**Windows Mobile Application Development**

**(Part 1)**

**Topic 1: Visual Studio 2008 and additional tools
Creating your first Windows Mobile application**

[**http://msdn.microsoft.com/rampup**](http://msdn.microsoft.com/rampup)

**Introduction**

Microsoft Visual Studio 2008 Professional Edition or better provides a development environment that allows you to build applications in either native code (C / C++) or managed code (C# / Visual Basic.NET) for Windows Mobile Devices.

This article provides information to correctly install Visual Studio 2008 and additional Windows Mobile Software Development Kits in order for you to create your first Windows Mobile application, targeting a Windows Mobile 6 device. This article also introduces the Device Emulator which allows you to test your Windows Mobile applications without the need to have a physical Windows Mobile device available.

**Installation of Visual Studio 2008 and additional SDK’s**

Let us assume you have a brand new development machine that you want to use to develop applications for Windows Mobile devices. The first thing you want to do is install Visual Studio 2008 and, depending on the speed of your Internet connection you might also want to install all MSDN help files locally on your development system. If you don’t own Visual Studio 2008 you can download an evaluation version of [**Visual Studio 2008 Professional**](http://msdn.microsoft.com/en-us/vstudio/products/aa700831.aspx). Installing both Visual Studio 2008 and MSDN Library for Visual Studio 2008 is pretty straight forward. After a number of installation prerequisite files have been installed on your development machine, immediately after you install the DVD containing Visual Studio 2008 Professional Edition in the DVD drive, you will see an setup wizard. By just selecting the default options, you will install Visual Studio 2008 with Smart Device Development enabled.

After installation of Visual Studio 2008 and the MSDN Library for Visual Studio 2008 there is one more thing you might want to do. If you take a look at the installation dialog for both Visual Studio 2008 and the MSDN Library, you can see a hyperlink to check for updates. It is my recommendation to do so, since this guarantees you to install the latest security updates, depending on the operating system you are running on.

Next you need to download and install [**Visual Studio 2008 Service Pack 1**](http://www.microsoft.com/downloads/details.aspx?FamilyId=FBEE1648-7106-44A7-9649-6D9F6D58056E&displaylang=en). First a relatively small installer will be downloaded, which downloads all necessary Service Pack 1 features, depending on the way you installed Visual Studio 2008. Depending on your download speed this might be a lengthy operation.

Visual Studio 2008 should now work properly on your development machine. To verify correct installation of Visual Studio 2008 it is now time to start it from the Windows Start Menu. After Visual Studio 2008 has started you can create a new Windows Mobile application by clicking **File** - **New Project**. In the *New Project* dialog that is displayed you can now select to create a Smart Device project, either in C# or in Visual Basic.NET. Let’s assume you want to create a Windows Mobile 6 application written in C#. To do this, expand the **C#** node in the **Project Types** window and select **Smart Device**. Visual Studio 2008 will give your project a default name and a default storage location. Optionally you can give your project a name and you can specify a location where your project will be stored. This is all the information you need to provide in this dialog. To continue creating your application you must now click the **OK** button in the *New Project* dialog. This action will open the *Add New Smart Device Project* dialog in which you can specify what type of project you want to create and what additional settings you want for your project.

Before you continue creating your first Windows Mobile application, expand the **Target Platform** combo box. If you just installed a clean copy of Visual Studio 2008 you will see that you can create projects for a limited number of devices and that there is no support to create Windows Mobile 6 projects. In order to create Windows Mobile 6 projects, you must first install one or both of the Windows Mobile 6 SDKs. An SDK or Software Development Kit adds documentation, sample code, header and library files, emulator images and tools to Visual Studio that let you build applications for a particular target. You need to install these SDKs separately because they were released independent and at a later time then Visual Studio 2008. More SDKs and additional emulator images might be available at a later time. To easily find new SDKs, there is a hyperlink to the [**Windows Mobile Developer Center**](http://msdn.microsoft.com/en-us/windowsmobile/default.aspx) on the *Add New Smart Device Project* dialog, from which you can download and install additional SDKs and emulator images. To make sure that the SDKs install properly you must make sure to close Visual Studio 2008.

NOTE: If you had the Windows Mobile 6 SDKs already installed, for instance to target Windows Mobile 6 devices from inside Visual Studio 2005, you need to either reinstall or repair the SDKs after installing Visual Studio 2008. If you don’t do this, you cannot target Windows Mobile 6 devices from inside Visual Studio 2008.

Figure - New Smart Device Project dialog

If you have installed one or both of the Windows Mobile 6 SDKs you are ready to create your first application for a Windows Mobile 6 device. In order to repeat the steps shown in this document, you at least will need to install the Windows Mobile 6 Professional SDK.

**Creating your first Windows Mobile application**

With all tools being installed you are now ready to create your first Windows Mobile 6 application. For this application you will target the Windows Mobile 6 Professional SDK. However, you can build applications and solutions for a large number of Windows Mobile devices and for generic Windows CE devices. With both Windows Mobile 6 SDKs install you can develop solutions for the following devices using Visual Studio 2008:

* Pocket PC 2003
* Windows CE (non Windows Mobile devices)
* Windows Mobile 5.0 Pocket PC
* Windows Mobile 5.0 Smartphone
* Windows Mobile 6 Professional
* Windows Mobile 6 Standard

With Visual Studio you can target two different versions of the .NET Compact Framework, a subset of the full .NET Framework that is used to write managed applications for Smart Devices. The following versions of the .NET Compact Framework can be targeted with Visual Studio 2008:

* .NET Compact Framework Version 2.0
* .NET Compact Framework Version 3.5

If you need to develop applications for older devices, or if you need to target a previous version of the .NET Compact Framework you must use Visual Studio 2005. This is beyond the scope of this article, but you can install Visual Studio 2005 and Visual Studio 2008 side by side.

### To create a new Smart Device Project

1. Open Microsoft Visual Studio 2008.
2. On the **File** menu, click **New**, and then select **Project**.
3. In **Project types**, select **Visual C#**, and then select **Smart Device**.
4. Specify a new name, location, or solution name for your Smart Device Project, and then click **OK**. An additional dialog similar to will open in which you can specify the type of project you want to create, the SDK you want to target and the version of the .NET Compact Framework you want to use.
5. In the **Target Platform** list select **Windows Mobile 6 Professional SDK**.
6. In the **.NET Compact Framework version** list select **.NET Compact Framework Version 3.5**.
7. In the available **Templates** select **Device Application**.
8. Click **OK** to create a new Smart Device solution project, which includes the following:
	* References to the necessary assemblies
	* AssemblyInfo.cs, a file that enables you to specify company, product information and version information for the Smart Device application
	* Form1.cs, a Visual C# code file that you can use to create the initial user interface for your Smart Device application (you can either view the source code in an edit window or the form in designer mode)
	* Program.cs, a Visual C# code file containing the entry point of your application

If you have created your first Smart Device project successful, you will see the Form Designer in Visual Studio 2008, displaying your empty form.

Figure - Visual Studio 2008 Form Designer

You can see several important areas inside Visual Studio 2008 in . The large area on the left hand side of Visual Studio 2008 that shows a skin of a Windows Mobile device is the Form Designer. You can add user interface controls by dragging them from the Toolbox and dropping them on the desired location inside **Form1**. The upper window on the right hand side of Visual Studio 2008 is the **Solution Explorer**. It contains a tree view of all your projects and files inside projects that are part of the solution you are working on. Under **Solution Explorer** you can see the **Properties Window**, which allows you to set properties for currently selected user interface controls.

NOTE: If the Properties window is not visible in your copy of Visual Studio 2008, you can make it visible by selecting **View** and then **Properties Window** on the Visual Studio 2008 menu.

Another important part inside your working area of Visual Studio 2008 is the **Target Device** dropdown box, which is marked by a red oval in . Here you can select to which device or which particular device emulator your application will be deployed once you are done developing it.

### To add functionality to your application

1. Drag a **Button** from the **Toolbox** and drop it anywhere on **Form1**.
2. In the **Properties Window**, change the **(Name)** property from **button1** to **btnClickMe**.
3. Also in the **Properties Window**, change the **Text** property from **button1** to **Click ME!**
4. Drag a **Label** from the **Toolbox** and drop it anywhere on **Form1**.
5. Resize the just added **Label** inside **Form1** so it occupies the entire width of **Form1**.
6. In the **Properties Window**, change the **(Name)** property from **label1** to **lblInfo**.
7. Also in the **Properties Window**, empty the contents of the **Text** property.
8. Make the properties of **Form1** visible in the **Properties Window** by clicking anywhere on **Form1**.
9. Set the **MinimizeBox** property to **False** to allow you to terminate the application by clicking the OK button that now will be displayed in the upper right hand corner of **Form1**.
10. Double click on the **Click ME!** Button to add a **Click** event handler to the button.
11. On the location where the cursor is now located inside the source code editor, add the following code:

 lblInfo.Text = "Button '" +

 btnClickMe.Text +

 "' clicked at " +

 DateTime.Now.ToLongTimeString();

NOTE: You will see syntax coloring inside the code editor and Intellisense will assist you entering this statement.

### To build and deploy your application

1. From the **Build** menu, select **Build Solution**.

Assuming you didn’t make any errors, you will see the following message in the status bar of Visual Studio 2008: **Build succeeded**.
2. Make sure that the **Windows Mobile 6 Classic Emulator** is selected in the **Target Device** dropdown box.
3. From the **Debug** menu, select **Start Debugging**.

After some time you will see the Windows Mobile 6 Classic Emulator starting. If necessary the .NET Compact Framework 3.5 will be deployed first to the emulator. Finally your application will be deployed to the device where it is automatically started.

Figure - Your first application running in the Device Emulator

### To test your application

1. Inside your application that is running in the Device Emulator, click the **Click ME!** Button and you should see some text appearing in the label you have created on your form.
2. Repeat clicking the button several times, and you should see the time inside your label being updated.
3. Click on **OK** in the upper right hand side of your application to terminate it.
4. Close down the emulator by selecting **File** on its menu, followed by **Save State and Exit**.
5. Close down Visual Studio 2008

# **Additional resources and references**

* [Window Mobile Developer Center](http://msdn.microsoft.com/en-us/windowsmobile/default.aspx)
* [Introduction to development with Microsoft Windows Mobile 6](http://msdn.microsoft.com/en-us/library/bb158532.aspx)
* [Development Tools and Resources for Windows Mobile 6](http://msdn.microsoft.com/en-us/library/bb158509.aspx)
* [Code Samples for Windows Mobile](http://msdn.microsoft.com/en-us/library/bb158662.aspx)
* [How do I Build a .NET Compact Framework 3.5 Application for Windows Mobile 6 Devices](http://msdn.microsoft.com/en-us/windowsmobile/bb975138.aspx)
* [Introduction to Windows Mobile Device Development](http://www.microsoft.com/events/EventDetails.aspx?CMTYSvcSource=MSCOMMedia&Params=~CMTYDataSvcParams%5E~arg%20Name=%22ID%22%20Value=%221032386646%22/%5E~arg%20Name=%22ProviderID%22%20Value=%22A6B43178-497C-4225-BA42-DF595171F04C%22/%5E~arg%20Name=%22lang%252)
* [What’s new for Developers in Windows Mobile 6](http://msdn.microsoft.com/en-us/library/bb278115.aspx)