# Windows Media Center Background Application Project Template

This project template includes the files necessary to create a simple Windows Media Center Background Application using managed code. The template produces a dialog box when run automatically after Windows Media Center launches.

## Getting Started

The following steps are unique to each application and must be performed before the application will run within Windows Media Center. After these steps are completed use the DevInstall.cmd script from a command prompt with administrator privileges to perform a development install of the application.

### Create a strong name key file and add to the project assembly

1. Select the project in the **Solution Explorer pane**.
2. Select **View** > **Property Pages** from the menu.
3. In the **Properties window**:
   1. Select the **Signing** tab.
   2. Check the box labeled **Sign the assembly**.
   3. Click on the **Choose a strong name key file** drop-down list and select **<New...>**.
   4. In the **Create Strong Name Key dialog**:
      1. Enter a key file name
      2. Optionally provide a password for the key file.
      3. Click **OK**.
4. Select **File > Save All** from the menu.
5. Select **Build > Build Solution** from the menu to build the project assembly with the strong name key file.

### Enter the strong name key in the registration file

**Note**: Steps 1 & 2 in this section can be accomplished by running [Windows Media Center SDK Install Path]\Tools\AddGetPublicKeyExternalTool.reg.

1. Select **Tools** > **External Tools** from the menu.
2. In the **External Tools dialog**:
   1. If there are already existing tools click **Add** to create a new one. Otherwise, if this is the first external tool **[New Tool 1]** is already selected
   2. Enter **Get Public Key** in the **Title** text box.
   3. Enter the following in the **Command** text box: **C:\Program Files\Microsoft SDKs\Windows\v6.0A\bin\sn.exe**
   4. Type the following in the **Arguments** text box: **-Tp "$(TargetPath)"**
   5. Uncheck all options except **Use Output window**.
   6. Click the **OK** button.
3. Select **Tools > Get Public Key** from the menu.
4. Copy the public key token value from the output window.
5. Open the **Registration.xml** file for editing with a double-click in the **Solution Explorer pane**.
6. Replace **insert\_public\_key\_token\_here** with the public key token reported by the Get Public Key tool.
7. Select **File > Save All** from the menu.
8. Select **Build > Rebuild Solution** from the menu.

## Finalizing and Deploying the Application

This project template creates several placeholders which should be replaced as follows – most are variables in the source code.

* **AssemblyInfo.cs**: Company and Copyright.
* **DevInstall.cmd**: CompanyName (should match what is contained in Setup.wxs).
* **License.rtf**, **License.txt** and **LicenseAccessible.txt**: All text should be replaced as appropriate.
* **Setup-en-us.wxl**: Property\_ArpHelpLink, Property\_ArpUrlInfoAbout, Property\_ProductLanguage, Property\_CompanyName.
* **Setup.wxs**: MyCompany

## Creating a Windows Installer Package (\*.msi) Setup Program

Also included are the necessary files to create a Windows Installer Package (\*.msi) based setup program using the Windows Installer XML (WiX) toolset. This feature of the application template is not enabled until the WiX toolset is installed and certain properties in the project modified as follows:

### Get Started

1. Download the latest version of the Windows Installer XML (WiX) toolset version 3.0 setup program (wix3.msi) from <http://wix.sourceforge.net/downloadv3.html> and install it on the development computer.
2. Start Microsoft Visual Studio 2005 or Microsoft Visual C# 2005 Express Edition.
3. Open the Windows Media Center application already created, or create a new one using the Windows Media Center Application project template by selecting **File** > **New Project** from the menu.

### Build the Windows Installer Package (\*.msi)

1. Select the project in the **Solution Explorer pane**.
2. Select **View > Property Pages** from the menu.
3. In the **Properties window**:
   1. Select the **Build Events** tab.
   2. Add a new line to the **Post-build event command line** text box.
   3. Type the following on the new line: **"$(ProjectDir)Setup\Build.bat" $(ConfigurationName)**
4. Select **File > Save All** from the menu.
5. Select **Build > Rebuild Solution** from the menu to build the project binaries and the Windows Installer Package.

A Windows Installer Package file named **Setup.msi** is created in the **\bin\release** or **\bin\debug** folder for the project depending on which configuration was built. The files in the \Setup folder for the project can be customized in order to change the behavior of the installer.

## Changing the Pre- and Post-Build Events

Use the following instructions to make changes to the build events in the project template. Note some build events must be enabled by directly modifying the project file (\*.csproj) outside of Visual C# 2005 Express Edition. For more information see the Windows Media Center Step By Step whitepaper included in the SDK.

### To Enable Advanced Build Configurations

Advanced build configurations must first be enabled in Visual C# 2005 Express Edition and Visual Basic 2005 Express Edition using the following steps:

1. Select **Tools > Options** from the menu.
2. In the **Options dialog**:
   1. Check the box in the bottom left corner labeled **Show all settings**.
   2. Select the **Projects and Solutions** item in the tree control.
   3. Check the box labeled **Show advanced build configurations**.
   4. Click **OK**.

### To Modify Pre- and Post-Build Events

1. Select the project in the **Solution Explorer pane**.
2. Select **View > Property Pages** from the menu.
3. In the **Properties window**:
   1. Click on the **Build Events tab**.
   2. Modify the values in the **Pre-build event command line** and **Post-build event command line** text boxes as needed.
   3. Click on the **Debug tab**.
   4. Modify the values in the **Command line arguments** and **Working directory** text boxes as needed.
4. Select **File > Save All** from the menu.

## Adding Code to the Project

Use the following basic steps to add additional code, markup (MCML) and images to the project created by the template.

### To Add Code

1. Select the **Code folder** in the **Solution Explorer pane**.
2. Select **Project > Add New Item** from the menu.
3. In the **Add New Item dialog**:
   1. Select the **Class, Interface or Code File** template.
   2. Enter the name for the file.
   3. Click the **Add** button.