v4/regex.hpp:82,
from /celibs/boost_1_64_0/boost/regex.hpp:31,
from <source>:1:
/celibs/boost_1_64_0/boost/regex/v4/match_results.hpp: In instantiation of 'class boost::match_results<const char*>':
/celibs/boost_1_64_0/boost/regex/v4/instances.hpp:182:44: required from here
/celibs/boost_1_64_0/boost/regex/v4/match_results.hpp:60:77: error: no type named 'const_reference' in 'class std::allocator<boost::sub_match<const char*> >'
60 | typedef typename Allocator::const_reference const_reference;
|

x86-64 gcc 10.1 (Editor #1, Compiler #1) C++

x86-64 gcc 10.1 --std=c++2a -Wall

Рекомендации

- Версионирование сборочного окружения

Рекомендации

- Версионирование сборочного окружения
- Контроль над 3-rd party

Рекомендации

- Версионирование сборочного окружения
- Контроль над 3-rd party
- CI и хорошее покрытие тестами

Рекомендации

- Версионирование сборочного окружения
- Контроль над 3-rd party
- CI и хорошее покрытие тестами
- Динамические анализаторы
 - asan, ubsan, tsan, valgrind

Рекомендации

- Версионирование сборочного окружения
- Контроль над 3-rd party
- CI и хорошее покрытие тестами
- Динамические анализаторы
 - asan, ubsan, tsan, valgrind
- Статические анализаторы

Рекомендации


- Версионирование сборочного окружения
- Контроль над 3-rd party
- CI и хорошее покрытие тестами
- Динамические анализаторы
 - asan, ubsan, tsan, valgrind
- Статические анализаторы
- Не отключать диагностики

Рекомендации

- Версионирование сборочного окружения
- Контроль над 3-rd party
- CI и хорошее покрытие тестами
- Динамические анализаторы
 - asan, ubsan, tsan, valgrind
- Статические анализаторы
- Не отключать диагностики
- ISO/IEC 14882

Рекомендации

- Версионирование сборочного окружения
- Контроль над 3-rd party
- CI и хорошее покрытие тестами
- Динамические анализаторы
 - asan, ubsan, tsan, valgrind
- Статические анализаторы
- Не отключать диагностики
- ISO/IEC 14882



обновление компилятора =
жесткая рутинная задача

Спасибо за внимание!

Елена Степанова

 st_1ena

 elena.1.stepanova@nokia.com

Страшилки из продакшена:

Александр Иргер

 alexander.irger@nokia.com

