Windows Services

A Windows service is a windows application without any GUI. It runs automatically when windows operating system was started and will be automatically shutdown when windows operating system was shutdown. Anti virus programs are examples for windows services. They run in the background of windows without any GUI and whenever a file containing virus was copied to the system, it will find it and remove the virus. In the same way when you connect a pen drive to the system then it will be automatically detected and a drive is created for it and all this is done by a windows service.

To create your own windows service, use “windows service” template in new project dialog box. A windows service is inherited from “ServiceBase” class and is available in the namespace “System.ServiceProcess”. “ServiceBase” class contains virtual methods “OnStart()”,”OnStop()”,”OnPause()”,”OnContinue()” and “OnShutdown()” that can be overridden in your windows service to write the code to be executed when the service was started, stopped, paused, continued and shutdown respectively.

Example : The following example creates a windows service that detect the changes made to a folder with the name “Test” in D drive and writes an entry to a file “MyService.Txt” whenever a change was made to the file. To identify the changes made to a folder in the operating system, “FileSystemWatcher” component available in components tab of the toolbox is used.

1. Create a new project by selecting “Windows Service” template in the new project dialog box and by specifying the name as “MyService” and in the solution explorer rename the file “Service1.cs” to “MyService.Cs”.

2. Open the properties of windows service and set the “DisplayName” property to “MyService”.

3. Place a “filesystemwatcher” on to the windows service and set following properties to it.

Name	: FSW
EnableRaisingEvents	: true
Filter	: *.*
IncludeSubDirectories	: true
Path	: D:\Test

4. Within the code add the reference tp System.IO namespace and then create a filestream and streamwriter in the class of the windows service as follows.

```
FileStream Fs;
  StreamWriter Sw;
```

5. Write the following code in the method “OnStart()” by overriding it.

```
protected override void OnStart(string[] args)
{
      Fs = new FileStream(@"D:\MyService.Txt", FileMode.OpenOrCreate,
                                                  FileAccess.Write);
      Sw = new StreamWriter(Fs);
      Sw.BaseStream.Seek(0, SeekOrigin.End);
      Sw.WriteLine("MyService Started At " + DateTime.Now.ToString());
      Sw.Flush();
}
```

6. Write the following code in the method “OnStop()” by overriding it.

```
protected override void OnStop()
{
      Sw.BaseStream.Seek(0, SeekOrigin.End);
      Sw.WriteLine("MyService Stopped At " + DateTime.Now.ToString());
      Sw.Flush();
      Sw.Close();
      Fs.Close();
}
```

7. Write the following code in “OnPause()” and “OnContinue()” methods by overriding them so that filesystemwatcher will not raise any event when service was paused and again raises events when continued.

```
protected override void OnPause()
{
    Fsw.EnableRaisingEvents = false;
}

protected override void OnContinue()
{
    Fsw.EnableRaisingEvents = true;
}
```

8. Write the following code in “Created” event of the filesystemwatcher that will be raised whenever a new file/folder was created in the given path.

```
private void Fsw_Created(object sender, FileSystemEventArgs e)
{
      Sw.BaseStream.Seek(0, SeekOrigin.End);
      Sw.WriteLine("A New File/Folder " + e.FullPath + " Was Created At " +
                                                DateTime.Now.ToString());
      Sw.Flush();
}
```

9. Write the following code in “Renamed” event of the filesystemwatcher that will be raised whenever a file/folder was renamed in the given path.

```
private void Fsw_Renamed(object sender, RenamedEventArgs e)
{
      Sw.BaseStream.Seek(0, SeekOrigin.End);
      Sw.WriteLine("A File/Folder " + e.OldFullPath + " Was Renamed as " +
                        e.FullPath + " At " + DateTime.Now.ToString());
      Sw.Flush();
}
```

10. Write the following code in “Deleted” event of the filesystemwatcher that will be raised whenever a file/folder was Deleted in the given path.

```
private void Fsw_Deleted(object sender, FileSystemEventArgs e)
{
      Sw.BaseStream.Seek(0, SeekOrigin.End);
      Sw.WriteLine("A File/Folder " + e.FullPath + " Was Deleted At " +
                                          DateTime.Now.ToString());
      Sw.Flush();
}
```


11. Add installer to the project by right clicking on design of the windows service and choosing “Add Installer”. To install a windows service, installer is required. This will add two components to the windows service, “Service Installer” and “Service process Installer”. “Service Installer” is used to install the windows service and write any entries to windows registry and “Service Process Installer” is used to start a particular process in windows service.

12. for the “Service Installer” set following properties.

DisplayName	: MyService
ServiceName	: MyService
StartType	: Automatic

13. for the “Service Process Installer” set the property “Account” to “LocalSystem” to indicate that this service is available for every user in the local system.

14. change the solution configuration to “Release” and build the solution using the shortcut “Ctrl+Shift+B” to generate an exe file for the project.

Installing Windows Service : To install windows service, use the command line tool “**Installutil.exe”** that has the following syntax.

InstallUtil /i pathofexefile

When you want to uninstall the windows service, use installutil command line tool with the following syntax.

InstallUtil /u pathofexefile