

## Re: PInvokes with managed and unmanaged objects

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.interop/2004-02/0215.html>

---

**From:** James Lapalme (*lapalmejames\_at\_hotmail.com*)

**Date:** 02/09/04

Date: Mon, 9 Feb 2004 11:06:05 -0500

Thank you,

The reason I'm asking all these questions is that I'm a grade student (C.S) looking at different ways of reusing C/C++ libraries without too much modification. I will explore the wrapping route after I get this one down.

Where could I find more information on PInvokes when inheritances and interfaces come into play?

I have read many books on Com/.Net interop but they all limit themselves to basic types, strings and structs.

I have not found all the answers to my questions on the msdn site either.

JL

"Zoiner Tejada" <anonymous@discussions.microsoft.com> wrote in message news:046943C9-0FE9-46CF-BB24-3A43BBDE2210@microsoft.com...

> > *If I have a C++ class that use inheritance, I guess I must redefined in C#*

> *all of its super classes?*

>

> *That's an interesting question and by interesting I mean hard to answer!*

In reality the question I would ask first is if C++ is your favorite development language then why not just start using managed C++? I suppose if this along the lines of your previous post, then what you are mainly talking about is a C# wrapper class that encapsulates PInvoke calls to your DLL. In this case, then yes you have to write all the DLLImport's for whatever functions you need but you don't necessarily have to follow the same inheritance as in C++ file, you get the privilege of flattening the hierarchy as you see fit.

>

> > *Also, if I have a C++ class that uses multiple inheritance, is it possible*

> *to be represented in C# for PInvoke marshalling? If not any way around it?*

>

microsoft.public.dotnet.framework.interop: Re: PInvokes with managed and unmanaged objects

> *Because C# does not allow multiple inheritance, the only way to simulate*  
it is to use interfaces. This can be a workaround because, while you can  
only inherit from one class at a time, you can implement multiple  
interfaces.

>

> *Happy Coding,*

> *Zoiner Tejada*

>

>

> *Visit my new VB.NET in 2D Guide for tips on calling the Win32 API using*  
PInvoke.

> <http://www.tejadaz.com/VBdotNet>

>