

## Re: Getting class pointer from interface pointer [don't use structures?]

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vc/2006-12/msg00273.html>

---

- *From:* "raylopez99" <[raylopez99@xxxxxxxxxx](mailto:raylopez99@xxxxxxxxxx)>
  - *Date:* 19 Dec 2006 03:17:41 -0800
- 

I'm not a professional programmer, just a hobbyist, but I recall reading a book on MFC programming by Richard M. Jones "Intro to MFC Programming with Visual C++" that indicated he had a bug involving a structure used inside of a class, and for the life of him he could not get rid of the bug until he changed the structure to a class. Since Jones is a professional and teaches as well, I concluded structures are poorly supported by Microsoft—change to a class and see if that helps. Compilers do stuff behind the scenes, especially with dotnet languages where everything is 'virtual' and/or placed on the heap, that you may not be aware of.

RL

Rory Plaire wrote:

Sure – thanks.

Say I have this struct definition:

```
struct IUserObject
{
    RESULT Invoke(int method) = 0;
    void Destroy() = 0;
}
```

And this class definition:

```
class ManagedWrapper : IUserObject
{
    RESULT Invoke(int method);
    void Destroy() ;
}
```

And finally, a method to get an instance of the base struct (interface):

```
IUserObject* GetUserObject();
```

Re: Getting class pointer from interface pointer [don't use structures?]

When I invoke the method to get the interface-typed instance, I can't cast it reliably:

```
ManagedWrapper* wrap = dynamic_cast<ManagedWrapper*>(GetObject());
```

... this invocation is always results in a pointer referenceing a location a DWORD preceding the ManagedWrapper\* implementation, and all dereferences are then invalid since they are all off by -4 bytes.

Can you see something I can't?

-r

"Peter Ritchie [C# MVP]" wrote:

Sorry, when you mentioned non-managed interface I assumed COM since C++ doesn't have the concept of an "interface", just struct or class.

Can you give me an example of what you're doing that doesn't work?