A proposal for a type trait to detect scoped enumerations

Document Number: P1048R0

Date: 2018-05-01

Reply-to: Juan Alday (alday@ieee.org)

Audience: Library Evolution Working Group

Introduction

This paper proposes is_scoped_enum, a new trait for the C++ Standard Library, to detect whether a type is a scoped enumeration.

Motivation and Scope

It is useful in certain contexts to know whether an enumeration is scoped or unscoped and apply (via SFINAE) different behavior depending on the type of such enumeration.

One use the author has recently worked on involves creating a set of unit tests to track the progress of a legacy library migration to modern C++. By using this trait, it is possible to define a unit test to track the progress of migration of unscoped to scoped enumerations.

Impact On The Standard

This proposal is a pure library extension.

It proposes changes to an existing header, <type_traits>, but it does not require changes to any standard classes or functions and it does not require changes to any of the standard requirement tables.

This proposal does not require any changes in the core language, and it has been implemented in standard C++.

This proposal does not depend on any other library extensions.

Naming

The existing trait to detect an enum type is **is_enum**, so variations containing ‘enumeration’ are not considered by the author.

is_scoped_enum is the suggested name, although there are many implementations of this trait using is_enum_class.

Wording

All proposed additions (there are no deletions) are relative to the post-Jacksonville working draft N4741. Editorial notes are displayed against a gray background

Insert into [meta.type.synop] (23.15.2) as shown:

```
template<class T>struct is_member_pointer;

template<class T>struct is_scoped_enum;


template<class T>

inline constexpr bool is_member_pointer_v = is_member_pointer<T>::value;

template<class T>

inline constexpr bool is_scoped_enum_v = is_scoped_enum<T>::value;
```

Insert into table 41

`template<class T>struct is_member_pointer;`	`T is a pointer-to-member type (6.7.2)`	
`template<class T>struct is_scoped_enum;`	`T is a scoped enumeration [dcl.enum]`	

Example implementation

```
template<class _T, bool = is_enum_v<_T>> struct __is_scoped_enum_helper : false_type {};
template<class _T>struct __is_scoped_enum_helper<_T, true>
  : public bool_constant<!is_convertible_v<_T, underlying_type_t<_T>>> {};
template<class _T>struct is_scoped_enum : public __is_scoped_enum_helper<_T> {};
```

Acknowledgments

Thanks to Jonathan Wakely for his comments on early versions of this draft

Bibliography

[N4741] Richard Smith: “Working Draft, Standard for Programming Language C++.” ISO/IEC JTC1/SC22/WG21 document N4618 (post-Jacksonville mailing), 2018–04–02. http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2018/n4741.pdf

7 Document history

Version	Date	Changes
0	2018-05-01	Initial draft