Document number: P0191R1
Date: 2016-03-20
Audience: Evolution Working Group
Reply-to: Daniel Markus <daniel@markus.email>

C++ virtual member function pointer comparison

I. Introduction

At times there is a need to compare two virtual member function pointers for equality. It would be a good thing to allow in the C++ standard since it would allow storing and searching for virtual member functions.

II. Motivation and Scope

Many may think that the C++ standard document is only interesting for compiler implementers. The truth is that it is almost as important to library writers so they are able to verify their library's portability. Code that compiles on one compiler may not compile on another due to the compilers' varying conformity to the standard. For a library feature that works on certain compilers but not others, it is more likely it will be released if it has support in the standard.

In many library components like Delegates [1], Observers [2], and Mocks [3] there is a need to use virtual member function pointers as part of search criteria. To be able to search for such pointer, at least the equality comparison must be supported. Today, comparing equality of two virtual member function pointers compiles and runs on many popular compilers like GCC, Visual C++, and Clang, but it is unspecified by the standard. Looking at N4527, §5.10/3 [expr.eq] we read:

"– If either is a pointer to a virtual member function, the result is unspecified."

This paper's proposal is to specify equality comparison for two virtual member function pointers but keep it unspecified if only one of the pointers is a virtual member function pointer.

III. Impact on the Standard

There is no impact on the standard as the specification of comparing pointers to virtual member functions would be a pure extension requiring no changes to the standard.

IV. Design Decisions

The comparison between virtual member function pointers could be extended to also specify less-than and greater-than to be able to use the pointers as parts of keys in ordered containers, but the equality comparison is sufficient to begin with as a first step. Compiler implementers should be contacted and if it turns out that less-than and greater-than are also feasible to implement, then of course those comparisons would be welcome too.

V. Technical Specifications

Proposed change in §5.10/3 [expr.eq]:

Comparing pointers to members is defined as follows:

- If two pointers to members are both the null member pointer value, they compare equal.
- If only one of two pointers to members is the null member pointer value, they compare unequal.
- If ~~either~~ only one of two pointers to members is a pointer to a virtual member function, the result is unspecified.
- If one refers to a member of class C1 and the other refers to a member of a different class C2, where neither is a base class of the other, the result is unspecified.
- If both refer to (possibly different) members of the same union (9.5), they compare equal.
- Otherwise, two pointers to members compare equal if they would refer to the same member of the same most derived object (1.8) or the same subobject if indirection with a hypothetical object of the associated class type were performed, otherwise they compare unequal.

VI. References

[1] Don Clugston, *Member Function Pointers and the Fastest Possible C++ Delegates*, April 2005, Available online at http://www.codeproject.com/Articles/7150/Member-Function-Pointers-and-the-Fastest-Possible

[2] Herb Sutter, *Generalizing Observer*, September 2003, Available online at http://www.drdobbs.com/cpp/generalizing-observer/184403873

[3] Daniel Markus, *Unimock*, March 2015, Available online at https://github.com/unimock-cpp/unimock