

Re: Passing values to a property in C#?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-01/msg01505.html>

- *From:* "Christof Nordiek" <cn@xxxxxxxxxx>
 - *Date:* Thu, 11 Jan 2007 11:54:43 +0100
-

<johnj@xxxxxxxx> schrieb im Newsbeitrag
<news:1168511494.566821.74690@xx>

I am converting an application I wrote in VB.NET to C#. I am not a C# guy, at all. I am having a problem passing values to a property in C#. I understand that it is probably not the best way of achieving my result, but it works in VB.NET, flawlessly.

Basically, I am storing a bunch of settings in a hashtable. Then, setting/getting my values in the hashtable with a public shared property. Below is the property. The function GetSetting's return will look similar to this....

```
Return System.Web.HttpContext.Current.Application("config")(item)
```

```
Public Shared Property Settings(ByVal Item As String) As String
```

```
Get  
Return CType(GetSetting(Item), String)  
End Get
```

```
Set(ByVal Value As String)
```

```
Try
```

```
System.Web.HttpContext.Current.Application("config")(Item) = Value
```

```
Catch ex As Exception
```

```
GetSetting(Item)
```

```
System.Web.HttpContext.Current.Application("config")(Item) = Value
```

```
End Try
```

```
End Set
```

Re: Passing values to a property in C#?

End Property

This is a parameterized property, which doesn't exist in C#. But in C# exist indexers, which are equivalent to default properties in VB.Net.

When I convert to C#, Visual Studio is screaming at me. And here is my C# version....

```
public static string Settings(string Item)
{
    get
    {
        return ((string)(GetSetting(Item)));
    }
    set
    {
        try
        {
            System.Web.HttpContext.Current.Application["config"][Item] = value;
        }
        catch (Exception ex)
        {
            GetSetting(Item);
        }
    }
}
```

You could translate this as:

```
public static string this [string key]
{
    get
    {
        return ((string)(GetSettings(key)));
    }
    set
    {
        try
        {
            System.Web.HttpContext.Current.Application["config"][key] = value;
        }
    }
}
```

Re: Passing values to a property in C#?

Re: Passing values to a property in C#?

```
}  
catch (Exception ex)  
{  
    GetSetting(Item);  
  
    System.Web.HttpContext.Current.Application["config"][key] = value;  
}  
}  
}
```

To access it use:

`ClassName[key]`

which can be read and written like a property access.

`ClassName` is the name of the class the indexer is declared in. If the indexer is not static, it is an instance expression of that class.

One disadvantage is, that you can't distinguish them by name. But you can overload them like methods.

Otherwise you have to declare the getter or setter as separate methods.

hth

.