



[MyFirstService.zip](#)

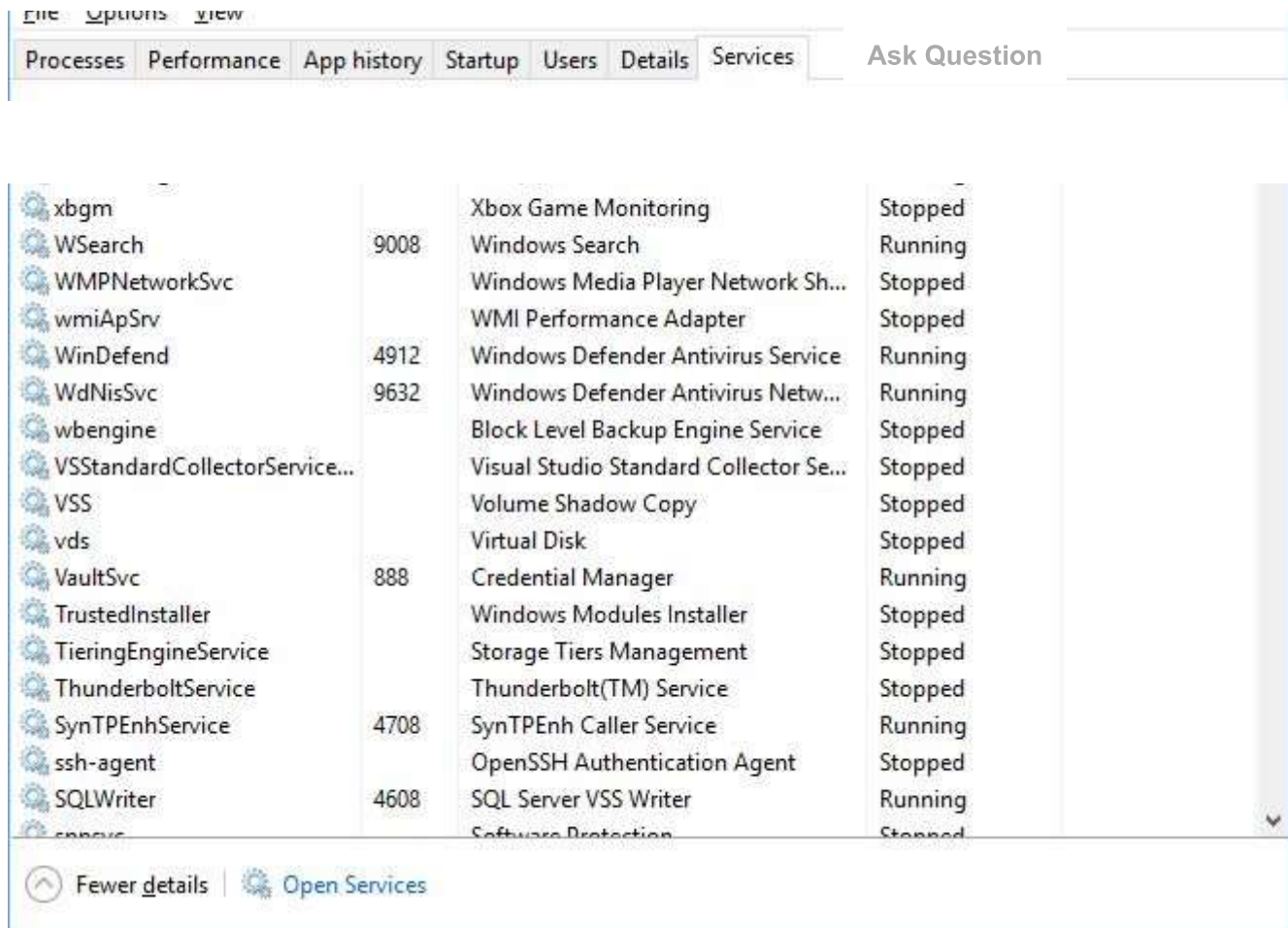
[Download .NET & JAVA Office File API for Free](#)

This article introduces Windows Services in .NET and how to create a Windows Service in C# and .NET using Visual Studio.

What is a Windows Service?

Windows Services are non-UI software applications that run in the background. Windows services usually start when an operating system boots and is scheduled to run in the background to execute some tasks. Windows services can also be started automatically or manually. You can also manually pause, stop and restart Windows services.

Windows service is a computer program that runs in the background to execute some tasks. Some examples of Windows services are auto-update of Windows, check emails, print documents, SQL Server Agent, file and folder scanning and indexing, etc. If you open your Task Manager and click on the Services tab, you will see hundreds of services running on your machine. You can also see the statuses of these services. Some services are running, some have paused, and some have stopped. You can start, stop, and pause a service from here by right click on the service.



You may also find all services running on your machine in the following ways:

- Go to Control Panel and select "Services" inside "Administrative Tools."
- Next, open the Run window (Window + R), type services.msc, and press ENTER.

How to create a Windows service in C#?

Let's create a Windows Service in C# using Visual Studio.

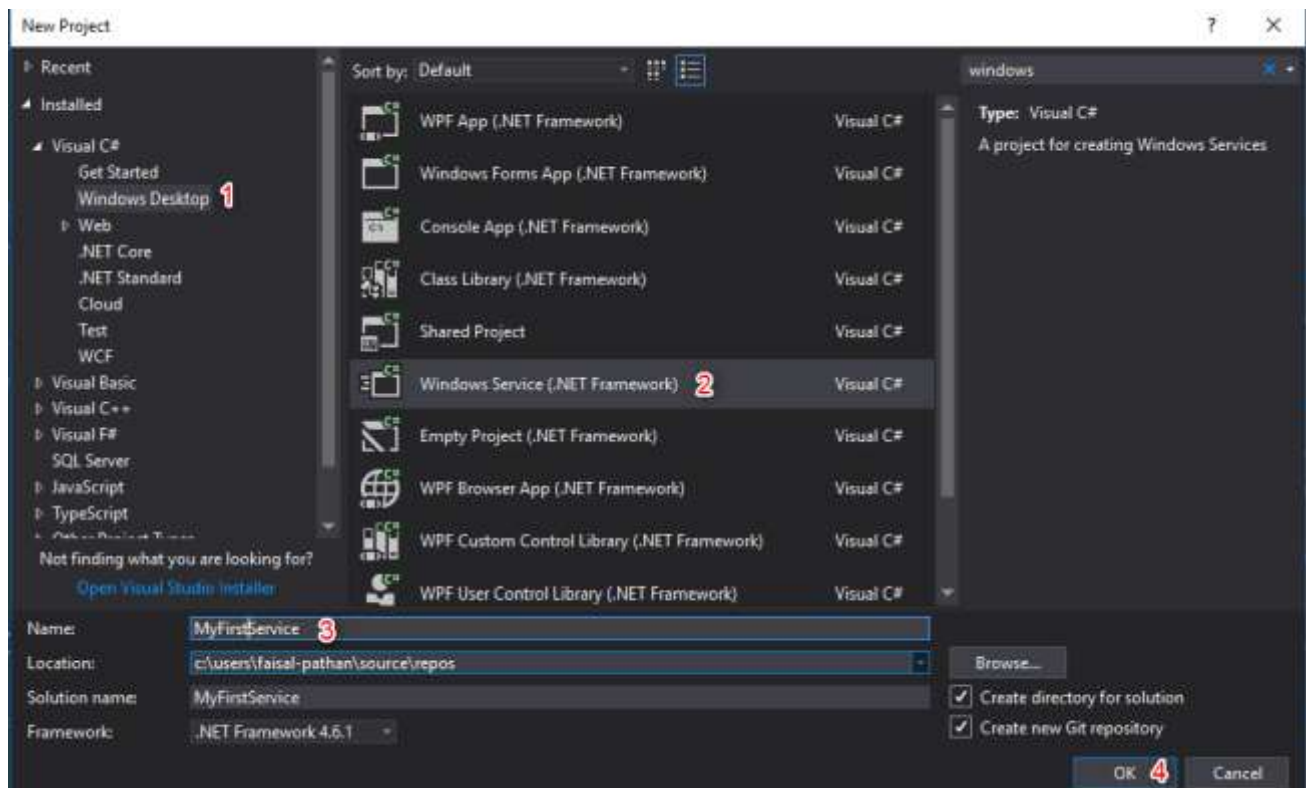
Step 1

Open Visual Studio, click File > New, and select a project. Next, select a new project from the Dialog box, select "Windows Service," and click the OK button.

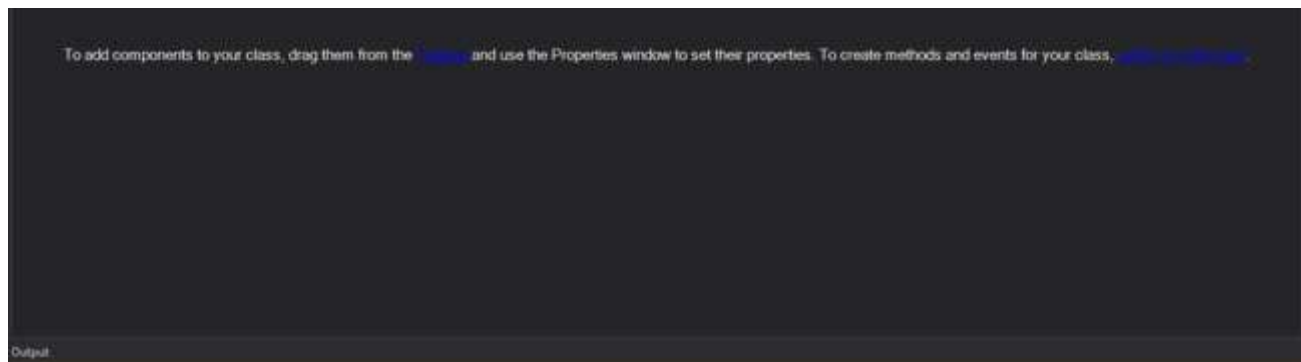


Step 2

Go to Visual C# -> "Windows Desktop" -> "Windows Service," give an appropriate name and then click OK.



Once you click the OK button, the below screen will appear, which is your service.

[Ask Question](#)

Step 3

Right-click on the blank area and select "Add Installer."

How to Add an Installer to a Windows Service

Before you can run a Windows Service, you need to install the Installer, which registers it with the Service Control Manager.



After Adding Installer, ProjectInstaller will add to your project, and ProjectInstakker.cs file will be open. Don't forget to save everything (by pressing the ctrl + shift + s key)



Solution Explore looks like this:



Step 4

Right-click on the blank area and select "View Code"



Step 5

It has Constructor, which contains the InitializeComponent method:

interface objects dragged on the running service and provide
Designer.

.....



Step 6

Select the InitializeComponent method and press the F12 key to go definition.



Step 7

Now add the below line:

```
this.serviceProcessInstaller1.Account = System.ServiceProcess.ServiceAccount.LocalSystem;
```

You also can add a description and display the service name (optionally).

```
1 this.serviceInstaller1.Description = "My First Service demo";  
2 this.serviceInstaller1.DisplayName = "MyFirstService.Demo";
```



Step 8

In this step, we will implement a timer and code to call the service at a given time. Then, we will create a text file and write the current time in the text file using the service.



Service1.cs class

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Diagnostics;  
using System.IO;  
using System.Linq;  
using System.ServiceProcess;  
using System.Text;
```

```
namespace MyFirstService {
```

[Ask Question](#)

```
    public partial class Service1 : ServiceBase {
```

```
        public Service1() {
```

```
            InitializeComponent();
```

```
        }
```

```
        protected override void OnStart(string[] args) {
```

```
            WriteToFile("Service is started at " + DateTime.Now);
```

```
            timer.Elapsed += new ElapsedEventHandler(OnElapsedTime);
```

```
            timer.Interval = 5000; //number in miliseconds
```

```
            timer.Enabled = true;
```

```
        }
```

```
        protected override void OnStop() {
```

```
            WriteToFile("Service is stopped at " + DateTime.Now);
```

```
        }
```

```
        private void OnElapsedTime(object source, ElapsedEventArgs e) {
```

```
            WriteToFile("Service is recall at " + DateTime.Now);
```

```
        }
```

```
        public void WriteToFile(string Message) {
```

```
            string path = AppDomain.CurrentDomain.BaseDirectory + "\\Logs";
```

```
            if (!Directory.Exists(path)) {
```

```
                Directory.CreateDirectory(path);
```

```
            }
```

```
            string filepath = AppDomain.CurrentDomain.BaseDirectory + "\\Log
```

```
            if (!File.Exists(filepath)) {
```

```
                // Create a file to write to.
```

```
                using(StreamWriter sw = File.CreateText(filepath)) {
```

```
                    sw.WriteLine(Message);
```

```
                }
```

```
            } else {
```

```
                using(StreamWriter sw = File.AppendText(filepath)) {
```

```
                    sw.WriteLine(Message);
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

... and write our message.

Step 9. Rebuild your application.



Step 10

Search "Command Prompt" and run as administrator.



Step 11

Fire the below command in the command prompt and press ENTER.

```
cd C:\Windows\Microsoft.NET\Framework\v4.0.30319
```



Step 12

Now Go to your project source folder > bin > Debug and copy the full path of your Windows Service exe file.



Installing a Windows Service

Open the command prompt and fire the below command and press ENTER.

Syntax

InstallUtil.exe + Your copied path + \your service name + .exe

Our path

InstallUtil.exe C:\Users\Faisal-Pathan\source\repos\MyFirstService\MyFirstService\bin\Debug\MyFirstService.exe



Open services by following the below steps:

[Ask Question](#)

3. Find your Service.



You may notice that the Windows service is running.



Check Windows Service Output

The service will create a text file with the following text in it.



The log folder will be created in your bin folder.

Uninstalling a Windows Service

If you want to uninstall your service, fire the below command.

1. Syntax InstallUtil.exe -u + Your copied path + \your service name + .exe
2. Our path InstallUtil.exe -u C:\Users\Faisal-Pathan\source\repos\MyFirstService\MyFirstService\bin\Debug\MyFirstService.exe

Summary

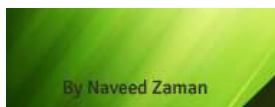
This article taught us how to create a Windows Service and install/Uninstall it using InstallUtil.exe from the command prompt.

I hope you found this tutorial easy to follow and understand.

I also uploaded this project on GitHub; here is the URL
<https://github.com/faisal5170/WindowsService>.

[C#](#)[Windows Services](#)[Windows Services In C#](#)

RECOMMENDED FREE EBOOK


[Download Now!](#)
[Ask Question](#)


SIMILAR ARTICLES

[Create And Deploy Outlook Add-Ins Using SPFx 1.10](#)
[Introduction to Interfacing Win Forms with VS Add-ins](#)
[Birthday Wish Scheduler in C#](#)
[Passing parameters to C# Windows Services](#)
[How To Deploy Outlook Add-ins To Your Organization](#)


Faisal Pathan *TOP 500*

Faisal Pathan is a .NET Project Manager, C# Corner MVP. He has extensive experience with designing and developing enterprise-scale applications. He has good skill of ASP.NET C#, ASP.NET Core, ASP.NET MVC, Angular, Angula... [Read more](#)

<https://www.upwork.com/o/profiles/users/~01eca51226cacbd7ac/>

https://www.youtube.com/channel/UCX6q-4zO_s9wccCWCBC895g?sub_confirmation=1

246

4.4m

2

[View All Comments](#)

76

76



Type your comment here and press Enter Key (Minimum 10 characters)

FEATURED ARTICLES

[Advanced C# Techniques for Better Code Performance](#)

[CSS3 Transforms and Transitions Best Practices](#)

[Getting Started with AlgoKit](#)

Android 14 'Upside Down Cake' - What's New In Latest Releas

[Ask Question](#)[Learn C# 8.0](#)**CHALLENGE YOURSELF****C# Skill****GET CERTIFIED****CSS - Beginner Level**[About Us](#) [Contact Us](#) [Privacy Policy](#) [Terms](#) [Media Kit](#) [Sitemap](#) [Report a Bug](#) [FAQ](#) [Partners](#)[C# Tutorials](#) [Common Interview Questions](#) [Stories](#) [Consultants](#) [Ideas](#) [Certifications](#)

©2023 C# Corner. All contents are copyright of their authors.