

Re: Pointer

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2006-12/msg01036.html>

- *From:* "Willy Denoyette [MVP]" <willy.denoyette@xxxxxxxxxx>
 - *Date:* Mon, 4 Dec 2006 11:42:28 +0100
-

"Sheikko" <sheikko@xxxxxxxxxx> wrote in message
news:1165221084.208747.240260@xx

Hi, friends. I am new in this group and I have a problem. I am just started to program in cSharp and I need a help. I need to import a dll and to use one of its function. The function, in C language, require in input a pointer char:

```
OpenFile(char* filename);
```

To use this function in cSharp I have writing the following code:

```
=====
[DllImport("My.dll")]
unsafe public static extern int OpenFile(char* filename);
.....
unsafe private void buttonGetDllTypes_Click(object sender, EventArgs e)
{
fileToOpen = "C:\\myFile.ext";
fixed (char* p = fileToOpen)
{
OpenFile(p);
}
}
}
=====
```

But it doesn't work. I don't know if this function is deprecated or there is a problem in the calling him. When I build or run my project there is not any error. I want to know the list of functions in the dll. Can I retrieve these from the dll?

Can someone Help me. Thanks

Well, you are new to CSharp and you start treating C# as if it was C, you will miss your start because C# isn't the same as C.

Re: Pointer

First, you should stay away from unsafe constructs whenever possible. Second, you should stay away from interop features like PInvoke, at least when you start learning the language and (more importantly) the framework.

To achieve the first goal, remove the unsafe from signature (DllImport) and change the argument as follows:
`public static extern int OpenFile(string filename);`

and, change your code into:

```
fileToOpen = "C:\\myFile.ext";  
OpenFile(fileToOpen);
```

this leaves you with one common issue, that is – what is a `char*` in C –.

What is it pointing to exactly, is it pointing to a "Unicode character" array or is it pointing to a "Single byte character" array or is it pointing to something else?

You should know that a 'char' in C# is Unicode encoded, that is it's a 'wide' character, and a string in .NET is an array of char's. This means that somehow you need to indicate to the PInvoke layer what the native code `char*` looks like, if you don't give an hint, PInvoke considers `char*` a pointer to a null terminated "single byte character" array. Indicating what the destination points to, can be done by using the `MarshalAs` attribute. For instance if the native code expects a pointer to a "Wide character" array, you have to apply the following attribute to the argument, like :

```
public static extern int OpenFile([MarshalAs(UnmanagedType.LPWStr)] string filename);  
if it's pointing to a single byte character array, you can change the signature into this:  
public static extern int OpenFile([MarshalAs(UnmanagedType.LPStr)] string filename);
```

The second goal can be achieved by using the framework classes instead of calling into unmanaged code, in the above case, it's just a matter of using the `FileStream` classes in order to access files on disk.

Please consult the docs for details on other possible marshaling types and consult the docs whenever you dive into interop or whatever. Don't start coding in C# like you would in C, read the docs, leverage the framework classes first, learn the C# language and forget about C (coding styles) and it's libraries.

Another remark, you declared the `OpenFile` to return an `int`

1) are you sure it returns an `int`?

2) why don't you use the return value in your code?

If you don't have access to the documentation of the library functions you are calling, you will have to guess both the arguments and return types, IMO in this case you shouldn't call into the library at all.

And last but not least, please be more explicit when posting questions like this, post the error message(s), stack traces etc....

"But it doesn't work. " isn't of any help

Willy.

.