The case of the constructor that was being ignored

devblogs.microsoft.com/oldnewthing/20220704-00

July 4, 2022

Raymond Chen

When I pointed out a way to build URL query strings in the Windows Runtime, the customer reported that it didn't work.

```
#include <winrt/Windows.Web.Http.h>

void test()
{
    auto encoder = HttpFormUrlEncodedContent({
        { L"v", L"dQw4w9WgXcQ" },
        { L"t", L"43s" },
    });
}
```

This failed with the error

```
error C2440: '<function-style-cast>': cannot convert from 'initializer list' to
'winrt::Windows::Web::Http::HttpFormUrlEncodedContent'
message : No constructor could take the source type, or constructor overload resolution was
ambiguous
```

Let's start debugging.

If you do a "Go to definition" on the `HttpFormUrlEncodedContent` in Visual Studio, you are taken to the class definition, which happens to be in the header file `impl/Windows.Web.Http.2.h`.

```
    struct __declspec(empty_bases) HttpFormUrlEncodedContent :
        Windows::Web::Http::IHttpContent,
        impl::require<HttpFormUrlEncodedContent, Windows::Foundation::IStringable>
    {
        HttpFormUrlEncodedContent(std::nullptr_t) noexcept {}
        HttpFormUrlEncodedContent(void* ptr, take_ownership_from_abi_t) noexcept :
            Windows::Web::Http::IHttpContent(ptr, take_ownership_from_abi) {}
        explicit HttpFormUrlEncodedContent(
            param::iterable<Windows::Foundation::Collections::
                IKeyValuePair<hstring, hstring>> const& content);
    };
```

Let's look at these constructors one at a time.

First up is the `nullptr` constructor for creating an empty `HttpFormUrlEncodedContent` smart pointer.

Next is the `take_ownership_from_abi` constructor for creating a `HttpFormUrlEncoded-Content` that takes over ownership of the object from a pointer obtained at the ABI layer. It is a two-parameter constructor and therefore would never be considered since we are calling the constructor with one parameter.

Last is the one we are trying to call: It takes a `param::iterable` of `IKeyValuePair<hstring, hstring>`.

There are also two implicitly defined constructors: The copy and move constructor. Those candidates look like this:

```
HttpFormUrlEncodedContent(HttpFormUrlEncodedContent const&) = default;
HttpFormUrlEncodedContent(HttpFormUrlEncodedContent &&) = default;
```

Okay, so we have four candidates that survived the arity check.

- `nullptr` constructor.
- `param::iterable` constructor.
- copy constructor.
- move constructor.

The error messages says that the compiler could not find a suitable constructor, so we have to think about why the `param::iterable` constructor wasn't chosen. We expect it to be chosen bcause `param::iterable` has a conversion constructor that takes an `initializer_list`. Why isn't that conversion being used?

I could not reproduce the error in my test project, so I asked the customer to send me theirs. I ran the file through the preprocessor so I could see exactly what the compiler saw, thinking that maybe the customer had some `#ifdef` or other weird configuration.

I searched the preprocessed file for the `param::iterable` constructor.

And it wasn't there!

The preprocessed file had a forward declaration for `param::iterable`, but no definition. That explains why the compiler couldn't convert the `initializer_list` to a `param::iterable`: Because the conversion constructor hadn't yet been declared!

The `param::iterable` template class is defined in the header file `winrt/Windows.Foundation.Collections.h`, following the C++/WinRT rule that you must explicitly include the header files for any namespaces you use. We are using the `Windows::Foundation::Collections` namespace because that's where the `IIterable` class resides, and that is the projected type of the parameter that the `HttpFormUrlEncodedContent` constructor accepts.

This question started out as a "C++/WinRT problem" (which is how I got roped into it), but all of the debugging just treated it as a "C++ problem": It turns out that if there's a particular constructor you want to use, you should make sure the parameter types are defined.

Raymond Chen

Follow