

# Using Windows Forms in a Vulcan.NET VO GUI Application

Paul Piko, February 2010

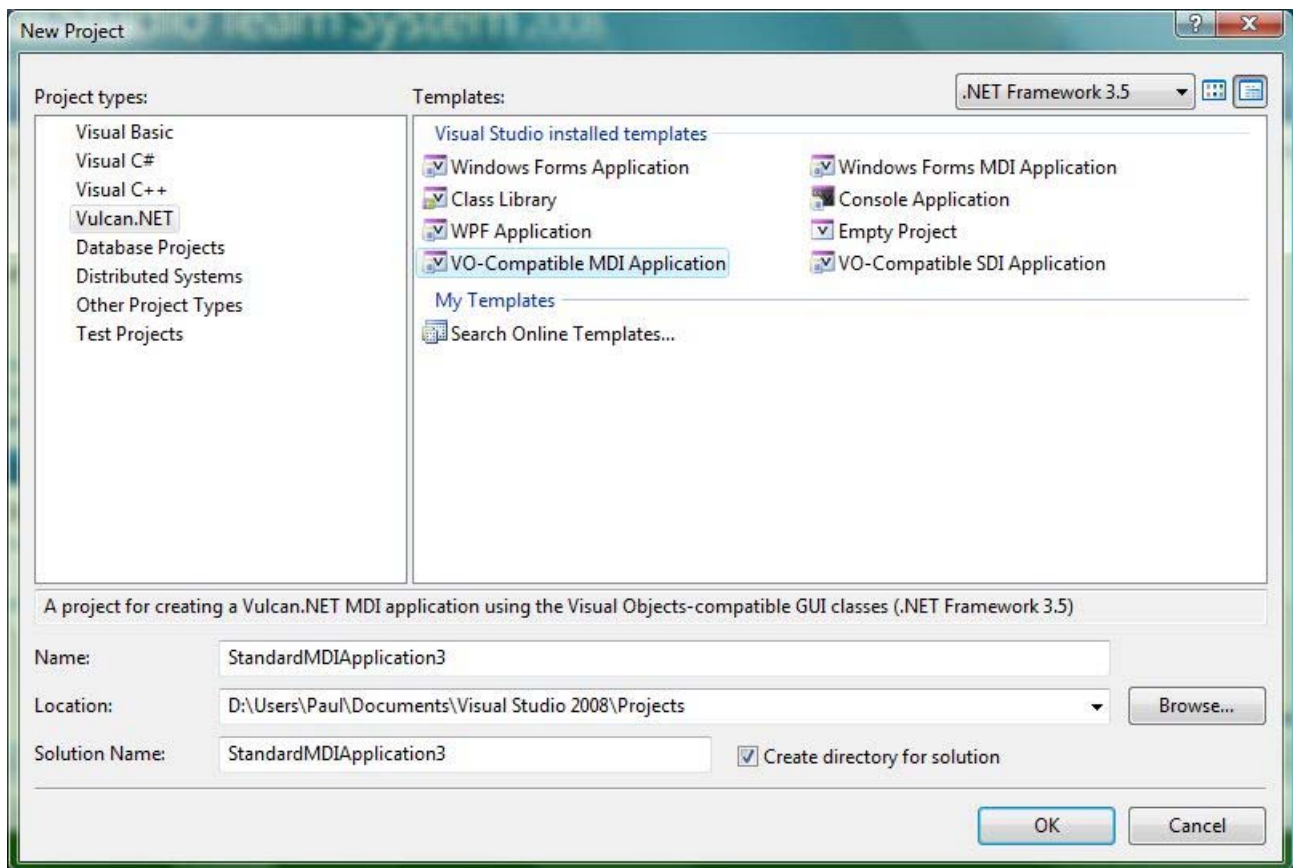
This article shows how to use a .NET Form in a Vulcan.NET application based on the VO GUI classes.

To achieve this there are two main steps:

- instead of using the App class, use VOWinFormApp
- use a ChildWinForm window to contain the .NET Windows Form

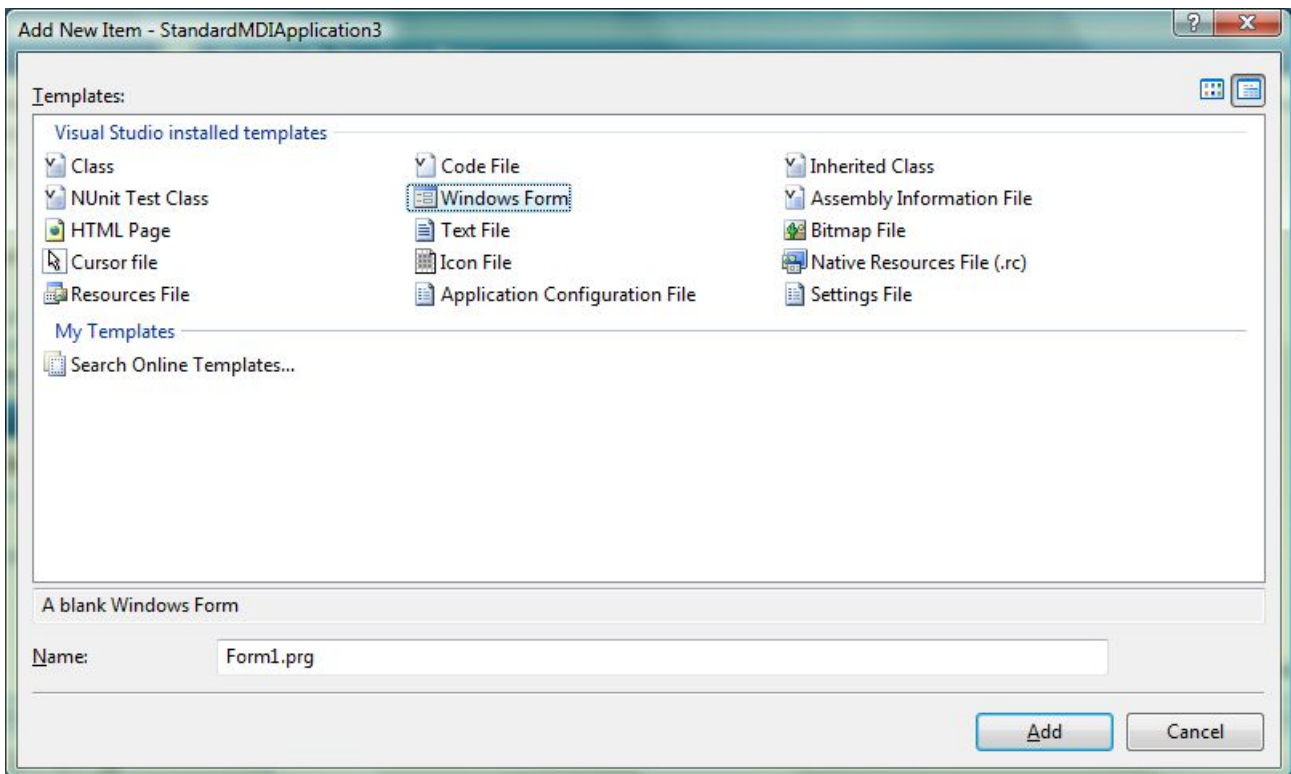
## Create Solution

You can either transport an existing VO application or create a VO-Compatible MDI Application using the Visual Studio New Project dialog. In this example we will do the latter, as shown below.

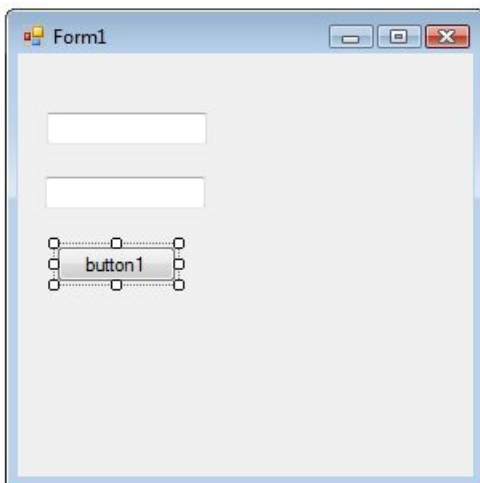


## Create a .NET Windows Form

Right-click on the project and select Add, New Item. The Add New Item dialog is displayed. Create a Windows Form.



Put controls on to the form.



## Display the Form

Normally you would update your menu to call the new form. For simplicity in this example we create the Windows Form in the application Start.

Firstly, rather than create an instance of App we need to use VOWinFormApp.

Next we create an instance of the Windows Form, and assign it to a ChildWinForm window. ChildWinForm is a subclass of VO's ChildAppWindow.

```
#include "VOGUIClasses.vh"

[STAThread] ;
FUNCTION Start() AS VOID

    LOCAL oApp AS App
    LOCAL oMainWindow AS StandardShellWindow
    RDDSetDefault( "DBFCDX" )
    TRY
```

```

oApp      := VOWinFormApp{}
oMainWindow := StandardShellWindow{}

oMainWindow:Show(SHOWCENTERED)

LOCAL o AS System.Windows.Forms.Form
LOCAL oChild AS ChildWinForm

o := Form1{}
oChild := ChildWinForm{oMainWindow,TRUE}
oChild:WinForm := o
oChild:Caption := "Windows Form"

oChild:Show()

oApp:Exec()

CATCH e AS Exception
    ErrorDialog( e )
END TRY

RETURN

```

Running the application shows the .NET Windows Form within the VO GUI ShellWindow.

