In the debugger, how can I get from a projected type back to the C++/WinRT implementation?

devblogs.microsoft.com/oldnewthing/20221107-00

November 7, 2022

Raymond Chen

Say you are looking at a crash dump, and you have a pointer to a Windows Runtime object projection, and you know that the Windows Runtime object is implemented in C++/WinRT, and you want to get to the implementation type so you can look at its private members.

The basic trick of treating the pointer as the start of an object doesn't work:

```
0:000> ?? s
struct winrt::Contoso::Sample
   +0x000 m_ptr : 0x00000205`9ab9cf20 winrt::impl::abi<winrt::Windows::Foundation::
IUnknown,void>::type
```

If you dump memory starting at this interface pointer, you get the expected vtable, but the rest does not match the expected implementation type.

```
0:000> dps 0x00000205`9ab9cf20
00000205`9ab9cf20  00007ff6`0bef8998 contoso!winrt::impl::produce<winrt::Contoso::
implementation::Sample,winrt::Contoso::ISample>::`vftable'
00000205`9ab9cf28  00007ff6`0bef8a38 contoso!winrt::impl::produce<winrt::Contoso::
implementation::Sample,winrt::Windows::Foundation::IClosable>::`vftable'
00000205`9ab9cf30  00000000`0000002a
00000205`9ab9cf38  00000205`9ab9dae0
00000205`9ab9cf40  00640065`00720046
00000205`9ab9cf48  00000000`00000000
00000205`9ab9cf50  00000000`00000004

0:000> dt contoso!winrt::Contoso::implementation::Sample 0x00000205`9ab9cf20
   +0x010 vtable           : winrt::impl::produce<winrt::Contoso::implementation::
Sample,winrt::Contoso::ISample>
   +0x018 vtable           : winrt::impl::produce<winrt::Contoso::implementation::
Sample,winrt::Windows::Foundation::IClosable>
   +0x000 __VFN_table : 0x00007ff6`0bef8998
   +0x008 m_references     : std::atomic<unsigned __int64>
   +0x020 m_value          : 0n7471174 ← garbage
   +0x028 m_name           : std::basic_string<wchar_t,std::char_traits<wchar_t>,
std::allocator<wchar_t> >

0:000> ?? ((contoso!winrt::Contoso::implementation::Sample*)0x00000205`9ab9cf20)-
>m_name
class std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >
   +0x000 _Mypair          : std::_Compressed_pair<std::allocator<wchar_t>,std::
_String_val<std::_Simple_types<wchar_t> >,1>

0:000> ??
((consolecppwinrt!winrt::MyProject::implementation::Sample*)0x00000205`9ab9cf20)-
>m_name._Mypair
class std::_Compressed_pair<std::allocator<wchar_t>,std::_String_val<std::
_Simple_types<wchar_t> >,1>
   +0x000 _Myval2          : std::_String_val<std::_Simple_types<wchar_t> >

0:000> ??
((consolecppwinrt!winrt::MyProject::implementation::Sample*)0x00000205`9ab9cf20)-
>m_name._Mypair._Myval2
class std::_String_val<std::_Simple_types<wchar_t> >
   +0x000 _Myproxy         : (null) ← garbage
   +0x008 _Bx              : std::_String_val<std::_Simple_types<wchar_t> >::_Bxty
   +0x018 _Mysize          : 0xabababab`fdfdfdfd ← garbage
   +0x020 _Myres           : 0xabababab`abababab ← garbage
```

The reason nothing matches up is that the projection vtable is not at the start of the object. Let’s look at the initial structure again:

```
0:000> dt contoso!winrt::Contoso::implementation::Sample
   +0x010 vtable           : winrt::impl::produce<winrt::Contoso::implementation::
Sample,winrt::Contoso::ISample>
   +0x018 vtable           : winrt::impl::produce<winrt::Contoso::implementation::
Sample,winrt::Windows::Foundation::IClosable>
   +0x000 __VFN_table : Ptr64
   +0x008 m_references     : std::atomic<unsigned __int64>
   +0x020 m_value          : Int4B
   +0x028 m_name           : std::basic_string<wchar_t,std::char_traits<wchar_t>,
std::allocator<wchar_t> >
```

Even though the debugger lists the `ISample` projection vtable first, it's not actually the start of the object. The offset is `0x010`.

The object starts with its private little vtable `__VFN_table`. In this case, we see that the projection vtable is at offset `0x10`, so we can subtract `0x10` to find the true start of the object:

```
0:000> ??
((consolecppwinrt!winrt::MyProject::implementation::Sample*)0x00000205`9ab9cf10)
struct winrt::MyProject::implementation::Sample * 0x00000205`9ab9cf20
   +0x010 vtable           : winrt::impl::produce<winrt::Contoso::implementation::
Sample,winrt::Contoso::ISample>
   +0x018 vtable           : winrt::impl::produce<winrt::Contoso::implementation::
Sample,winrt::Windows::Foundation::IClosable>
   +0x000 __VFN_table : 0x00007ff6`0bef8d68
   +0x008 m_references     : std::atomic<unsigned __int64>
   +0x020 m_value          : 0n42 ← good value
   +0x028 m_name           : std::basic_string<wchar_t,std::char_traits<wchar_t>,
std::allocator<wchar_t> >

0:000> ??
((consolecppwinrt!winrt::MyProject::implementation::Sample*)0x00000205`9ab9cf10)-
>m_name._Mypair._Myval2
class std::_String_val<std::_Simple_types<wchar_t> >
   +0x000 _Myproxy         : 0x00000205`9ab9dae0 std::_Container_proxy
   +0x008 _Bx              : std::_String_val<std::_Simple_types<wchar_t> >::_Bxty
   +0x018 _Mysize          : 4 ← good value
   +0x020 _Myres           : 7 ← good value

0:000> ??
((consolecppwinrt!winrt::MyProject::implementation::Sample*)0x00000205`9ab9cf10)-
>m_name._Mypair._Myval2._Bx
union std::_String_val<std::_Simple_types<wchar_t> >::_Bxty
   +0x000 _Buf             : [8]  "Fred" ← good value
   +0x000 _Ptr             : 0x00640065`00720046  "--- memory read error at address
0x00640065`00720046 ---"
   +0x000 _Alias           : [8]  "F"
```

We happened to know that the `m_ptr` came from a projection of the `Sample` runtime class, which means that it's a pointer to the default interface, which is `ISample`. If `m_ptr` came from a projection f the `IClosable` interface, then we would have to adjust by `0x018` to get to the start of the object.

My strategy is simply to go backward from the projection pointer and see if a non-projection vtable emerges.

```
0:000> dps 0x00000205`9ab9cf20-40 0x00000205`9ab9cf20
00000205`9ab9cee0  00000205`9ab9cdc0
00000205`9ab9cee8  00000205`9ab9dab0
00000205`9ab9cef0  00000000`00000000
00000205`9ab9cef8  00000001`00000000
00000205`9ab9cf00  00000000`00000050
00000205`9ab9cf08  fdfdfdfd`00000118
00000205`9ab9cf10  00007ff6`0bef8d68 contoso!winrt::impl::heap_implements<winrt::
Contoso::implementation::Sample>::`vftable'
00000205`9ab9cf18  00000000`00000001
00000205`9ab9cf20  00007ff6`0bef8998 contoso!winrt::impl::produce<winrt::Contoso::
implementation::Sample,winrt::Contoso::ISample>::`vftable'
```

Raymond Chen

Follow