Adjustor thunks

devblogs.microsoft.com/oldnewthing/20040206-00

February 6, 2004

Raymond Chen

Yesterday we learned about the layout of COM objects and I hinted at "adjustor thunks".

If you find yourself debugging in disassembly, you'll sometimes find strange little functions called "adjustor thunks". Let's take another look at the object we laid out last time:

```
class CSample : public IPersist, public IServiceProvider
{
public:
  // *** IUnknown ***
  STDMETHODIMP QueryInterface(REFIID riid, void** ppv);
  STDMETHODIMP_(ULONG) AddRef();
  STDMETHODIMP_(ULONG) Release();
  // *** IPersist ***
  STDMETHODIMP GetClassID(CLSID* pClassID);
  // *** IQueryService ***
  STDMETHODIMP QueryService(REFGUID guidService,
                  REFIID riid, void** ppv);
private:
  LONG m_cRef;
  ...
};
```

p	→	lpVtbl	→	→	→	QueryInterface (1)
q	→	lpVtbl	→	QueryInterface (2)		AddRef (1)
		m_cRef		AddRef (2)		Release (1)
		…		Release (2)		GetClassID (1)
				QueryService (2)		

In the diagram, p is the pointer returned when the IPersist interface is needed, and q is the pointer for the IQueryService interface.

Now, there is only one QueryInterface method, but there are two entries, one for each vtable. Remember that each function in a vtable receives the corresponding interface pointer as its "this" parameter. That's just fine for QueryInterface (1); its interface pointer is the same as the object's interface pointer. But that's bad news for QueryInterface (2), since its interface pointer is q, not p.

This is where the adjustor thunks come in.

The entry for QueryInterface (2) is a stub function that changes q to p, and then lets QueryInterface (1) do the rest of the work. This stub function is the adjustor thunk.

```
[thunk]:CSample::QueryInterface`adjustor{4}':
  sub     DWORD PTR [esp+4], 4 ; this -= sizeof(lpVtbl)
  jmp     CSample::QueryInterface
```

The adjustor thunk takes the "this" pointer and subtracts 4, converting q into p, then it jumps to the QueryInterface (1) function to do the real work.

Whenever you have multiple inheritance and a virtual function is implemented on multiple base classes, you will get an adjustor thunk for the second and subsequent base class methods in order to convert the "this" pointer into a common format.

Raymond Chen

Follow