

Peeking inside WRL weak references

 devblogs.microsoft.com/oldnewthing/20200511-00

May 11, 2020



Raymond Chen

The Windows Runtime C++ Template Library (WRL) also provides a standard implementation of the `IWeakReference` interface. Here's what it looks like:

```
0:000> $ 64-bit version
0:000> dt Microsoft::WRL::Details::WeakReferenceImpl
+0x000 __VFN_table : Ptr64
+0x00c refcount_   : Int4B
+0x010 strongRefCount_ : Int4B
+0x018 unknown_   : Ptr64 IUnknown

0:000> $ 32-bit version
0:000> dt Microsoft::WRL::Details::WeakReferenceImpl
+0x000 __VFN_table : Ptr32
+0x008 refcount_   : Int4B
+0x00c strongRefCount_ : Int4B
+0x010 unknown_   : Ptr32 IUnknown
```

The offsets are different, but the basic idea should look familiar: It's basically the same as [C++/CX weak references](#). The offsets are different because the WRL control block is made up out of multiple base classes, so you end up with alignment padding.

The members serve the same purpose as they did in C++/CX, so I will defer to earlier discussion.

You can find the implementation of WRL weak references in `wrl/implements.h`.

[Raymond Chen](#)

Follow

