Equivalence in cross-compilation compiler warnings

Tom Isaacson

Navico

@parsley72

Cross-compilation?

"A cross compiler is a compiler capable of creating executable code for a platform other than the one on which the compiler is running".

Not that

"Equivalence in cross-platform compiler warnings"?

Basically:

- We have a large (legacy) codebase.

- Used to build many different products.

- Build for both Windows and Linux.

- How can we make sure that code that compiles for one platform also compiles for the other?

Compiler flags

- MSVC (Visual Studio) for Windows
  - /W4 – Level 4
  - /WX – Treat warnings as errors
  - Also:
    - /permissive- https://blogs.msdn.microsoft.com/vcblog/2016/11/16/permissive-switch/
- GCC for Linux
  - -Wall – All (for some previously defined value of all).
  - -Wextra – Extra
  - -Wconversion - Warn for implicit conversions that may alter a value.
  - -Werror – Treat warnings as errors.
  - Also:
    - -Wpedantic - Warnings demanded by strict ISO C++
    - https://kristerw.blogspot.co.nz/2017/09/useful-gcc-warning-options-not-enabled.html

Other MSVC flags – disable

Flag	Description
/wd4127	C4127 conditional expression is constant
/wd4221	C4221 nonstandard extension used : 'identifier' : cannot be initialized using address of automatic variable
/wd4251	C4251 'identifier' : class 'type' needs to have dll-interface to be used by clients of class 'type2'
/wd4265	C4265 'class' : class has virtual functions, but destructor is not virtual
/wd4351	C4351 new behavior: elements of array 'array' will be default initialized
/wd4373	C4373 '%$S': virtual function overrides '%$pS', previous versions of the compiler did not override when parameters only differed by const/volatile qualifiers
/wd4718	C4718 'function call' : recursive call has no side effects, deleting
/wd4996	C4996 The compiler encountered a deprecated declaration.

Other MSVC flags – enable

Flag	Description
/w14062	C4062 enumerator 'identifier' in switch of enum 'enumeration' is not handled
/w14191	C4191 'operator/operation' : unsafe conversion from 'type of expression' to 'type required'
/w14242	C4242 'identifier' : conversion from 'type1' to 'type2', possible loss of data
/w14244	C4244 'conversion' conversion from 'type1' to 'type2', possible loss of data
/w14254	C4254 'operator' : conversion from 'type1' to 'type2', possible loss of data
/w14296	C4296 'identifier' : 'const' automatic data initialized with compiler generated default constructor produces unreliable results
/w14302	C4302 'conversion' : truncation from 'type 1' to 'type 2'
/w14311	C4311 'variable' : pointer truncation from 'type' to 'type'
/w14355	C4355 'this' : used in base member initializer list
/w14388	C14388 signed/unsigned mismatch
/w14431	C4431 missing type specifier - int assumed. Note: C no longer supports default-int
/w14623	C4623 'derived class' : default constructor was implicitly defined as deleted because a base class default constructor is inaccessible or deleted
/w14263	C4263 'function' : member function does not override any base class virtual member function
/w14264	C4264 'virtual_function' : no override available for virtual member function from base 'class'; function is hidden
/w14266	C4266 'function' : no override available for virtual member function from base 'type'; function is hidden
/w14928	C4928 illegal copy-initialization; more than one user-defined conversion has been implicitly applied

C++ source #1 ✕

A▾   💾   ⬆   ▤

```cpp
1   #include <iostream>
2   #include <boost\Format.hpp>
3
4   int main(int /*argc*/, char** /*argv*/)
5   {
6       std::cout << boost::format("%|1$1| %|2$3|") % "Hello" % 3 << std::endl;
7   }
8
```

#1 with x86-64 MSVC 19 2017 RTW ✕

```
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:symbol_demangle Unknown operator: i
err:msvcrt:symbol_demangle Unknown operator: i
err:msvcrt:symbol_demangle Unknown operator: i
err:msvcrt:symbol_demangle Unknown operator: i
Compiler exited with result code 0
```

x86-64 MSVC 19 2017 RTW (Editor #1, Compiler #1) ✕

x86-64 MSVC 19 2017 RTW ▾

/EHsc /W4 /WX

A▾   11010   .LX0:   .text   //   \s+   Intel   Demangle   🗏   🖪   💾   💡   ⬆

```
1  $SG78848 DB      '/', 00H, 'o', 00H, 'p', 00H, 't', 00H, '/', 00H, 'c', 00H
```

⚠ Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64 - 10557ms

C++ source #1 ✕

A▾ 💾 ⬆ ▤

```cpp
1    #include <iostream>
2    #include <boost\Format.hpp>
3
4    int main(int /*argc*/, char** /*argv*/)
5    {
6        std::cout << boost::format("%|1$1| %|2$3|") % "Hello" % 3 << std::endl;
7    }
8
```

#1 with x86-64 gcc 7.2 ✕

```
2 : <source>:2:10: fatal error: boost\Format.hpp: No such file or directory
 #include <boost\Format.hpp>
          ^~~~~~~~~~~~~~~~~~
compilation terminated.
Compiler exited with result code 1
```

x86-64 gcc 7.2 (Editor #1, Compiler #1) ✕

x86-64 gcc 7.2 ▼   -Wall -Wextra -Werror

A▾ 11010 .LX0: .text // \s+ Intel Demangle ▯ ☰ ▢ ♦ ⬆

```
1  <Compilation failed>
```

⚠ g++ (GCC-Explorer-Build) 7.2.0 - cached

C++ source #1 ✕

A▾ 💾 ⬆ ▤

```cpp
1   class tRCoord
2   {
3   public:
4       tRCoord(double x, double y) : m_x(x), m_y(y) {}
5   private:
6       double m_x, m_y;
7   };
8
9   class tCoordinate
10  {
11  public:
12      enum eCoordinateType
13      {
14          eMercatorMeters,
15          eDecimalDegrees,
16          eRadians
17      };
18
19      explicit tCoordinate(const tRCoord& r, double altitude);
20  };
21
22  void UseCoord()
23  {
24      tRCoord rCoord(1.5, -1.5);
25      tCoordinate coord = tCoordinate(rCoord, tCoordinate::eRadians);
26  }
27
```

#1 with x86-64 MSVC 19 2017 RTW ✕

```
example.cpp
Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64
Copyright (C) Microsoft Corporation.  All rights reserved.
Compiler exited with result code 0
```

x86-64 MSVC 19 2017 RTW (Editor #1, Compiler #1) ✕

x86-64 MSVC 19 2017 RTW ▾   /EHsc /W4 /WX

A▾   11010   .LX0:   .text   //   \s+   Intel   Demangle   ▣ ↪ 💾 ◆ 📌

```
1  ; Listing generated by Microsoft (R) Optimizing Compiler Ver
2  INCLUDELIB LIBCMT
```

⚠ Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64 - 519ms

C++ source #1 ✕

x86-64 MSVC 19 2017 RTW (Editor #1, Compiler #1) ✕

x86-64 MSVC 19 2017 RTW

/EHsc /W4 /WX

A▾ 11010 .LX0: .text // \s+ Intel Demangle

```cpp
 4        tRCoord(double x, double y) : m_x(x), m_y(y) {}
 5    private:
 6        double m_x, m_y;
 7    };
 8
 9    class tCoordinate
10    {
11    public:
12        enum class eCoordinateType
13        {
14            eMercatorMeters,
15            eDecimalDegrees,
16            eRadians
17        };
18
19        explicit tCoordinate(const tRCoord& r, double altitude);
20    };
21
22    void UseCoord()
23    {
24        tRCoord rCoord(1.5, -1.5);
25        tCoordinate coord = tCoordinate(rCoord, tCoordinate::eCoordinateType::eRadiar
26    }
27
```

```
1  <Compilation failed>
```

⚠ Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for

x64- 984ms

#1 with x86-64 MSVC 19 2017 RTW ✕

```
example.cpp
25 : <source>(25): error C2440: '<function-style-cast>': cannot convert from 'initializer list' to 'tCoordinate'
25 : <source>(25): note: No constructor could take the source type, or constructor overload resolution was ambiguous
25 : <source>(25): error C2512: 'tCoordinate': no appropriate default constructor available
10 : <source>(10): note: see declaration of 'tCoordinate'
Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64
Copyright (C) Microsoft Corporation.  All rights reserved.
Compiler exited with result code 2
```

C++ source #1 ✕   ⬜ ✕

A▾ 💾 ⬆ ▤

```cpp
 1   class tRCoord
 2   {
 3   public:
 4       tRCoord(double x, double y) : m_x(x), m_y(y) {}
 5   private:
 6       double m_x, m_y;
 7   };
 8
 9   class tCoordinate
10   {
11   public:
12       enum eCoordinateType
13       {
14           eMercatorMeters,
15           eDecimalDegrees,
16           eRadians
17       };
18
19       explicit tCoordinate(const tRCoord& r, double altitude);
20   };
21
22   void UseCoord()
23   {
24       tRCoord rCoord(1.5, -1.5);
25       tCoordinate coord = tCoordinate(rCoord, tCoordinate::eRadians
26   }
27
```

#1 with x86-64 gcc 7.2 ✕   ⬜ ✕

```
<source>: In function 'void UseCoord()':
25 : <source>:25:17: error: variable 'coord' set but not used [-Werror=unus
        tCoordinate coord = tCoordinate(rCoord, tCoordinate::eRadians);
                    ^~~~~
cc1plus: all warnings being treated as errors
Compiler exited with result code 1
```

x86-64 gcc 7.2 (Editor #1, Compiler #1) ✕   ⬜ ✕

x86-64 gcc 7.2   ▼   -Wall -Wextra -Wconversion -Werror

A▾   11010   .LX0:   .text   //   \s+   Intel   Demangle   ▣ ⇱ 💾 🔧 ⬆

```
1  <Compilation failed>
```

⚠ g++ (GCC-Explorer-Build) 7.2.0 - cached

C++ source #1 ✕

```cpp
17        };
18
19          explicit tCoordinate(const tRCoord& r, double altitude);
20        };
21
22        void UseCoord()
23        {
24            tRCoord rCoord(1.5, -1.5);
25            tCoordinate coord = tCoordinate(rCoord, tCoordinate::eCoordinateType::eRadians);
26        }
```

x86-64 gcc 7.2 (Editor #1, Compiler #1) ✕

x86-64 gcc 7.2   ▾

-Wall -Wextra -Wconversion -Werror

A▾   11010   .LX0:   .text   //   \s+   Intel   Demangle   ▣

1 <Compilation failed>

⚠ g++ (GCC-Explorer-Build) 7.2.0 -cached

#1 with x86-64 gcc 7.2 ✕

```
<source>: In function 'void UseCoord()':
25 : <source>:25:83: error: no matching function for call to 'tCoordinate::tCoordinate(tRCoord&, tCoordinate::eCoordinateType)'
     tCoordinate coord = tCoordinate(rCoord, tCoordinate::eCoordinateType::eRadians);
                                                                                   ^
19 : <source>:19:14: note: candidate: tCoordinate::tCoordinate(const tRCoord&, double)
     explicit tCoordinate(const tRCoord& r, double altitude);
              ^~~~~~~~~~~
19 : <source>:19:14: note:   no known conversion for argument 2 from 'tCoordinate::eCoordinateType' to 'double'
9 : <source>:9:7: note: candidate: constexpr tCoordinate::tCoordinate(const tCoordinate&)
 class tCoordinate
       ^~~~~~~~~~~
9 : <source>:9:7: note:   candidate expects 1 argument, 2 provided
9 : <source>:9:7: note: candidate: constexpr tCoordinate::tCoordinate(tCoordinate&&)
9 : <source>:9:7: note:   candidate expects 1 argument, 2 provided
25 : <source>:25:17: error: unused variable 'coord' [-Werror=unused-variable]
     tCoordinate coord = tCoordinate(rCoord, tCoordinate::eCoordinateType::eRadians);
                 ^~~~~
cc1plus: all warnings being treated as errors
Compiler exited with result code 1
```

C++ source #1 ✕   ☐ ✕

A▾ 💾 ⬆ ▤

```cpp
 1   #include <iostream>
 2
 3   enum class eState
 4   {
 5       State1,
 6       State2,
 7       State3
 8   };
 9
10   void PrintState(eState state)
11   {
12       switch (state)
13       {
14           case eState::State1: std::cout << "State1"; break;
15           case eState::State2: std::cout << "State2"; break;
16       }
17   }
18
```

#1 with x86-64 gcc 7.2 ✕   ☐ ✕

```
<source>: In function 'void PrintState(eState)':
12 : <source>:12:12: error: enumeration value 'State3' not handled in switch [-Werr
         switch (state)
               ^
cc1plus: all warnings being treated as errors
Compiler exited with result code 1
```

x86-64 gcc 7.2 (Editor #1, Compiler #1) ✕   ☐ ✕

| x86-64 gcc 7.2 ▼ | | -Wall -Werror |

A▾ 11010 .LX0: .text // \s+ Intel Demangle 🖿 ☰ ☐ ♦ ⬆

```
1  <Compilation failed>
```

⚠ g++ (GCC-Explorer-Build) 7.2.0- cached

C++ source #1 ✕

```cpp
1   #include <iostream>
2
3   enum class eState
4   {
5       State1,
6       State2,
7       State3
8   };
9
10  void PrintState(eState state)
11  {
12      switch (state)
13      {
14          case eState::State1: std::cout << "State1"; break;
15          case eState::State2: std::cout << "State2"; break;
16          case eState::State3: std::cout << "State3"; break;
17      }
18  }
19
```

x86-64 gcc 7.2 (Editor #1, Compiler #1) ✕

x86-64 gcc 7.2

-Wall -Wextra -Wconversion -Werror

11010   .LX0:   .text   //   \s+   Intel   Demangle

```asm
1   .LC0:
2       .string "State1"
3   .LC1:
4       .string "State2"
5   .LC2:
6       .string "State3"
7   PrintState(eState):
8       push rbp
9       mov rbp, rsp
10      sub rsp, 16
11      mov DWORD PTR [rbp-4], edi
12      mov eax, DWORD PTR [rbp-4]
13      cmp eax, 1
14      je .L3
15      cmp eax, 2
16      je .L4
17      test eax, eax
18      je .L5
19      jmp .L6
20  .L5:
21      mov esi, OFFSET FLAT: LC0
```

⚠ g++ (GCC-Explorer-Build) 7.2.0-*cached*

#1 with x86-64 gcc 7.2 ✕

Compiler exited with result code 0

C++ source #1 ✕   ☐ ✕

A ▾   💾   ⬆   ▤

```cpp
1   #include <iostream>
2
3   enum class eState
4   {
5       State1,
6       State2,
7       State3
8   };
9
10  void PrintState(eState state)
11  {
12      switch (state)
13      {
14          case eState::State1: std::cout << "State1"; break;
15          case eState::State2: std::cout << "State2"; break;
16          default: std::cout << "Not handled"; break;
17      }
18  }
19
```

x86-64 gcc 7.2 (Editor #1, Compiler #1) ✕   ☐ ✕

x86-64 gcc 7.2 ▾   -Wall -Wextra -Wconversion -Werror

A ▾   11010   .LX0:   .text   //   \s+   Intel   Demangle   ▣ ☑ ▭ ⬦ ⬥

```asm
1   .LC0:
2     .string "State1"
3   .LC1:
4     .string "State2"
5   .LC2:
6     .string "Not handled"
7   PrintState(eState):
8     push rbp
9     mov rbp, rsp
10    sub rsp, 16
11    mov DWORD PTR [rbp-4], edi
12    mov eax, DWORD PTR [rbp-4]
13    test eax, eax
14    je .L3
15    cmp eax, 1
16    je .L4
17    jmp .L6
18  .L3:
19    mov esi, OFFSET FLAT:.LC0
20    mov edi, OFFSET FLAT:std::cout
21    call std::basic_ostream<char, std::char_traits<char> >& std::ope
```

⚠ g++ (GCC-Explorer-Build) 7.2.0- *cached*

#1 with x86-64 gcc 7.2 ✕   ☐ ✕

Compiler exited with result code 0

C++ source #1 ✕   □ ✕   #1 with x86-64 MSVC 19 2017 RTW ✕   □ ✕

```cpp
 1    #include <iostream>
 2
 3    enum class eState
 4    {
 5        State1,
 6        State2,
 7        State3
 8    };
 9
10    void PrintState(eState state)
11    {
12        switch (state)
13        {
14            case eState::State1: std::cout << "State1"; break;
15            case eState::State2: std::cout << "State2"; break;
16        }
17    }
18
```

```
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
Compiler exited with result code 0
```

x86-64 MSVC 19 2017 RTW (Editor #1, Compiler #1) ✕

x86-64 MSVC 19 2017 RTW ▾   /EHsc /WX /W4

A▾   11010   .LX0:   .text   //   \s+   Intel   Demangle

```
 1    $SG25832 DB    'State1', 00H
 2    $SG25834 DB    'State2', 00H
```

⚠ Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64 -3184ms

C++ source #1 ✕   ⊡ ✕

A▾   💾   ⬆   ▤

```cpp
1   #include <iostream>
2
3   enum class eState
4   {
5       State1,
6       State2,
7       State3
8   };
9
10  void PrintState(eState state)
11  {
12      switch (state)
13      {
14          case eState::State1: std::cout << "State1"; break;
15          case eState::State2: std::cout << "State2"; break;
16      }
17  }
18
```

#1 with x86-64 MSVC 19 2017 RTW ✕   ⊡ ✕

```
example.cpp
16 : <source>(16): error C2220: warning treated as error - no 'object' file generat
16 : <source>(16): warning C4062: enumerator 'eState::State3' in switch of enum 'eS
4 : <source>(4): note: see declaration of 'eState'
Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64
Copyright (C) Microsoft Corporation.  All rights reserved.
Compiler exited with result code 2
```

x86-64 MSVC 19 2017 RTW (Editor #1, Compiler #1) ✕   ⊡ ✕

x86-64 MSVC 19 2017 RTW ▾   /EHsc /W4 /w14062 /WX

A▾   11010   .LX0:   .text   //   \s+   Intel   Demangle   ▣ ⬀ ▢ ◈ ⬆

```
1
```

⚠ Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64 - *cached*

C++ source #1

```cpp
1   #include <iostream>
2
3   enum class eState
4   {
5       State1,
6       State2,
7       State3
8   };
9
10  void PrintState(eState state)
```

#1 with x86-64 MSVC 19 2017 RTW

```
example.cpp
/opt/compiler-explorer/windows/10.0.10240.0/ucrt/malloc.h(44): error C2220: warning treated as error - no 'object' file generated
/opt/compiler-explorer/windows/10.0.10240.0/ucrt/malloc.h(44): warning C4820: '_heapinfo': '4' bytes padding added after data member '_heapinfo::_useflag'
/opt/compiler-explorer/windows/19.10.25017/lib/native/include/eh.h(31): warning C4987: nonstandard extension used: 'throw (...)'
/opt/compiler-explorer/windows/19.10.25017/lib/native/include/vcruntime_exception.h(25): warning C4820: '__std_exception_data': '7' bytes padding added after data memb
/opt/compiler-explorer/windows/19.10.25017/lib/native/include/iosfwd(356): warning C4365: 'return': conversion from 'int' to 'std::char_traits<wchar_t>::int_type', sig
/opt/compiler-explorer/windows/19.10.25017/lib/native/include/iosfwd(452): warning C4365: 'return': conversion from 'int' to 'std::char_traits<unsigned short>::int_typ
/opt/compiler-explorer/windows/19.10.25017/lib/native/include/vcruntime_typeinfo.h(41): warning C4820: '__std_type_info_data': '7' bytes padding added after data membe
/opt/compiler-explorer/windows/19.10.25017/lib/native/include/vcruntime_typeinfo.h(41): warning C4623: '__std_type_info_data': default constructor was implicitly defin
/opt/compiler-explorer/windows/19.10.25017/lib/native/include/vcruntime_typeinfo.h(41): warning C4626: '__std_type_info_data': assignment operator was implicitly defin
/opt/compiler-explorer/windows/19.10.25017/lib/native/include/vcruntime_typeinfo.h(41): warning C5027: '__std_type_info_data': move assignment operator was implicitly
/opt/compiler-explorer/windows/19.10.25017/lib/native/include/xlocinfo.h(59): warning C4820: '_Collvec': '4' bytes padding added after data member '_Collvec::_Page'
/opt/compiler-explorer/windows/19.10.25017/lib/native/include/xlocinfo.h(65): warning C4820: '_Ctypevec': '4' bytes padding added after data member '_Ctypevec::_Page'
```

x86-64 MSVC 19 2017 RTW (Editor #1, Compiler #1)

x86-64 MSVC 19 2017 RTW

`/EHsc /WX /Wall`

11010 | .LX0: | .text | // | \s+ | Intel | Demangle

```
1
```

⚠ Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64 - *cached*

C++ source #1 ✕   #1 with x86-64 MSVC 19 2017 RTW ✕

```cpp
 1    #include <iostream>
 2
 3    enum class eState
 4    {
 5        State1,
 6        State2,
 7        State3
 8    };
 9
10    void PrintState(eState state)
11    {
12        switch (state)
13        {
14            case eState::State1: std::cout << "State1"; break;
15            case eState::State2: std::cout << "State2"; break;
16            case eState::State3: std::cout << "State3"; break;
17        }
18    }
19
```

```
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
```

x86-64 MSVC 19 2017 RTW (Editor #1, Compiler #1) ✕

x86-64 MSVC 19 2017
RTW ▾   /EHsc /W4 /w14062 /WX

A▾   11010   .LX0:   .text   //   \s+   Intel   Demangle

C++ source #1 ✕   #1 with x86-64 MSVC 19 2017 RTW ✕

```cpp
1   #include <iostream>
2
3   enum class eState
4   {
5       State1,
6       State2,
7       State3
8   };
9
10  void PrintState(eState state)
11  {
12      switch (state)
13      {
14          case eState::State1: std::cout << "State1"; break;
15          case eState::State2: std::cout << "State2"; break;
16          default: std::cout << "Not handled"; break;
17      }
18  }
19
```

```
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
err:msvcrt:demangle_datatype Unknown type @
Compiler exited with result code 0
```

x86-64 MSVC 19 2017 RTW (Editor #1, Compiler #1) ✕

x86-64 MSVC 19 2017 RTW ▾   /EHsc /W4 /w14062 /WX

A▾   11010   .LX0:   .text   //   \s+   Intel   Demangle

```
1   $SG25841 DB    'State1', 00H
```

⚠ Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64 - 3222ms

GCC: -Wconversion

Warn for implicit conversions that may alter a value. This includes conversions between real and integer, like abs (x) when x is double; conversions between signed and unsigned, like unsigned ui = -1; and conversions to smaller types, like sqrtf (M_PI). Do not warn for explicit casts like abs ((int) x) and ui = (unsigned) -1, or if the value is not changed by the conversion like in abs (2.0). Warnings about conversions between signed and unsigned integers can be disabled by using -Wno-sign-conversion.

For C++, also warn for confusing overload resolution for user-defined conversions; and conversions that never use a type conversion operator: conversions to void, the same type, a base class or a reference to them. Warnings about conversions between signed and unsigned integers are disabled by default in C++ unless -Wsign-conversion is explicitly enabled.

https://gcc.gnu.org/onlinedocs/gcc/Warning-Options.html

C++ source #1 ✕

A▾ 💾 ⬆ ▤

```cpp
1   int RVG(int inputAmplitude, float range, float rangeLimit)
2   {
3       int retVal(0);
4
5       retVal = static_cast<int>((range * inputAmplitude) / rangeLimit);
6
7       return retVal;
8   }
9
```

x86-64 gcc 7.2 (Editor #1, Compiler #1) ✕

x86-64 gcc 7.2   ▼   -Wall -Wextra -Wconversion -Werror

A▾ 11010 .LX0: .text // \s+ Intel Demangle

1 <Compilation failed>

⚠ g++ (GCC-Explorer-Build) 7.2.0-*cached*

#1 with x86-64 gcc 7.2 ✕

```
<source>: In function 'int RVG(int, float, float)':
5 : <source>:5:40: error: conversion to 'float' from 'int' may alter its value [-Werror=conversion]
     retVal = static_cast<int>((range * inputAmplitude) / rangeLimit);
                                        ^~~~~~~~~~~~~~

cc1plus: all warnings being treated as errors
Compiler exited with result code 1
```

C++ source #1 ✕

A▾ 💾 ⬆ ▤

```cpp
1   int RVG(int inputAmplitude, float range, float rangeLimit)
2   {
3       int retVal(0);
4
5       retVal = static_cast<int>((range * static_cast<float>(inputAmplitude)) / rangeLimit);
6
7       return retVal;
8   }
9
```

#1 with x86-64 gcc 7.2 ✕

Compiler exited with result code 0

x86-64 gcc 7.2 (Editor #1, Compiler #1) ✕

x86-64 gcc 7.2 ▾

-Wall -Wextra -Wconversion -Werror

A▾ 11010 .LX0: .text // \s+ Intel Demangle 🖿 ⎘ ⬚ 💡 ⬆

```asm
1  RVG(int, float, float):
2    push rbp
3    mov rbp, rsp
4    mov DWORD PTR [rbp-20], edi
```

⚠ g++ (GCC-Explorer-Build) 7.2.0- cached

C++ Core Guidelines

ES.46: Avoid lossy (narrowing, truncating) arithmetic conversions

Reason

A narrowing conversion destroys information, often unexpectedly so.

Note

The guideline support library offers a narrow_cast operation for specifying that narrowing is acceptable and a narrow ("narrow if") that throws an exception if a narrowing would throw away information:

Enforcement

A good analyzer can detect all narrowing conversions. However, flagging all narrowing conversions will lead to a lot of false positives. Suggestions:

- flag all floating-point to integer conversions (maybe only float->char and double->int. Here be dragons! we need data)
- flag all long->char (I suspect int->char is very common. Here be dragons! we need data)
- consider narrowing conversions for function arguments especially suspect

Guideline Support Library (GSL)

The Guideline Support Library (GSL) contains functions and types that are suggested for use by the C++ Core Guidelines maintained by the Standard C++ Foundation. The library includes types like owner<>, not_null<>, span<>, string_span and others.

https://github.com/Microsoft/GSL

The entire implementation is provided inline in the headers under the gsl directory. The implementation generally assumes a platform that implements C++14 support. There are specific workarounds to support MSVC 2015.

https://github.com/martinmoene/gsl-lite

Single-file header-only variant of Microsoft's implementation adapted for C++98, C++03. It should also work when compiled as C++11, C++14.

C++ source #1 ✕

```cpp
1    #include <gsl/gsl>
2
3    int RVG(int inputAmplitude, float range, float rangeLimit)
4    {
5        int retVal(0);
6
7        retVal = static_cast<int>((range * gsl::narrow_cast<float>(inputAmplitude)) / rangeLimit);
8
9        return retVal;
10   }
11
```

#1 with x86-64 gcc 7.2 ✕

Compiler exited with result code 0

x86-64 gcc 7.2 (Editor #1, Compiler #1) ✕

x86-64 gcc 7.2 ▾   -Wall -Wextra -Wconversion -Werror

A▾   11010   .LX0:   .text   //   \s+   Intel   Demangle

```
1   RVG(int, float, float):
2     push rbp
3     mov  rbp, rsp
4     sub  rsp, 32
```

⚠ g++ (GCC-Explorer-Build) 7.2.0- *cached*

Compiler exited with result code 0

```cpp
#include <iostream>
#include <gsl/gsl>

int RVG(int inputAmplitude, float range, float rangeLimit)
{
    int retVal(0);

    try
    {
        retVal = static_cast<int>((range * gsl::narrow<float>(inputAmplitude)) / rangeLimit);
    }
    catch (const gsl::narrowing_error& e)
    {
        std::cout << e.what();
    }

    return retVal;
}
```

x86-64 gcc 7.2 ▾

-Wall -Wextra -Wconversion -Werror

A▾ 11010 .LX0: .text // \s+ Intel Demangle
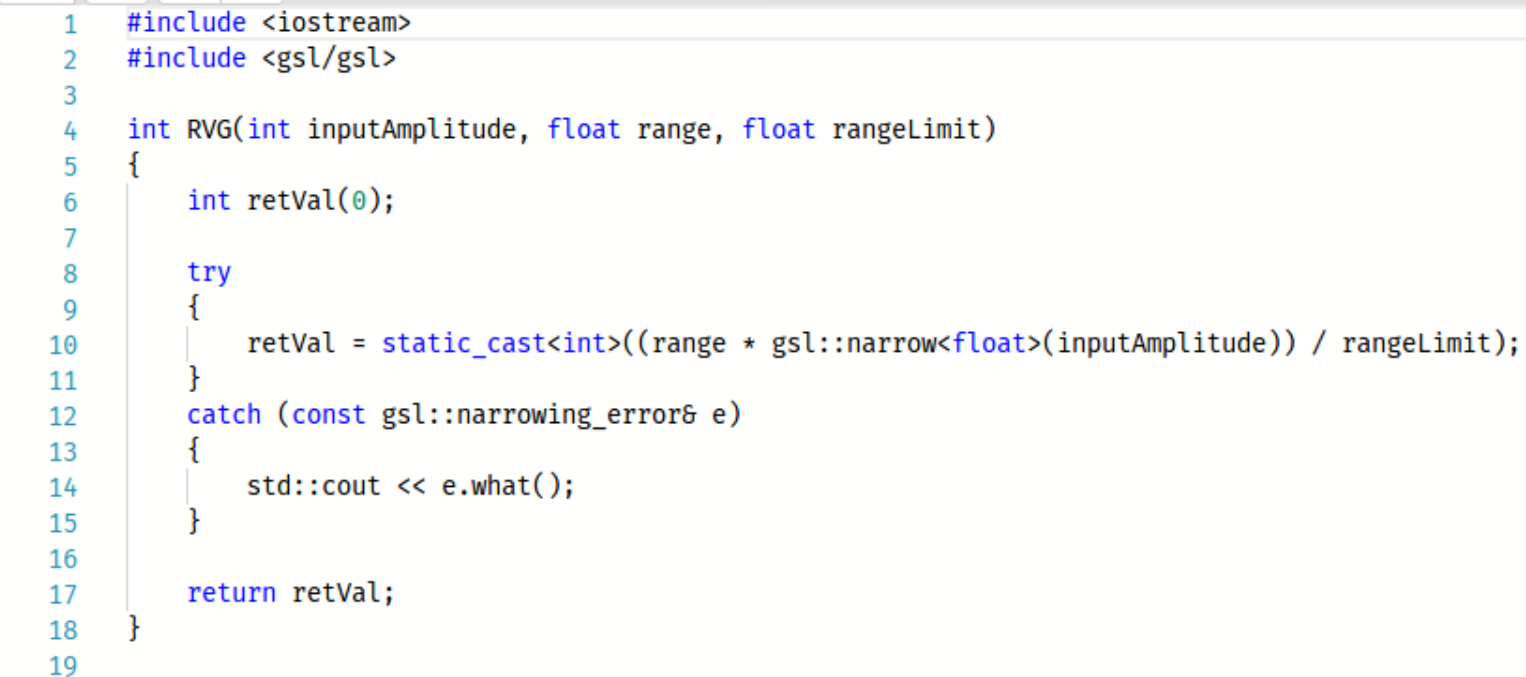
```asm
1  std::exception::exception():
2    push rbp
```

⚠ g++ (GCC-Explorer-Build) 7.2.0 - cached

C++ source #1 ✕

```cpp
1    int RVG(int inputAmplitude, float range, float rangeLimit)
2    {
3        int retVal(0);
4
5        retVal = static_cast<int>((range * inputAmplitude) / rangeLimit);
6
7        return retVal;
8    }
9
```

#1 with x86-64 MSVC 19 2017 RTW ✕

```
example.cpp
Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64
Copyright (C) Microsoft Corporation.  All rights reserved.
Compiler exited with result code 0
```

x86-64 MSVC 19 2017 RTW (Editor #1, Compiler #1) ✕

x86-64 MSVC 19 2017 RTW ▾

/EHsc /WX /Wall

11010   .LX0:   .text   //   \s+   Intel   Demangle

```
1    retVal$ = 0
2    inputAmplitude$ = 32
```

⚠ Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64- *cached*

C++ source #1 ✕

```cpp
1   int aa;
2   unsigned short bb;
3   int main()
4   {
5       int b = 0, c = 0;
6       short a = b + c;    // C4244
7
8       bb += c;    // C4244
9       bb = bb + c;    // C4244
10      bb += (unsigned short)aa;    // C4244
11      bb = bb + (unsigned short)aa;    // OK
12  }
13
```

#1 with x86-64 MSVC 19 2017 RTW ✕

```
example.cpp
6 : <source>(6): error C2220: warning treated as error - no 'object' file generated
6 : <source>(6): warning C4244: 'initializing': conversion from 'int' to 'short', possible loss of data
8 : <source>(8): warning C4244: '+=': conversion from 'int' to 'unsigned short', possible loss of data
9 : <source>(9): warning C4244: '=': conversion from 'int' to 'unsigned short', possible loss of data
6 : <source>(6): warning C4189: 'a': local variable is initialized but not referenced
Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64
Copyright (C) Microsoft Corporation.  All rights reserved.
Compiler exited with result code 2
```

x86-64 MSVC 19 2017 RTW (Editor #1, Compiler #1) ✕

x86-64 MSVC 19 2017 RTW

/EHsc /W4 /WX

11010   .LX0:   .text   //   \s+   Intel   Demangle

```
1
```

⚠ Microsoft (R) C/C++ Optimizing Compiler Version 19.10.25017 for x64 - *cached*

C++ Standard

https://isocpp.org/std/the-standard

Purchase the C++14 official standard for US$133 or working drafts are free.

Only 4 mentions of warnings?

Key phrase is "diagnostic required" but this is only mentioned 17 times.

"The only thing it requires from compilers is to diagnose when the program is ill-formed."

Future improvements?

CppCast #111 with Patrice Roy

http://cppcast.com/2017/07/patrice-roy/

Been a participating member in the ISO C++ Standards Committee since late 2014

30:40:

"People have even talked about standardising some warning messages. It's something that's floating around, there's nothing official there, we're just talking right now."

C++ Core Guidelines

ES.46: Avoid lossy (narrowing, truncating) arithmetic conversions

Reason

A narrowing conversion destroys information, often unexpectedly so.

Note

The guideline support library offers a narrow_cast operation for specifying that narrowing is acceptable and a narrow ("narrow if") that throws an exception if a narrowing would throw away information:

Enforcement

A good analyzer can detect all narrowing conversions. However, flagging all narrowing conversions will lead to a lot of false positives. Suggestions:

- flag all floating-point to integer conversions (maybe only float->char and double->int. Here be dragons! we need data)
- flag all long->char (I suspect int->char is very common. Here be dragons! we need data)
- consider narrowing conversions for function arguments especially suspect

Summary

- Unfortunately true platform compiler equivalence is not yet possible.
- Compilers offer different warnings.
- Differences exist between compiler warning implementations.