

Use VB and C# Code in Limnor Applications

1	Introduction.....	1
2	Use UserFunction Performer	1
3	VB Code Sample	2
4	C# Sample.....	4
5	Use the Performer in Applications.....	6

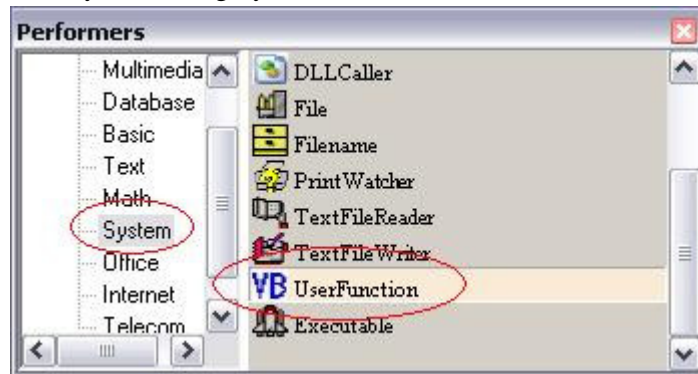
1 Introduction

UserFunction is a Performer allowing you to use VB.NET and C# code inside your Limnor applications. It is an alternative and simple way of developing Performers to extend Limnor’s programming capability.

The public functions, properties and events you create in your code automatically become methods, properties and events of the UserFunction Performer. Therefore your applications use VB and C# code in the same codeless manner.

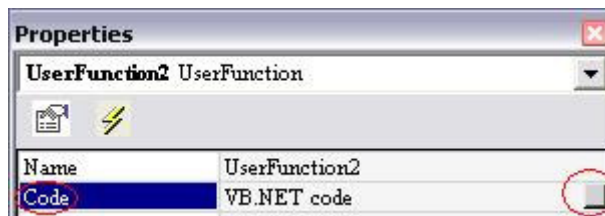
2 Use UserFunction Performer

This performer is under “System” category in the tool box:



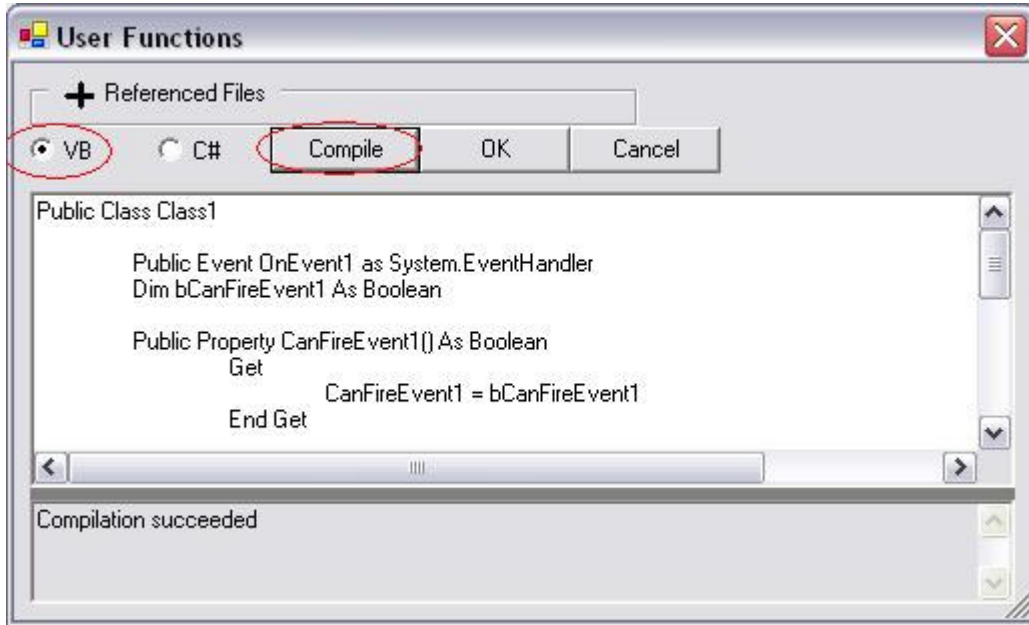
Note that although the icon show “VB”, the same Performer also support C#.

Add a UserFunction Performer to a page or to the Application Performer to use it. To write code, set its Code property:



3 VB Code Sample

You write your code in the text box, and click “Compile” button to check syntax errors. The compilation error message will be displayed:



The VB.NET code sample we use here is:

```
Public Class Class1

    Public Event OnEvent1 as System.EventHandler
    Dim bCanFireEvent1 As Boolean

    Public Property CanFireEvent1() As Boolean
        Get
            CanFireEvent1 = bCanFireEvent1
        End Get
        Set(ByVal Value As Boolean)
            bCanFireEvent1 = Value
        End Set
    End Property


    Public Sub Test1()
        System.Windows.Forms.MessageBox.Show ("Hello from VB code. CanFireEvent1 = " &
        bCanFireEvent1)
        If bCanFireEvent1 Then
```

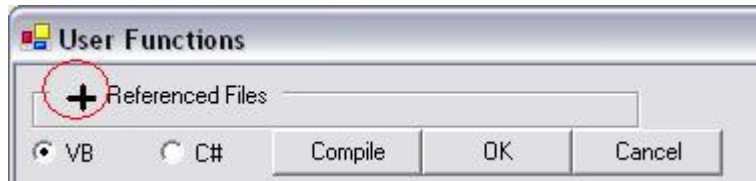
```
        RaiseEvent OnEvent1(Me, Nothing)
    End If
End Sub
Public Sub Test2(b As Boolean)
    bCanFireEvent1 = b
End Sub
End Class
```

In this code, we define two functions, Test1 and Test2; one property CanFireEvent1; and one event OnEvent1.

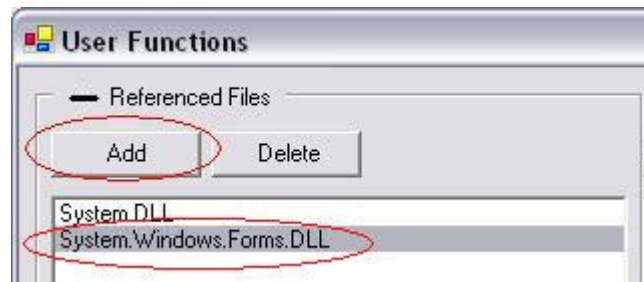
- An event, OnEvent1, is used to demonstrate creating event.
- A private variable, bCanFireEvent1, is used to control whether event OnEvent1 should fire.
- Function Test1 is called to fire event OnEvent1. In Test1, a message box is displayed showing message with the value of bCanFireEvent1: "Hello from VB code. CanFireEvent1 = " & bCanFireEvent1. Then the value of bCanFireEvent1 is checked. If bCanFireEvent1 is true then the event OnEvent1 is fired.
- Property CanFireEvent1 exposes variable bCanFireEvent1 as a public property.
- Function Test2 is used to demonstrate a function with parameters. It has a Boolean parameter which will be assigned to bCanFireEvent1.

Note that the code System.Windows.Forms.MessageBox.Show uses MessageBox object which is in a .NET DLL, System.Windows.Forms.DLL. We need to instruct the compiler to use this DLL. To do it,


click  :



Click Add button and type System.Windows.Forms.DLL:



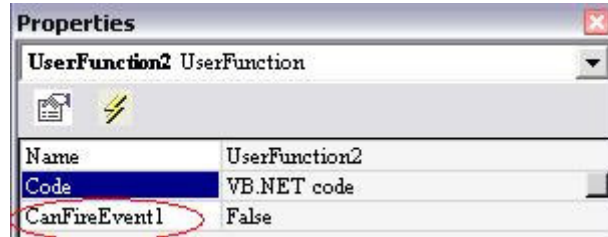
Do not use path for System.Windows.Forms.DLL because it is a system DLL, not your own DLL.

After adding all DLL files, click  to hide DLL list.

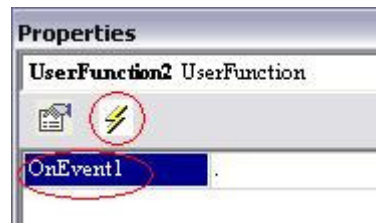
Click OK button when you finish the coding.

The property CanFireEvent1 becomes a property of the Performer:

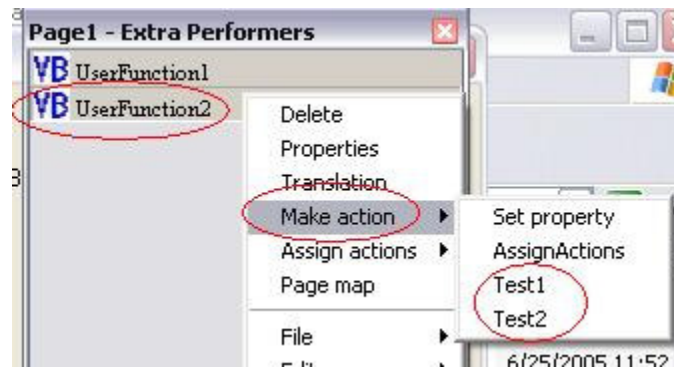
Use VB & C# in Limnor



The event OnEvent1 becomes an event of the performer:

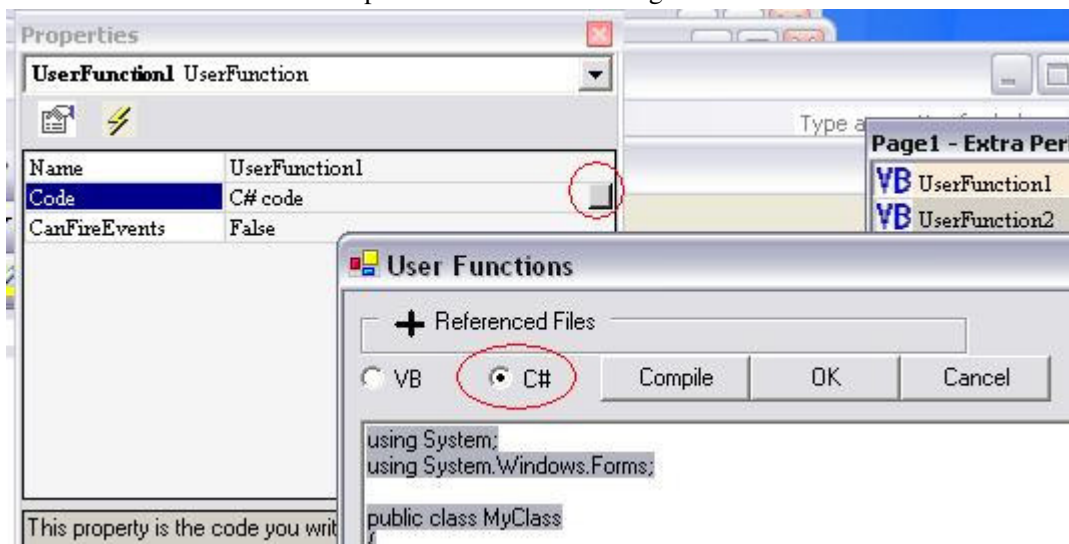


The function Test1 and Test2 become two functions of the performer:



4 C# Sample

We need add another UserFunction performer to do C# coding.



The C# code sample in this sample application is:

```
using System;
using System.Windows.Forms;

public class MyClass
{
    public event EventHandler OnEvent1 = null;
    protected bool bCanfireEvent1 = false;
    public void test()
    {
        MessageBox.Show("Hello from C# code. CanFireEvents="+bCanfireEvent1.ToString() );
        if( bCanfireEvent1 )
        {
            if( OnEvent1 != null )
            {
                OnEvent1(this,null);
            }
        }
    }
    public void test2(bool b)
    {
        bCanfireEvent1 = b;
    }
    public bool CanFireEvents
    {
        get
        {
            return bCanfireEvent1;
        }
        set
        {
            bCanfireEvent1 = value;
        }
    }
}
```

It has the same functionality as the VB sample code. We do not need to go into details again here.

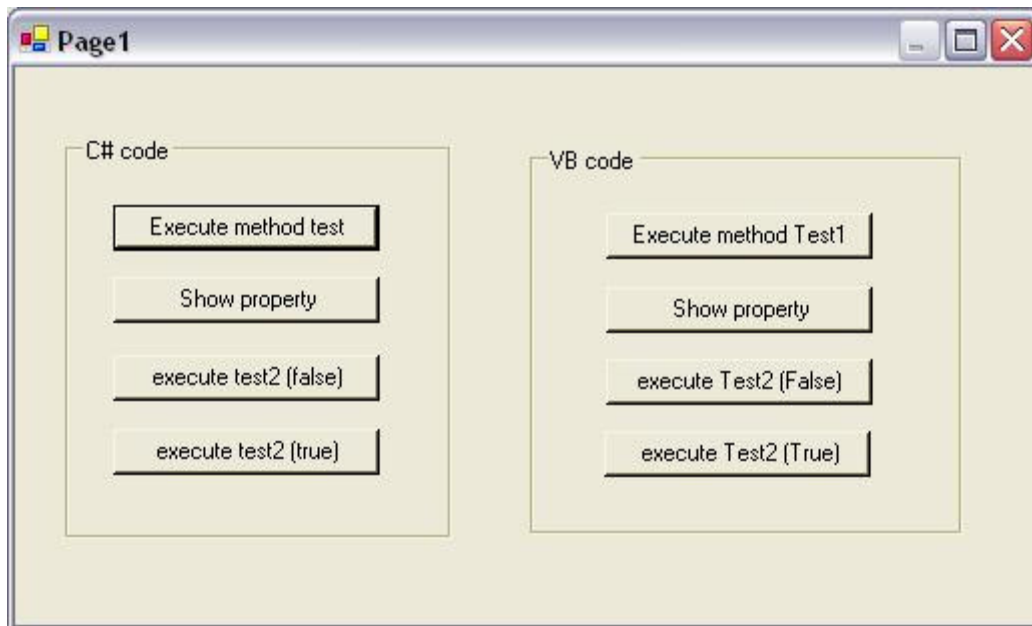
5 Use the Performer in Applications

Now we have properties, methods and events, there is no difference in using your VB/C# code than using all other Performers.

In this sample, we created some actions to show the use of UserFunction performers:

	Action name
▶	Application.OnEvent1FromC#
	Application.OnEvent1FromVB
	Application.ShowPropertyC#
	Application.ShowPropertyVB
	UserFunction1.test
	UserFunction1.test2FALSE
	UserFunction1.test2TRUE
	UserFunction2.Test1
	UserFunction2.Test2False
	UserFunction2.Test2True

Application.OnEvent1FromC# and Application.OnEvent1FromVB are assigned to Onevent1 events of the C# and VB code respectively; other actions are assigned to buttons:



You may download the sample application from <http://www.limnor.com/downloads/code.cab>