

RE: Newbie Qn – Interfaces as parameters and members

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005-08/msg02278.html>

- *From:* "steve" <steve@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 10 Aug 2005 22:46:11 -0700
-

Thanks for that Mark.

I thought it was probably ok.. at least the compiler didn't complain

"Mark R. Dawson" wrote:

```
> Hi Steve,
> no reason why you cannot do this. The idea of having an interface is that
> you can abstract away in your code from having to worry about the actual
> concrete class that implements the interface, you don't care what type the
> object is in your functions as long as it implements your interface.
>
> Mark.
>
> "steve" wrote:
>
>> Hi all
>>
>> I want to know if I can have interfaces as formal parameters in methods.
>>
>> For instance, if I have something like this
>>
>> public interface INameTranslator
>> {
>>     string sToLocal(string sRemoteName);
>>     string sToRemote(string sLocalName);
>> }
>>
>> Can I do this..
>>
>> public class CMyNameTranslator: INameTranslator
>> {
>>     public string sToLocal(string sRemoteName)
>>     {
>>         return sRemoteName;
>>     }
>> }
```

RE: Newbie Qn – Interfaces as parameters and members

```
>> public string sToRemote(string sLocalName)
>> {
>> return sLocalName;
>> }
>> }
>>
>> public class CFoo
>> {
>> public INameTranslator mNameTranslator;
>> CFoo(INameTranslator NameTranslator) //<<---- Is this legal /desirable
>> {
>> mNameTranslator = NameTranslator;
>> }
>> }
>>
>> AND then somewhere else..
>>
>> CMyNameTranslator NameTranslator = new CMyNameTranslator();
>> CFoo Foo = new CFoo(NameTranslator);
>> string sTemp = Foo.mNameTranslator.sToLocal("Bob"); // sTemp should be
>> "Bob"
>>
>> Cheers
>>
>> Steve
>>
.
```

• **References:**

◆ **[Newbie Qn – Interfaces as parameters and members](#)**

◇ *From:* steve

◆ **[RE: Newbie Qn – Interfaces as parameters and members](#)**

◇ *From:* Mark R. Dawson

• Prev by Date: **[RE: Cannot capture DEL key in DataGrid event](#)**

• Next by Date: **[Re: class composition](#)**

• Previous by thread: **[RE: Newbie Qn – Interfaces as parameters and members](#)**

• Next by thread: **[RE: Newbie Qn – Interfaces as parameters and members](#)**

• Index(es):

◆ **[Date](#)**

◆ **[Thread](#)**