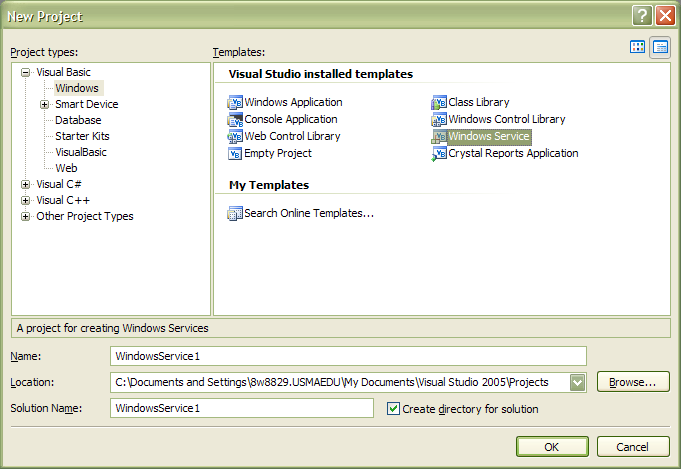
# Creating a Windows Service using VB.NET

The creation of Windows services is no longer limited to C++ gurus. This article will show you how to create, install, start and stop a service with VB.NET. Windows services are helpful when you need an application that constantly monitors something, or that performs an action at a set interval in the background.

**Creating a Windows Service**

Open Visual Studio 2005 Professional Edition and create a new Windows Service project.



**Adding a Continuous Action**

You can perform an action at a set interval by using the Timer control. Please note that placing a Timer control on the interface from the toolbox will not work. We will instead create a new thread and place an instance of a timer in it.

Open the ***Service1.vb*** file in code view. The following code creates an instance of a timer with a 30 second interval.

Imports System.Threading

Public Class Service1

Private oTimer As System.Threading.Timer

Protected Overrides Sub OnStart(ByVal args( ) As String)

‘Code to start the service

Dim oCallback As New TimerCallback(AddressOf OnTimedEvent)

oTimer = New System.Threading.Timer(oCallback, Nothing, 30000, 30000)

End Sub

Private Sub OnTimedEvent(ByVal state As Object)

‘Code to execute when timer ticks

End Sub

End Class

**Adding an Installer to the Project**

Open the service1.vb design window, right-click on it and select the ***Add Installer*** option, which will add an installer project (called ProjectInstaller.vb) with two controls -- ServiceProcessInstaller1 and ServiceInstaller1.

Select the ***ServiceInstaller1*** control and open the property window. Change the *ServiceName* property and *DisplayName* property to the name you want to appear in the list of services in the services window.

Select the ***ServiceProcessInstaller1*** control and open the property window. Change the Account property to *LocalSystem* (this needs to be specified as we need to run the service on our local machine).

Build the application to create an executable (application\_name.exe).

**Installing/Uninstalling the Service**

To install our service we need to use the ***InstallUtil*** program, which is a .NET utility to install Windows services.

Open the Visual Studio command prompt, and type the following command:

InstallUtil C:\PathToBuild\MyService\Bin\Myservice.exe

To uninstall the service (before installing a new build), you add the **/u** switch to the command (InstallUtil /u C:\PathToBuild\MyService\Bin\Myservice.exe)

Replace “PathToBuild” with the path to the executable that you just created. If you build in debug or release mode, there may be a “Debug” or “Release” folder under the “Bin” folder.

**Starting/Stopping the Service**

When you install the service using InstallUtil you are running the service but have yet to start it. You should now notice the service under Control Panel > Administrative Tools > Services.

Locate your service and click the Start or Stop button or right-click the service and selected Start or Stop.

You can open the Event Viewer to see any logs created by the service. Refresh (F5) the list to see the latest log entries.

**Making Changes to the Windows Service**

If changes are made in Visual Studio and a new build is available, you need to uninstall the service before reinstalling the new build. These steps should be followed:

1. Stop the service if it is running.
2. Uninstall the service (using the /u switch).
3. Build the project.
4. Install the newly built service.
5. Start the service.

**Tips**

* Stop the service and close the service window before you install/uninstall the service.
* Always uninstall/install if you make any changes to the service application.
* Try avoiding a user interface, inputs and message boxes in the service application.
* Open ProjectInstaller.vb, select the ServiceInstaller1 control, and open the property window. Change the StartType property to automatic if you want to start the service automatically.