Create A Windows Serv e In C#

Faisal Pathan Jun 26 2018 375.1k 42 46

This tutorial explains how to create a Windows Service in C#. You will also learn how to configure and start a Windows Service in .NET.

MyFirstService.zip

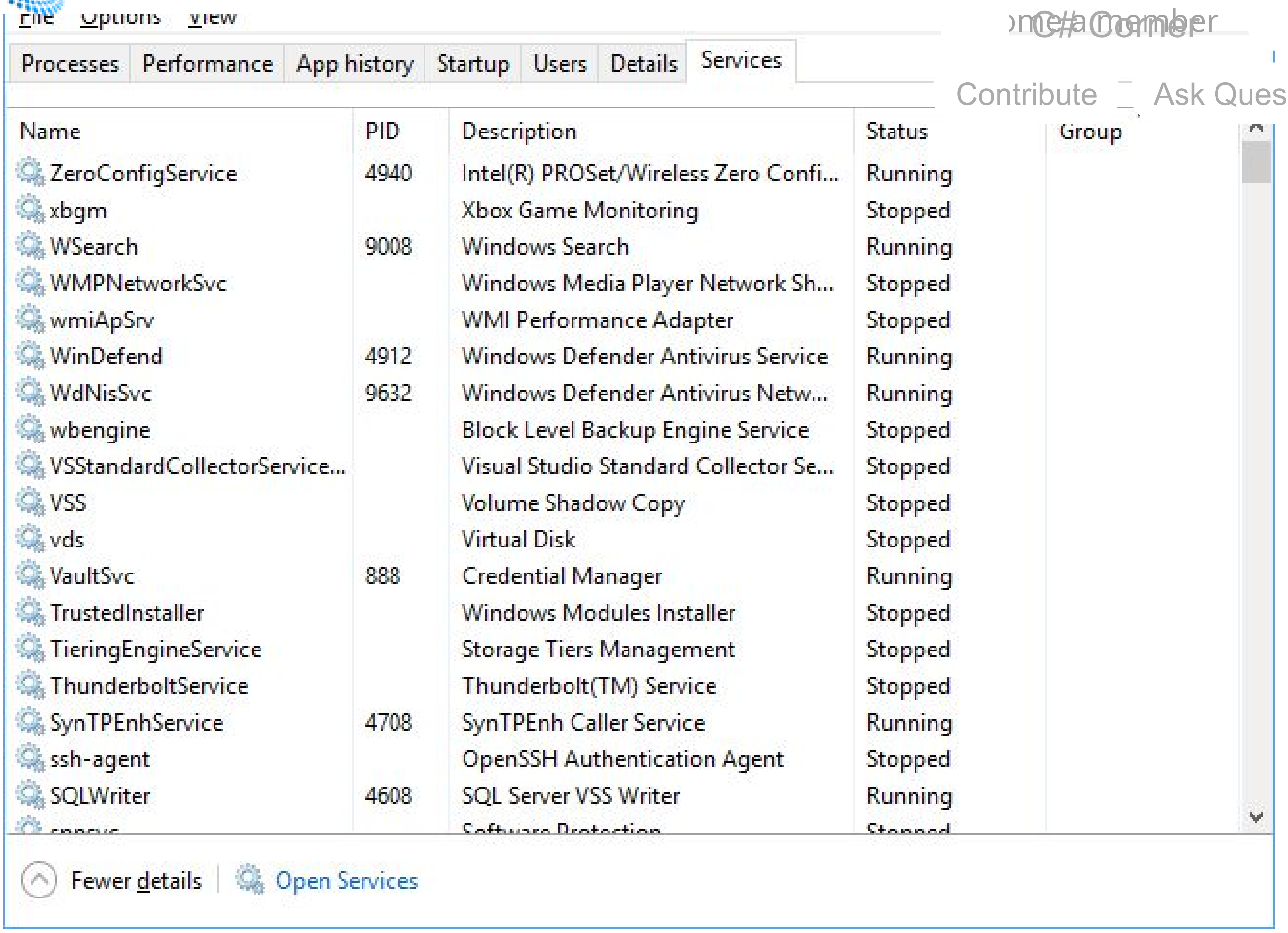
This article is an introduciton to Windows Services in .NET and how to create a Windows Service in C# and .NET using Visual Studio.

What is a Windows Service?

Windows Services are non UI software applications that run in the background. Windows services

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 and so on. If you open your Task Manager and click on 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.



Name	PID	Description	Status	Group
ZeroConfigService	4940	Intel(R) PROSet/Wireless Zero Confi...	Running	
xbgm		Xbox Game Monitoring	Stopped	
WSearch	9008	Windows Search	Running	
WMPNetworkSvc		Windows Media Player Network Sh...	Stopped	
wmiApSrv		WMI Performance Adapter	Stopped	
WinDefend	4912	Windows Defender Antivirus Service	Running	
WdNisSvc	9632	Windows Defender Antivirus Netw...	Running	
wbengine		Block Level Backup Engine Service	Stopped	
VSStandardCollectorService...		Visual Studio Standard Collector Se...	Stopped	
VSS		Volume Shadow Copy	Stopped	
vds		Virtual Disk	Stopped	
VaultSvc	888	Credential Manager	Running	
TrustedInstaller		Windows Modules Installer	Stopped	
TieringEngineService		Storage Tiers Management	Stopped	
ThunderboltService		Thunderbolt(TM) Service	Stopped	
SynTPEnhService	4708	SynTPEnh Caller Service	Running	
ssh-agent		OpenSSH Authentication Agent	Stopped	
SQLWriter	4608	SQL Server VSS Writer	Running	
sppsvc		Software Protection	Stopped	

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

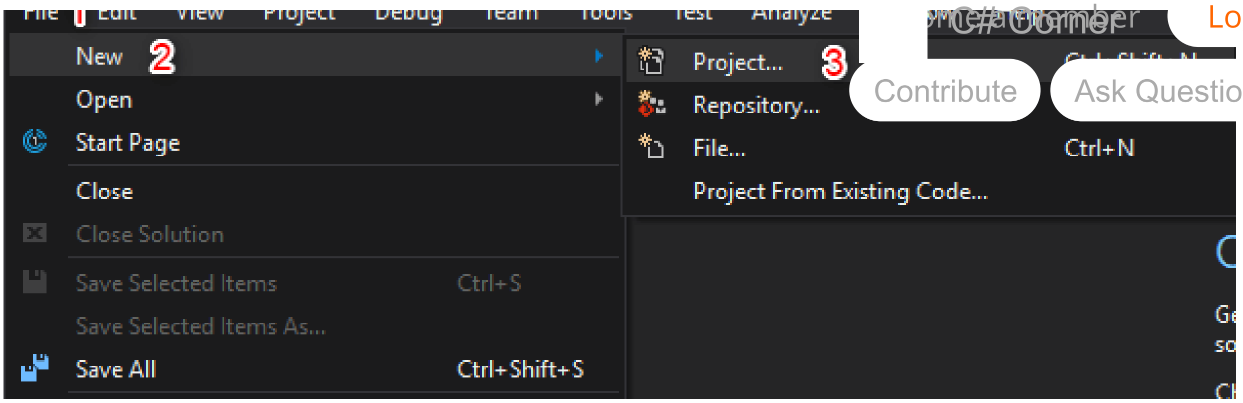
- Go to Control Panel select “Services” inside “Administrative Tools”.

How to create a Windows service in C#?

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

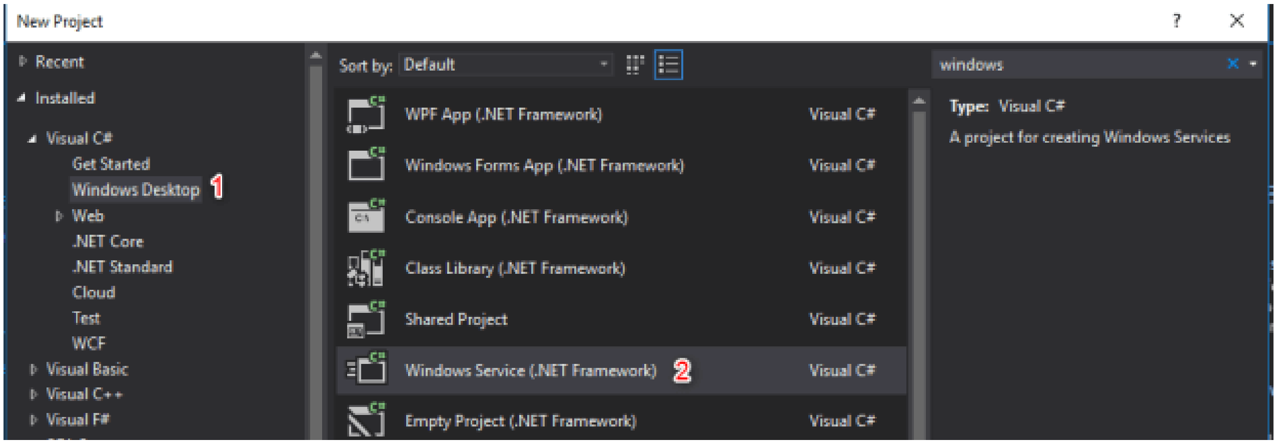
Step 1

Open Visual Studio, go to File > New and select Project. Now select new project from Dialog box and select “Window Service” and click on OK button.


File 1 Edit View Project Debug Team Tools Test Analyze
New 2
Open
Start Page
Close
Close Solution
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Project... 3
Repository...
File... Ctrl+N
Project From Existing Code...

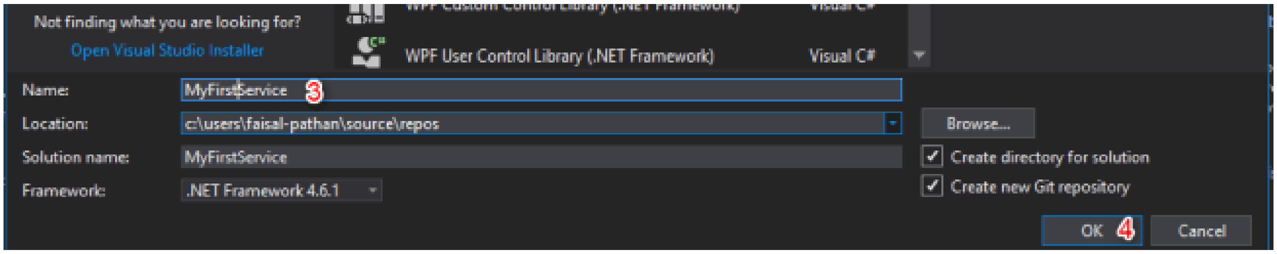
Step 2

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


New Project
Recent
Installed
Visual C#
Get Started
Windows Desktop 1
Web
.NET Core
.NET Standard
Cloud
Test
WCF
Visual Basic
Visual C++
Visual F#
Sort by: Default
windows
WPF App (.NET Framework) Visual C#
Windows Forms App (.NET Framework) Visual C#
Console App (.NET Framework) Visual C#
Class Library (.NET Framework) Visual C#
Shared Project Visual C#
Windows Service (.NET Framework) 2 Visual C#
Empty Project (.NET Framework) Visual C#

Type: Visual C#

A project for creating Windows Services


Not finding what you are looking for?
Open Visual Studio Installer
WPF User Control Library (.NET Framework) Visual C#
Browse...
OK 4
Cancel

Name: MyFirstService 3

Location: c:\users\faisal-pathan\source\repos

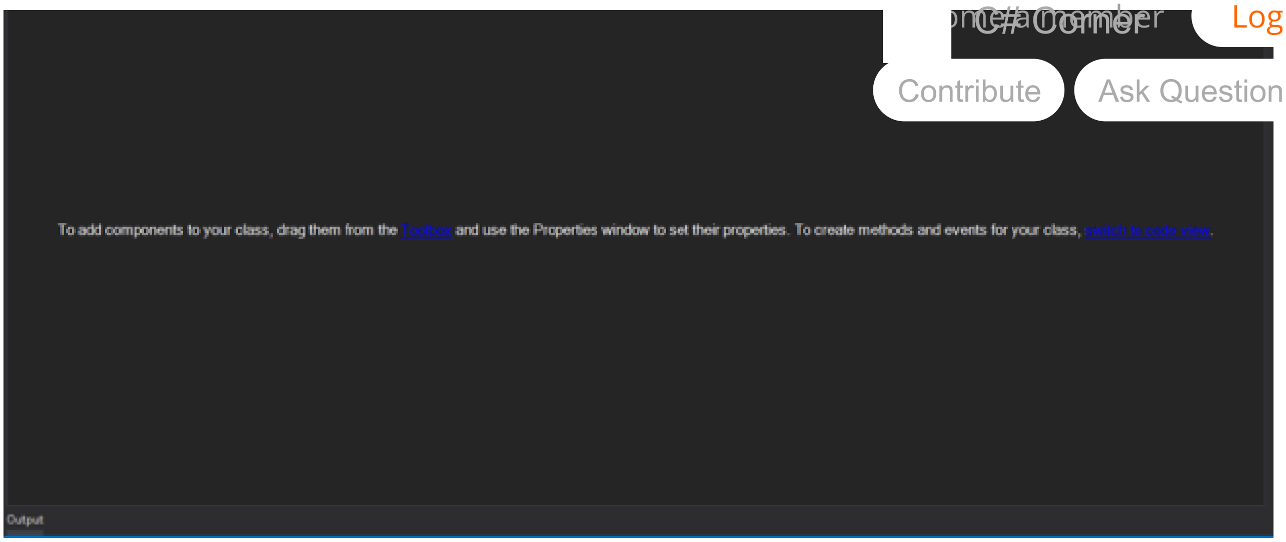
Solution name: MyFirstService

Framework: .NET Framework 4.6.1

☑ Create directory for solution

☑ Create new Git repository

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



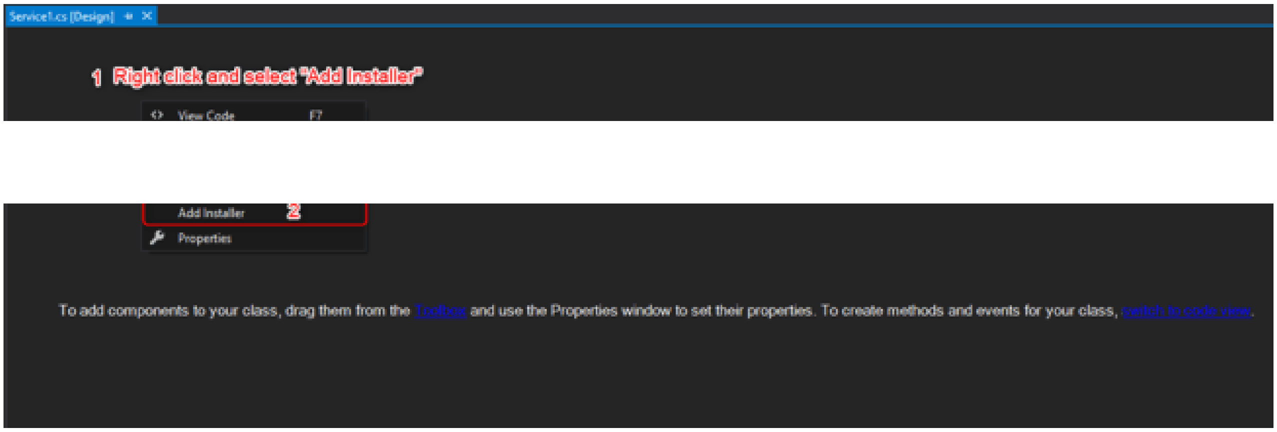
To add components to your class, drag them from the Toolbox and use the Properties window to set their properties. To create methods and events for your class, switch to code view.

Step 3

Right click on the blank area and select “Add Installer”

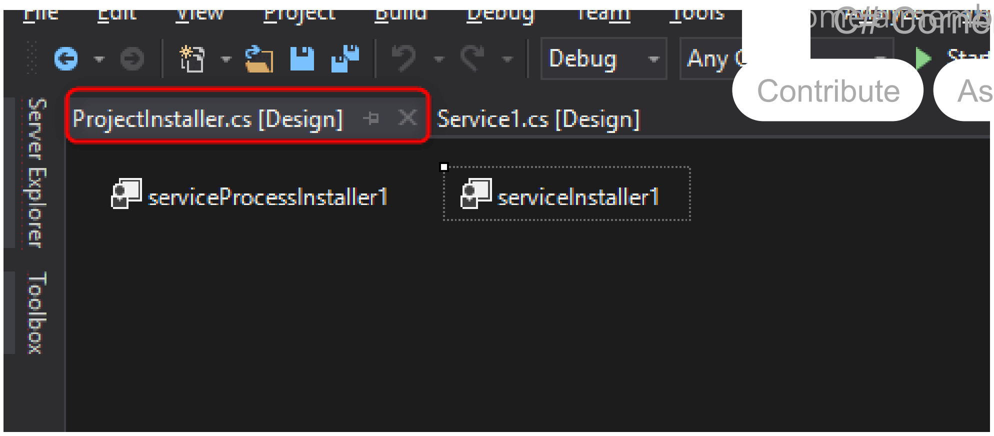
How to Add an Installer to a Windows Service

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

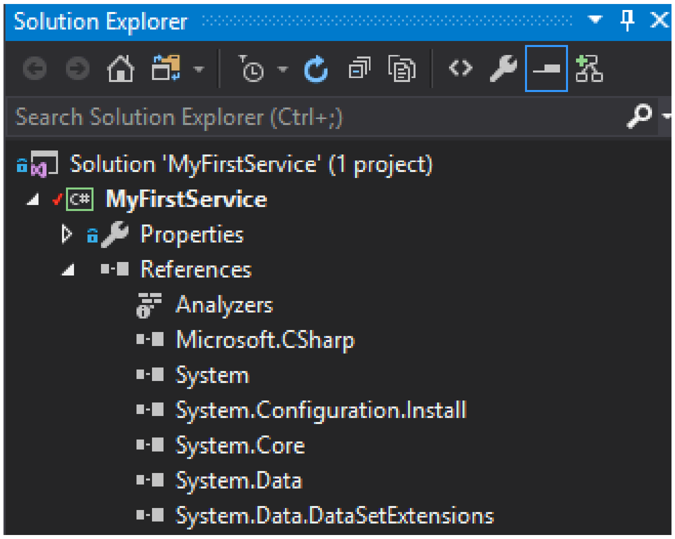


To add components to your class, drag them from the Toolbox and use the Properties window to set their properties. To create methods and events for your class, switch to code view.

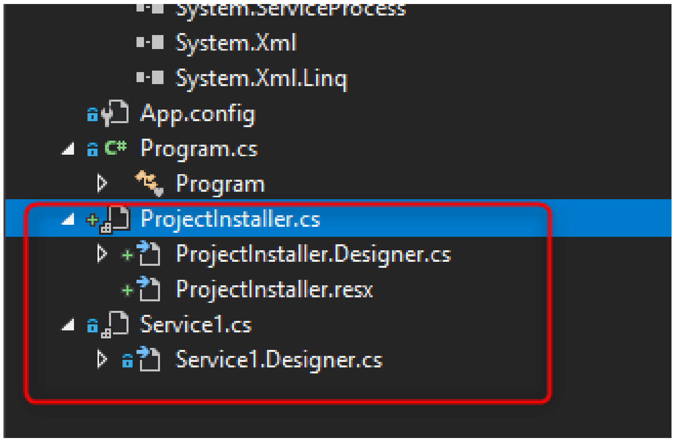
After Adding Installer, ProjectInstaller will add in your project and ProjectInstakker.cs file will be open. Don’t forgot to save everything (by pressing ctrl + shift + s key)



Solution Explore looks like this:



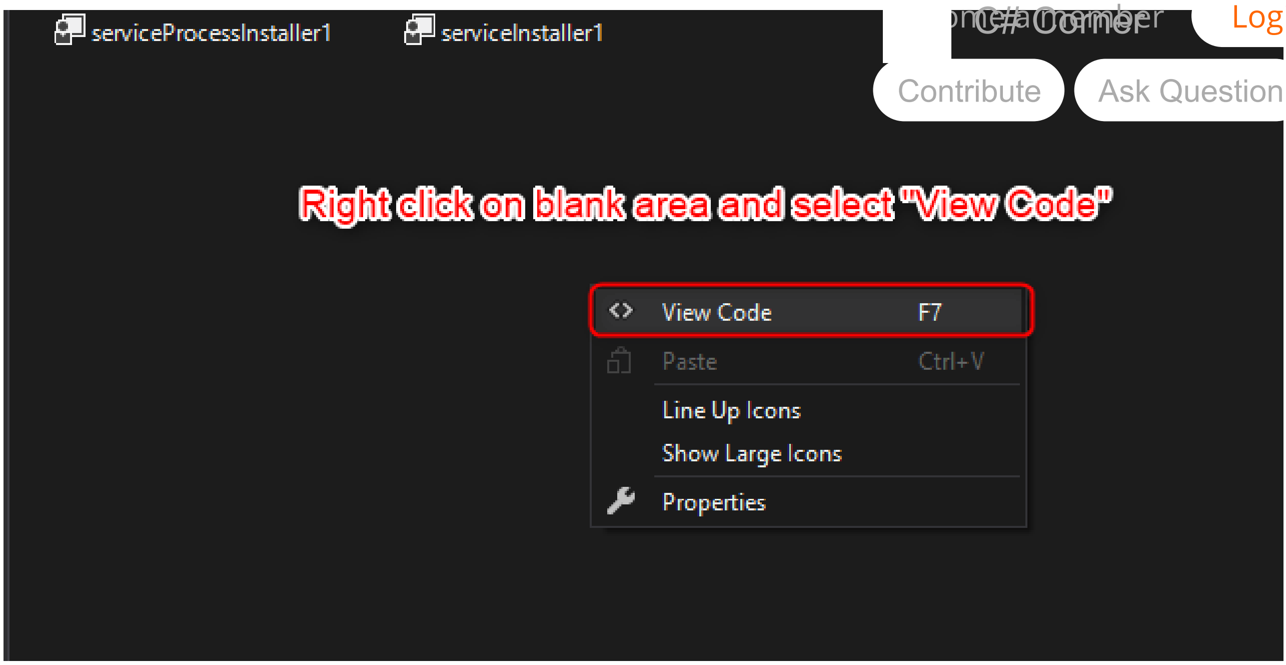
```
Solution 'MyFirstService' (1 project)
  MyFirstService
    Properties
    References
      Analyzers
      Microsoft.CSharp
      System
      System.Configuration.Install
      System.Core
      System.Data
      System.Data.DataSetExtensions
```



```
      System.ServiceProcess
      System.Xml
      System.Xml.Linq
    App.config
    Program.cs
      Program
    ProjectInstaller.cs
      ProjectInstaller.Designer.cs
      ProjectInstaller.resx
    Service1.cs
      Service1.Designer.cs
```

Step 4

Right click on blank area and select “View Code”



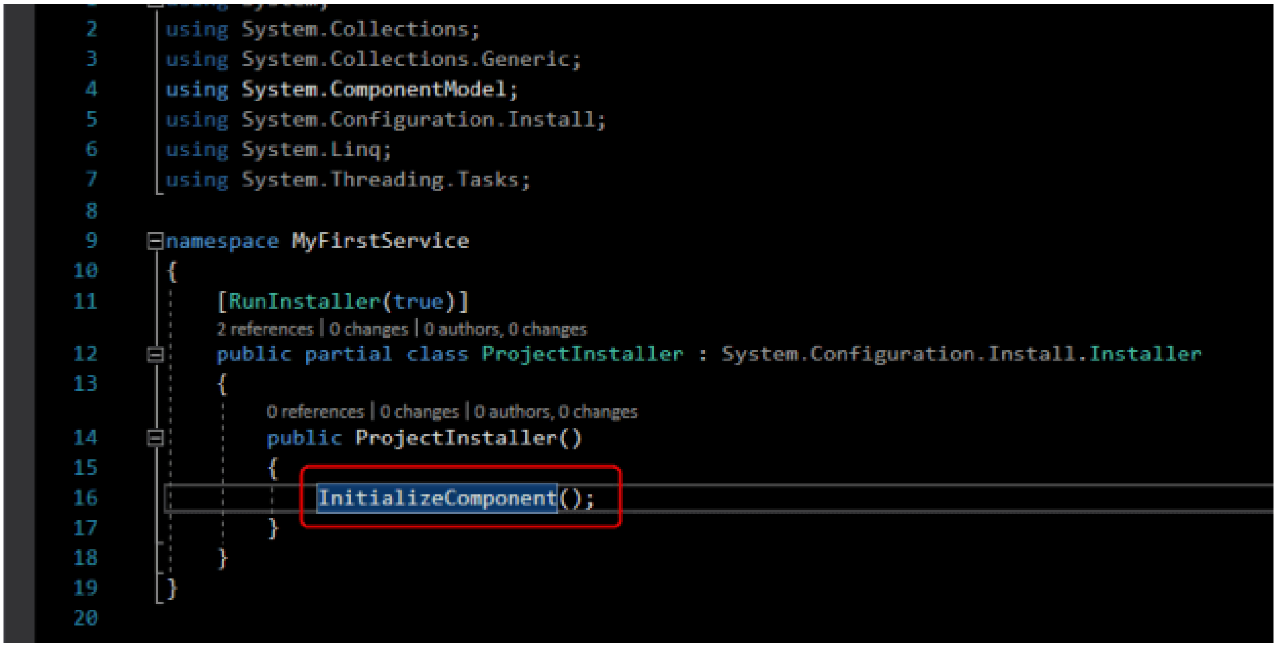
Right click on blank area and select "View Code"

Step 5

Its has Construtor which contains InitializeComponent method:

The InitializeComponent method contains the logic which create and initializes the user interface objects dragged on the form surface and provided the Property Grid of Form Designer.

Very important: Don't ever try to call any method before the call of InitializeComponent method.



```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Configuration.Install;
using System.Linq;
using System.Threading.Tasks;

namespace MyFirstService
{
    [RunInstaller(true)]
    2 references | 0 changes | 0 authors, 0 changes
    public partial class ProjectInstaller : System.Configuration.Install.Installer
    {
        0 references | 0 changes | 0 authors, 0 changes
        public ProjectInstaller()
        {
            InitializeComponent();
        }
    }
}
```

Select InitializeComponent method and press F12 key to go defini

windows service

Step 7

Now add the below line:

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

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

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

windows service

Step 8

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

windows service

Service1.cs class

```
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.ServiceProcess;
using System.Text;
using System.Threading.Tasks;
using System.Timers;
namespace MyFirstService {
    public partial class Service1: ServiceBase {
        Timer timer = new Timer(); // name space(using System.Timers;)
        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 milisecinds
            timer.Enabled = true;
            WriteToFile("Service is stopped at "     teTime.Now);
        }
        private void OnElapsedTime(object source,   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);
                }
            }
        }
    }
}
```

Code explanation - the above code will call service every 5 seconds and create a folder if none exists and write our message.

Step 9. Rebuild your application.

windows service

Step 10

Search “command Prompt” and run as administrator

windows service

Step 11

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

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

windows service

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

windows service

windows service

Installing a Windows Service

Open 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

windows service

Check status of a Windows Service

1. Press Window key + R.
2. Type services.msc
3. Find your Service.

windows service

windows service

You may notice that the Windows service is running.

windows service

Check Windows Service Output

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

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

In this article, we learned 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.

Next Recommended Article

Creating a Windows Service in C#

This article shows how to create a simple Windows Service using C# and Visual Studio 2005.

C# Windows Services Windows Services In C#

Faisal Pathan TOP 500

Faisal Pathan is a .NET Project Manager, C# Corner MVP, Founder of TheCodeHubs.com and senior .NET Core developer at Justlearn. He has extensive experience with designing and developing enterprise-scale applications. He... Read more

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

https://www.c-sharpcorner.com/members/pathan-faisal

216 1.3m 1

View Previous Comments

46 42

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

Nice, how can I use a timer in windows Service with Core 2.2? Timer & Threading Timer doesnt work :(

Isaac Vasquez Nov 25, 2019

1862 2 0 0 0 Reply

Very nice. good one. Its very easy to learn. Thanks Faisal

Deepak Chaudhary Nov 11, 2019

1769 95 3.7k 0 0 Reply

I have created scheduler and its running fine without any error, but now the if i want to control the logic of stop/start of service and configuring the service to run with user defined time interval instead of hard coding the value from code, is that possible

srivatsa bs Oct 07, 2019

1742 122 0 0 1 Reply

Yes, possible

Faisal Pathan Oct 07, 2019

216 9k 1.3m 0

Timothy Gray Oct 03, 2019

1860 4 0 0 1 Reply

This code will not ask to log in while installing

Faisal Pathan Oct 03, 2019

216 9k 1.3m 0

10 minutes to read your post, create project, install and run (fully working) Win Service. Simply... incredible. What else: Thank you a lot :-)

Luciano Db Sep 22, 2019

1859 5 0 1 1 Reply

Thanks for reading and valuable feedback!

Faisal Pathan Sep 23, 2019

216 9k 1.3m 1

Hi, Can I use this to run a console or batch file regardless of user login to a server?

Pius O Sep 13

Presentation was very good.. i liked it.

shavil shaik

1848 16 0

Aug 29, 2019

1 1 Reply

Thanks for feedback

Faisal Pathan

216 9k 1.3m

Aug 29, 2019

0

Nice and simple, awesome work.

Hasitha Chinthaka

1857 7 0

Aug 22, 2019

1 1 Reply

Thanks for the feedback!

Faisal Pathan

216 9k 1.3m

Aug 22, 2019

0

Helpful Explanation

Kaustav basu

1858 6 0

Aug 21, 2019

0 1 Reply

Thanks for feedback

Faisal Pathan

216 9k 1.3m

Aug 21, 2019

0

Thank you very much!

Farid Guseynli

1772 92 0

Aug 04, 2019

1 0 Reply

FEATURED ARTICLES

Continuous Integration and Deployment (CI/CD) of ASP.NET Core Application on Azure Web App using Jenkins - Day Two

LINQ Join Operations Explained

Task Parallel Library 101 Using C#

Combine ADO.NET, EF Core And Dapper In Same Blazor App

Learn Routing In Angular 8

View All ○

TRENDING UP

01 Deep Dive Into C# 9

02 How to Protect an Azure Storage Account using Shared Access Signatures

03 C# 8.x Next

04 SQL Functions Explained

05 Task Parallel Library 101 Using C#

06 Sending an Email to a Distribution List in Microsoft Flow

07 Install ALL Things SQL Server

08 Quick Look At Cognitive Service Content Moderator Web Application

09 Time Management In Organizations

10 Sign-In Page Customization for Specific Branding in Azure

View All ○