

C# Events & Event Handlers

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2006-09/msg03399.html>

- *From:* "Scott M." <NoSpam@xxxxxxxxxxx>
 - *Date:* Wed, 20 Sep 2006 22:20:47 -0400
-

I'm trying to get my head around what's involved in C# to declare, raise and respond to an event.

I'm coming from the VB.NET world where to declare and event we wrote:

```
Public Event Foo()
```

To raise the event, we wrote:

```
RaiseEvent Foo()
```

To create an instance that could respond to events, we wrote:

```
WithEvents x As SomeTypeThatHasTheFooEvent
```

And finally, we wrote the event handler as a regular method, but with a "handles" clause:

```
Sub Foo() Handles x.Foo()
```

Can someone please explain the C# for this VB.NET code? I've looked into it and I'm getting code like this, but I don't know exactly where it should go and what it does:

```
// The following is like the VB.NET Public Event Foo() declaration, right?  
public event FooEventHandler Foo;
```

```
// I have no idea what this does!  
public delegate void FooEventHandler();
```

```
//The following raises the event?  
Foo();
```

```
// Now, in the code that will use this class:  
// I have no idea what this is all about!  
x.Foo += new System.EventHandler(x_Foo);
```

```
//Then, to handle the event in the code that uses the class:
```

C# Events & Event Handlers

```
private static void myCar_SpeedLimitExceeded()
{
// do something to handle the event here;
}
```

I have everything where I believe it should be, but I am getting a build error on this line:

```
x.Foo += new System.EventHandler(x_Foo);
```

The error indicates that I can't implicitly convert a System.EventHandler to a FooEventHandler

-Thanks!

.