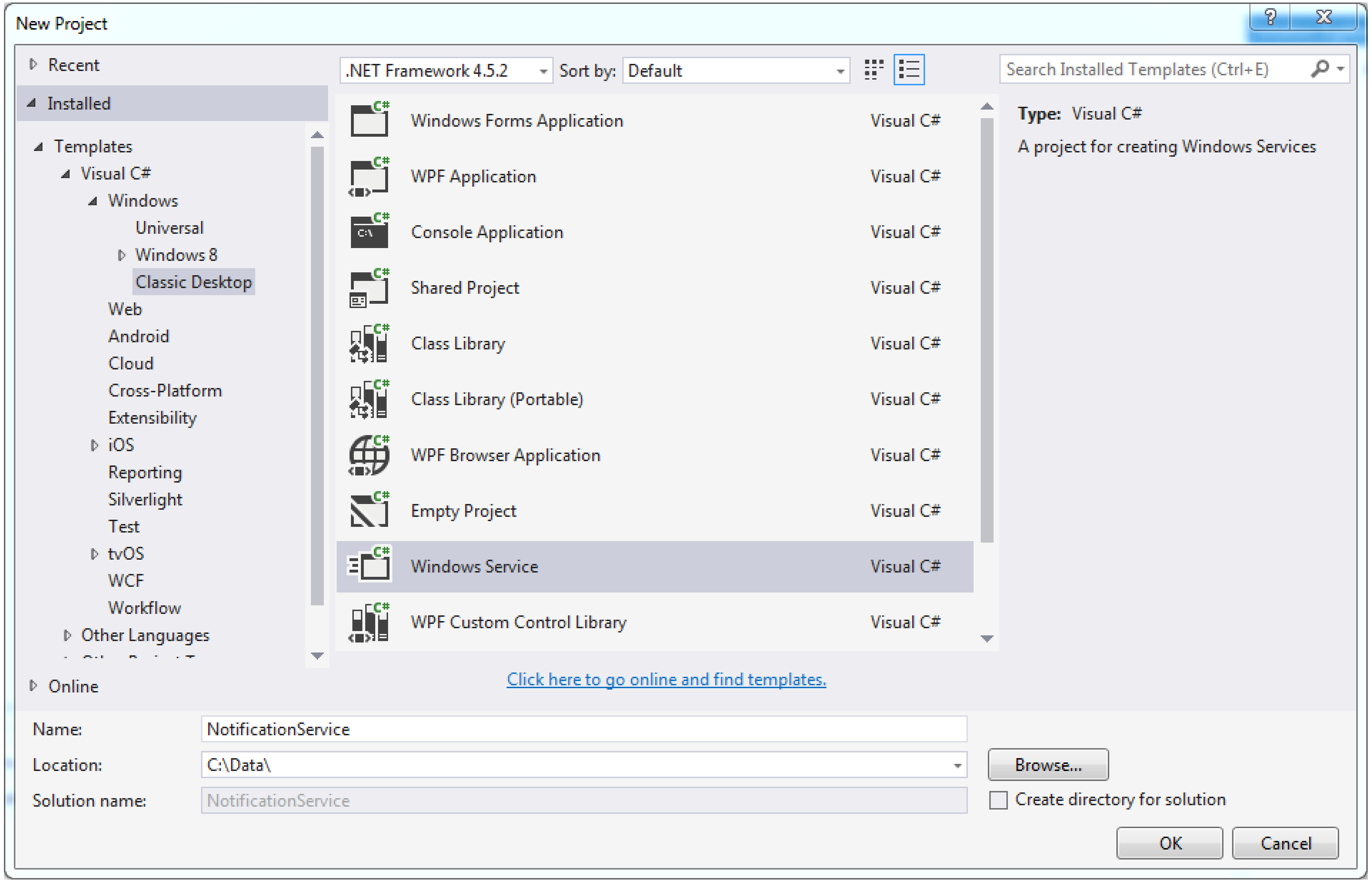
Create a new Windows Service project from the New Project dialog. (C# → Classic Desktop → Windows Service)



New Project

Recent
Installed
Templates
Visual C#
Windows
Universal
Windows 8
Classic Desktop
Web
Android
Cloud
Cross-Platform
Extensibility
iOS
Reporting
Silverlight
Test
tvOS
WCF
Workflow
Other Languages
Online

.NET Framework 4.5.2 | Sort by: Default | Search Installed Templates (Ctrl+E)

Windows Forms Application	Visual C#
WPF Application	Visual C#
Console Application	Visual C#
Shared Project	Visual C#
Class Library	Visual C#
Class Library (Portable)	Visual C#
WPF Browser Application	Visual C#
Empty Project	Visual C#
Windows Service	Visual C#
WPF Custom Control Library	Visual C#

Type: Visual C#

A project for creating Windows Services

Click here to go online and find templates.

Name: NotificationService

Location: C:\Data\

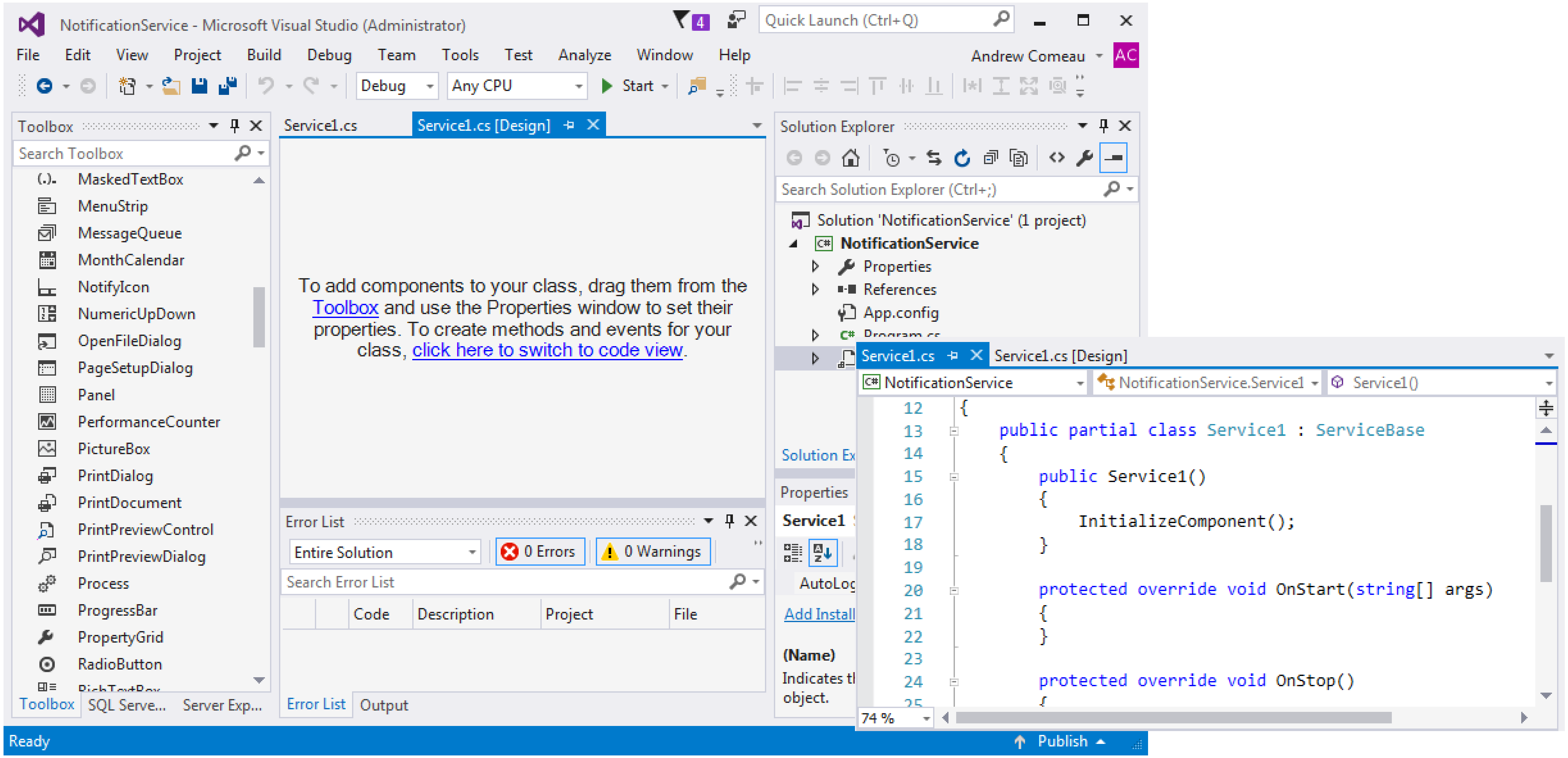
Solution name: NotificationService

Browse...

☐ Create directory for solution

OK | Cancel

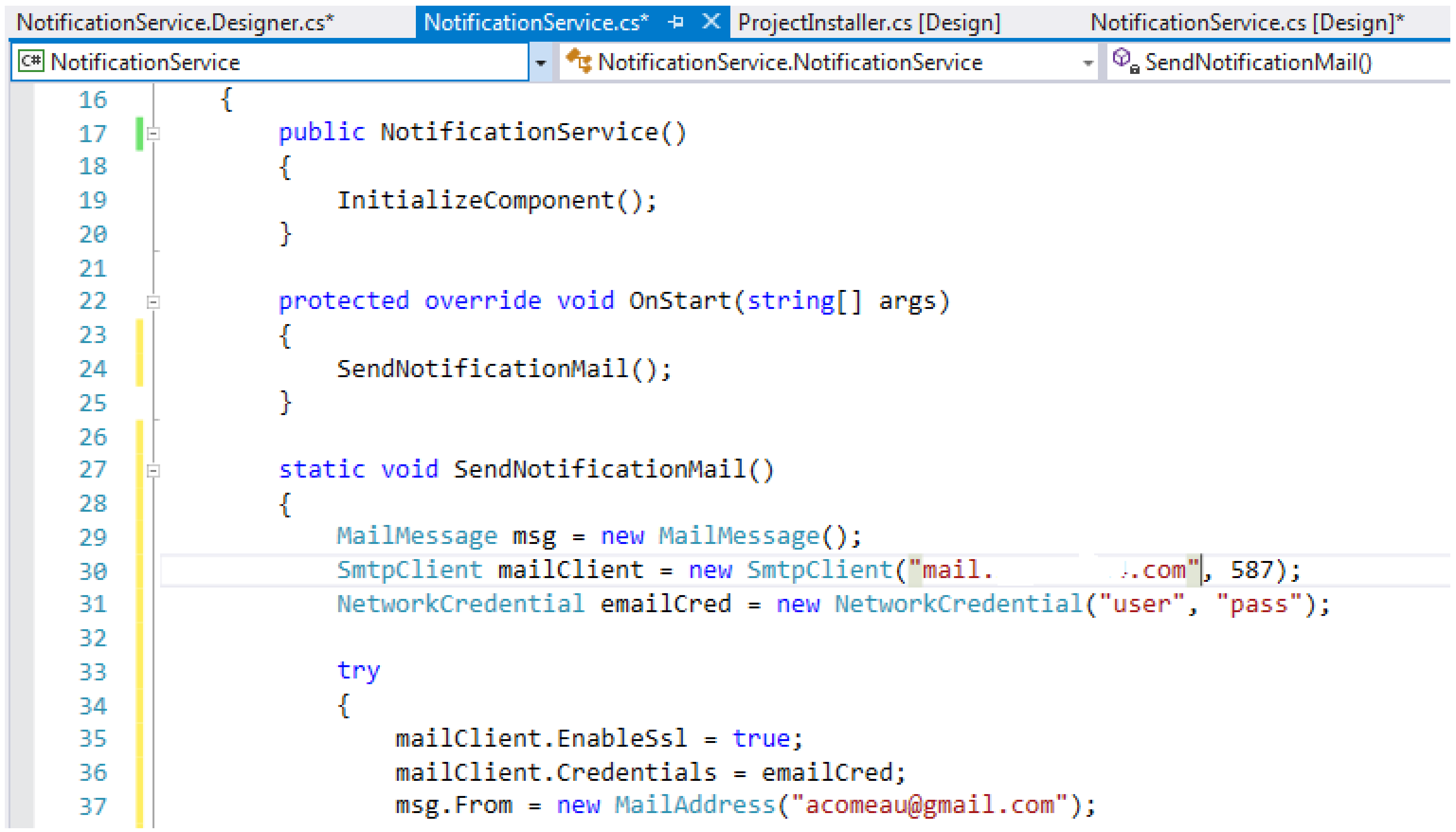
The Service1 class inherits from ServiceBase and overrides its OnStart / OnStop methods. This service can be renamed as needed.



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, click here to switch to code view.

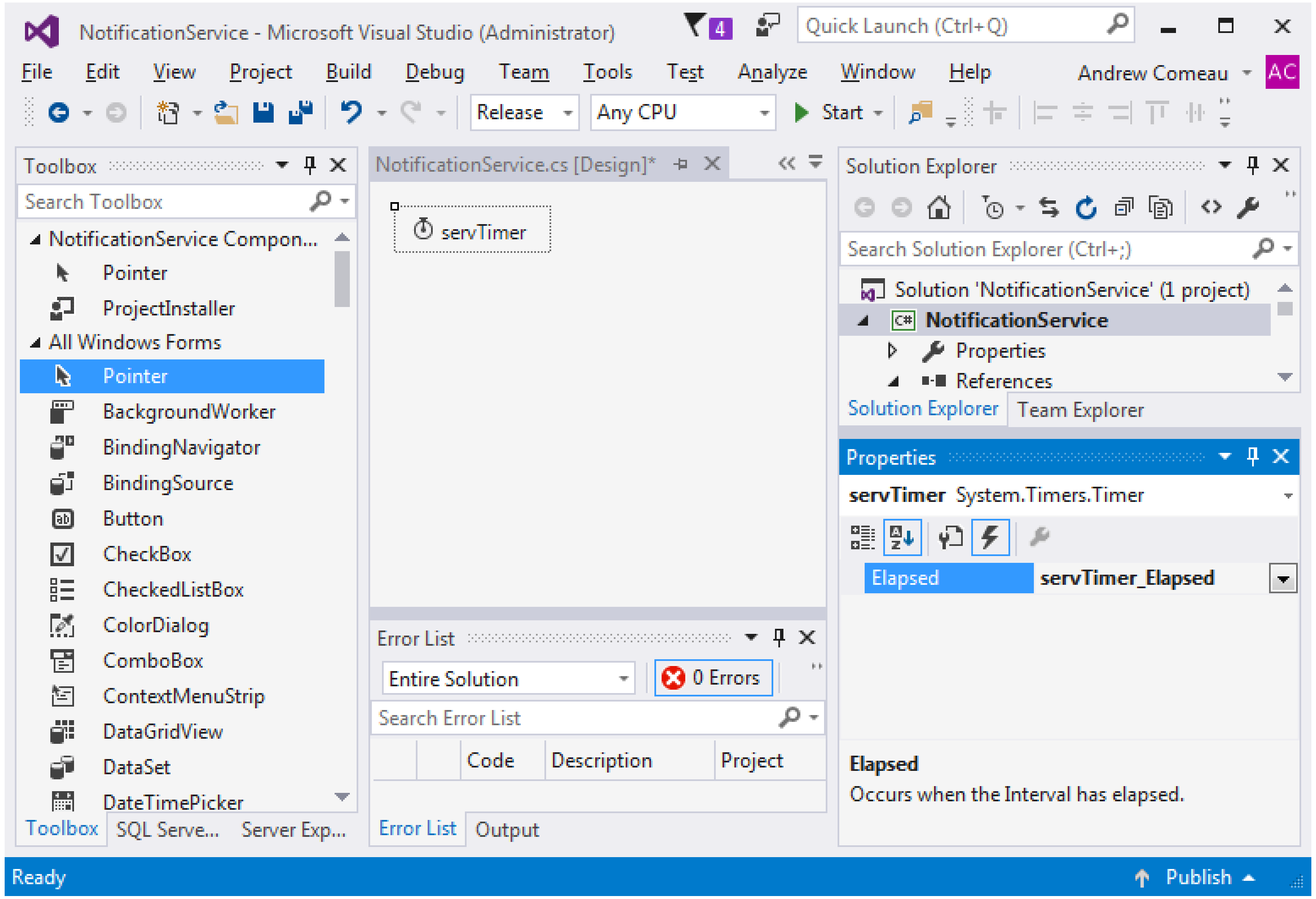
```
{
    public partial class Service1 : ServiceBase
    {
        public Service1()
        {
            InitializeComponent();
        }

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

        protected override void OnStop()
        {
```

You can use the OnStart and OnStop methods to instruct the service on what actions to take at both points.


NotificationService.Designer.cs*
NotificationService.cs*
ProjectInstaller.cs [Design]
NotificationService.cs [Design]*
NotificationService
NotificationService.NotificationService
SendNotificationMail()

```
    {
        public NotificationService()
        {
            InitializeComponent();
        }

        protected override void OnStart(string[] args)
        {
            SendNotificationMail();
        }

        static void SendNotificationMail()
        {
            MailMessage msg = new MailMessage();
            SmtpClient mailClient = new SmtpClient("mail.[redacted].com", 587);
            NetworkCredential emailCred = new NetworkCredential("user", "pass");

            try
            {
                mailClient.EnableSsl = true;
                mailClient.Credentials = emailCred;
                msg.From = new MailAddress("acomeau@gmail.com");
```

You can also use controls like the Timer (System.Timers.Timer) to have the service carry out events periodically.

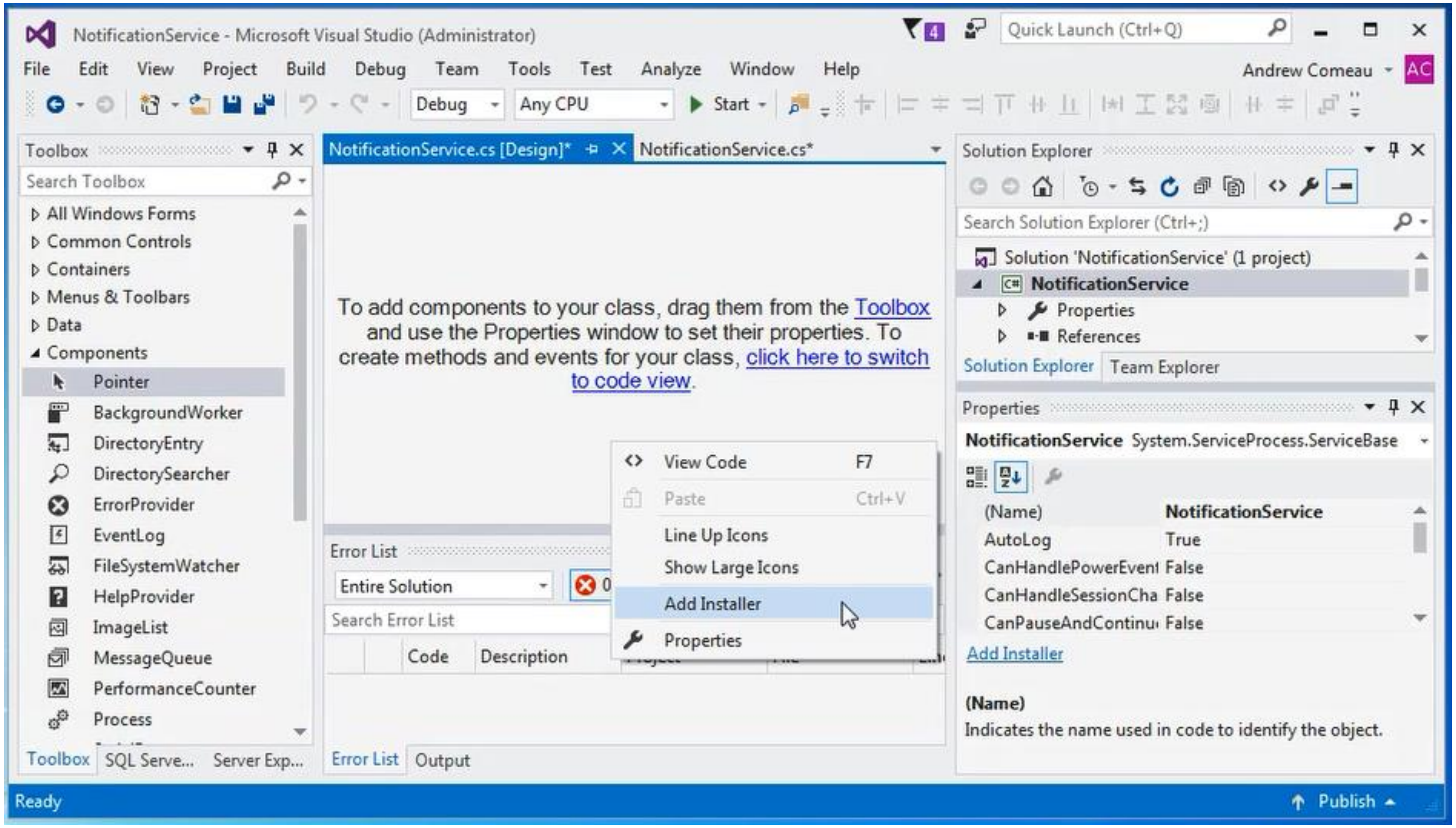


servTimer System.Timers.Timer

Elapsed servTimer_Elapsed

Elapsed
Occurs when the Interval has elapsed.

Once you've added all the necessary functions to the service, you will need to add an Installer class in order to install the service within Windows. From the Designer View context menu, select Add Installer.

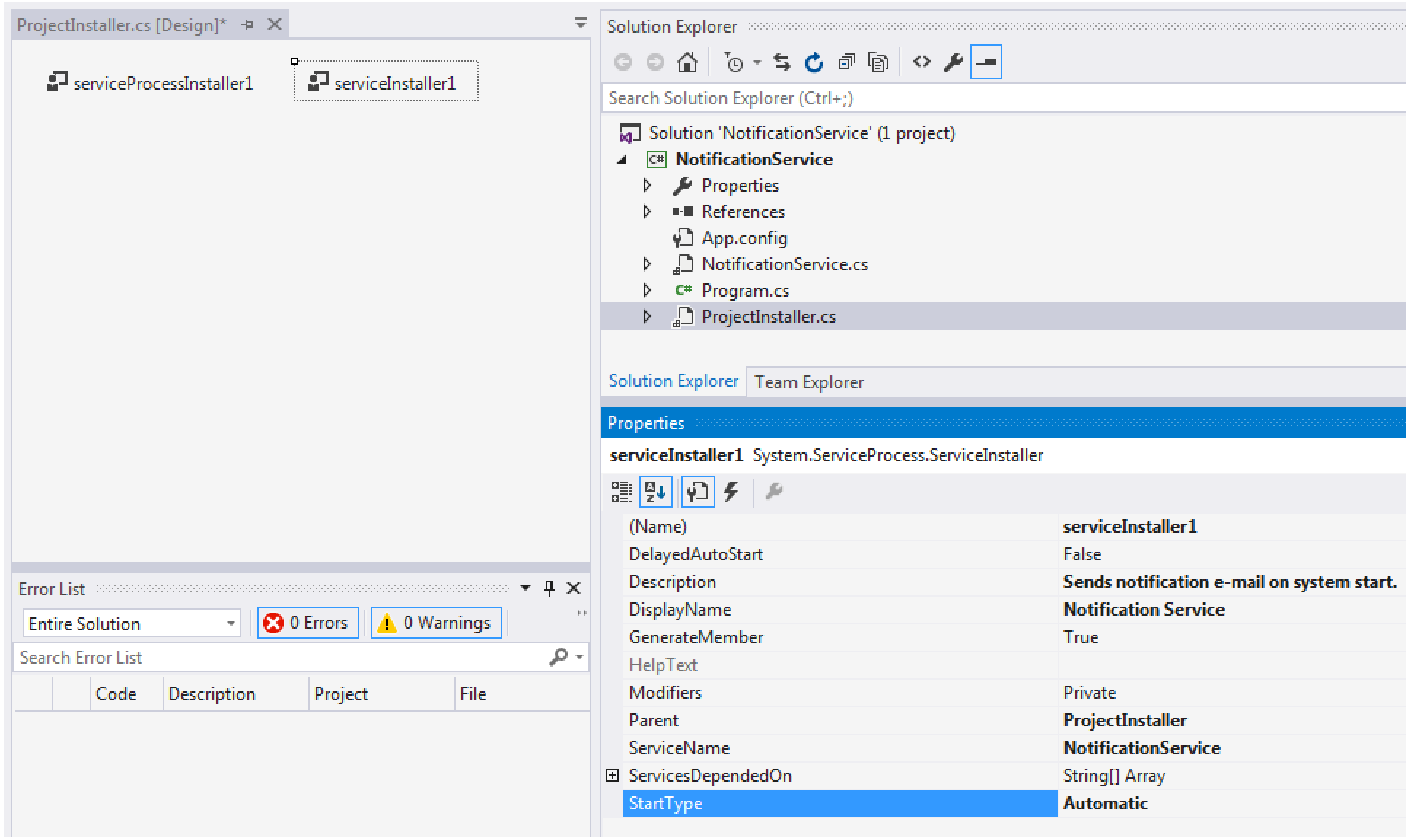


To add components to your class, drag them from the Toolbox and use the Properties window to set their properties. To create methods and events for your class, click here to switch to code view.

(Name)	NotificationService
AutoLog	True
CanHandlePowerEvent	False
CanHandleSessionCha	False
CanPauseAndContinu	False

(Name)
Indicates the name used in code to identify the object.

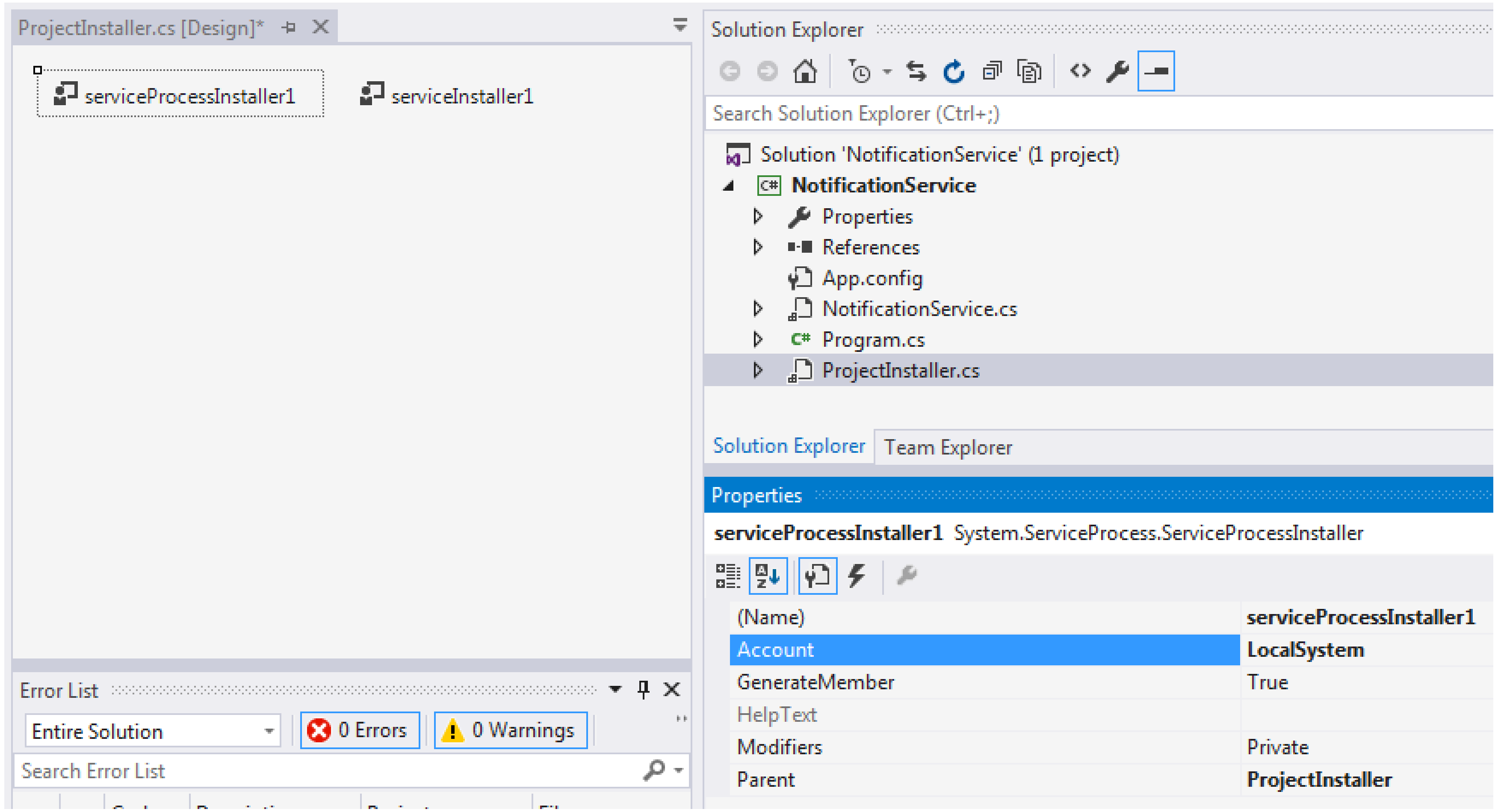
The ProjectInstaller class is added to your project and contains two installers. Select the one called serviceInstaller1. Change the ServiceName, Description, StartType and DisplayName properties. See below for examples.



serviceInstaller1 System.ServiceProcess.ServiceInstaller

(Name)	serviceInstaller1
DelayedAutoStart	False
Description	Sends notification e-mail on system start.
DisplayName	Notification Service
GenerateMember	True
HelpText	
Modifiers	Private
Parent	ProjectInstaller
ServiceName	NotificationService
ServicesDependedOn	String[] Array
StartType	Automatic

Under the serviceProcessInstaller1 installer, change the Account property to LocalSystem or LocalService account. The LocalSystem has greater permissions so it's a good idea to see if your service works with the LocalService account first.

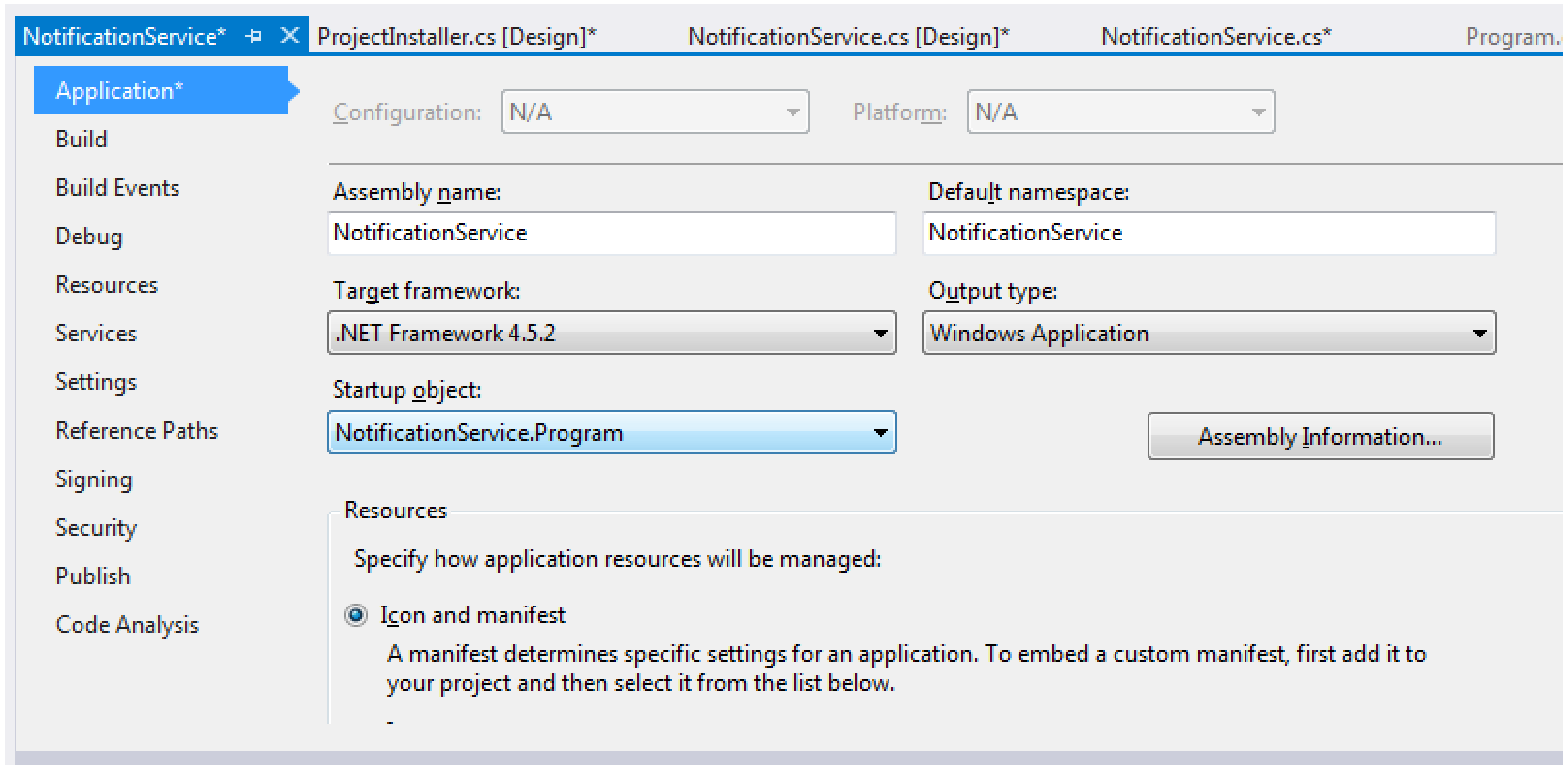


serviceProcessInstaller1 System.ServiceProcess.ServiceProcessInstaller

(Name)	serviceProcessInstaller1
Account	LocalSystem
GenerateMember	True
HelpText	
Modifiers	Private
Parent	ProjectInstaller

In the Properties panel for your project, select the Application table and then set the Startup object to the Program object within your project.

Then choose the Build → Build Solution menu option or press F6 to build the project.


NotificationService* | ProjectInstaller.cs [Design]* | NotificationService.cs [Design]* | NotificationService.cs* | Program.

Application*
Build
Build Events
Debug
Resources
Services
Settings
Reference Paths
Signing
Security
Publish
Code Analysis

Configuration: N/A
Platform: N/A

Assembly name: NotificationService
Default namespace: NotificationService
Target framework: .NET Framework 4.5.2
Output type: Windows Application
Startup object: NotificationService.Program

Assembly Information...

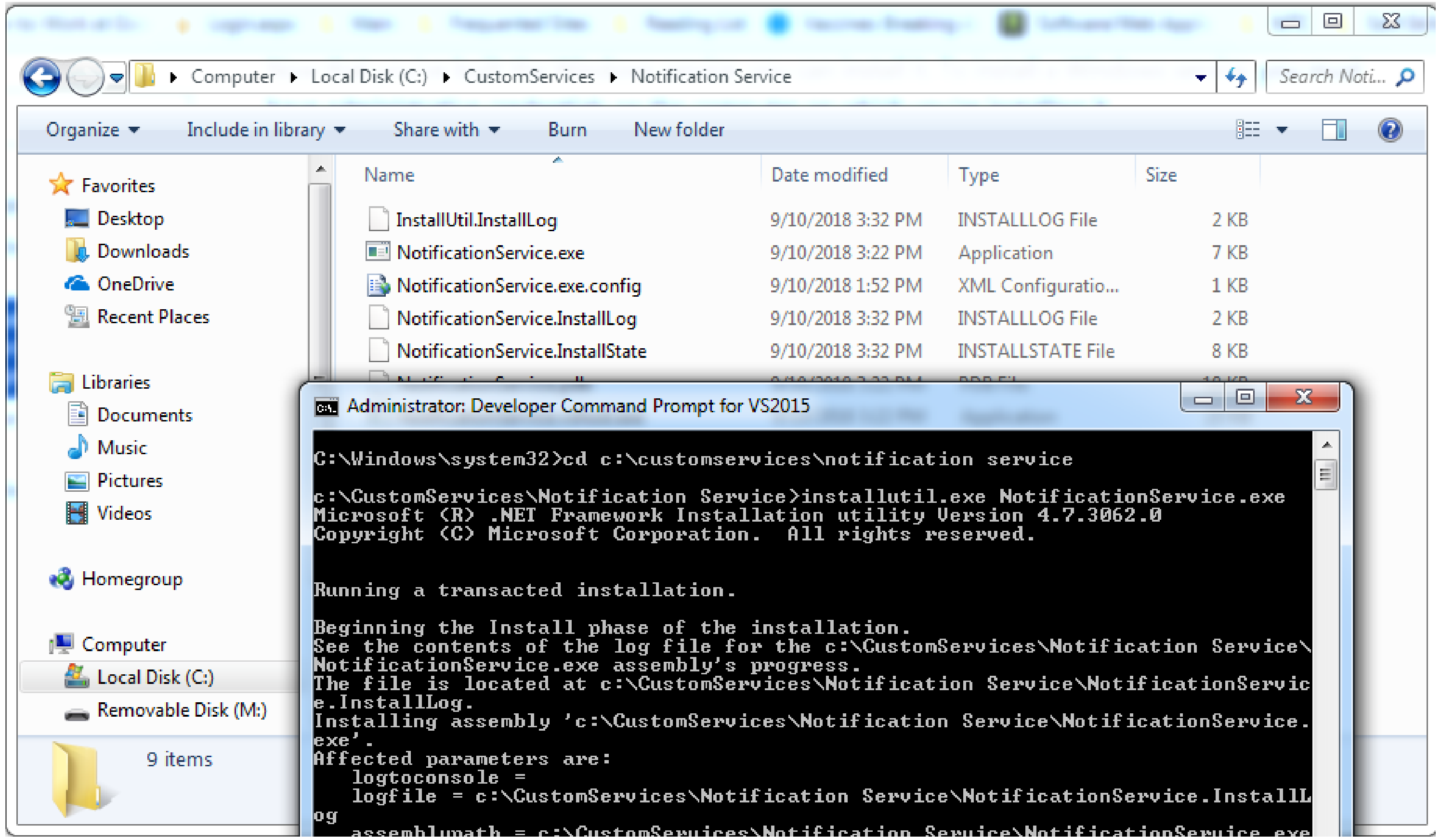
Resources

Specify how application resources will be managed:

(●) Icon and manifest

A manifest determines specific settings for an application. To embed a custom manifest, first add it to your project and then select it from the list below.

When you install your service in Windows, it's best to copy the build files to a deployment directory. Open the Developer command prompt in Administrator mode, navigate to that directory and enter the command **`installutil.exe`** ***`<Your EXE Name>`*** as shown below. The installation process will run and report on its success or failure.



Computer ▸ Local Disk (C:) ▸ CustomServices ▸ Notification Service

Name	Date modified	Type	Size
InstallUtil.InstallLog	9/10/2018 3:32 PM	INSTALLLOG File	2 KB
NotificationService.exe	9/10/2018 3:22 PM	Application	7 KB
NotificationService.exe.config	9/10/2018 1:52 PM	XML Configuratio...	1 KB
NotificationService.InstallLog	9/10/2018 3:32 PM	INSTALLLOG File	2 KB
NotificationService.InstallState	9/10/2018 3:32 PM	INSTALLSTATE File	8 KB

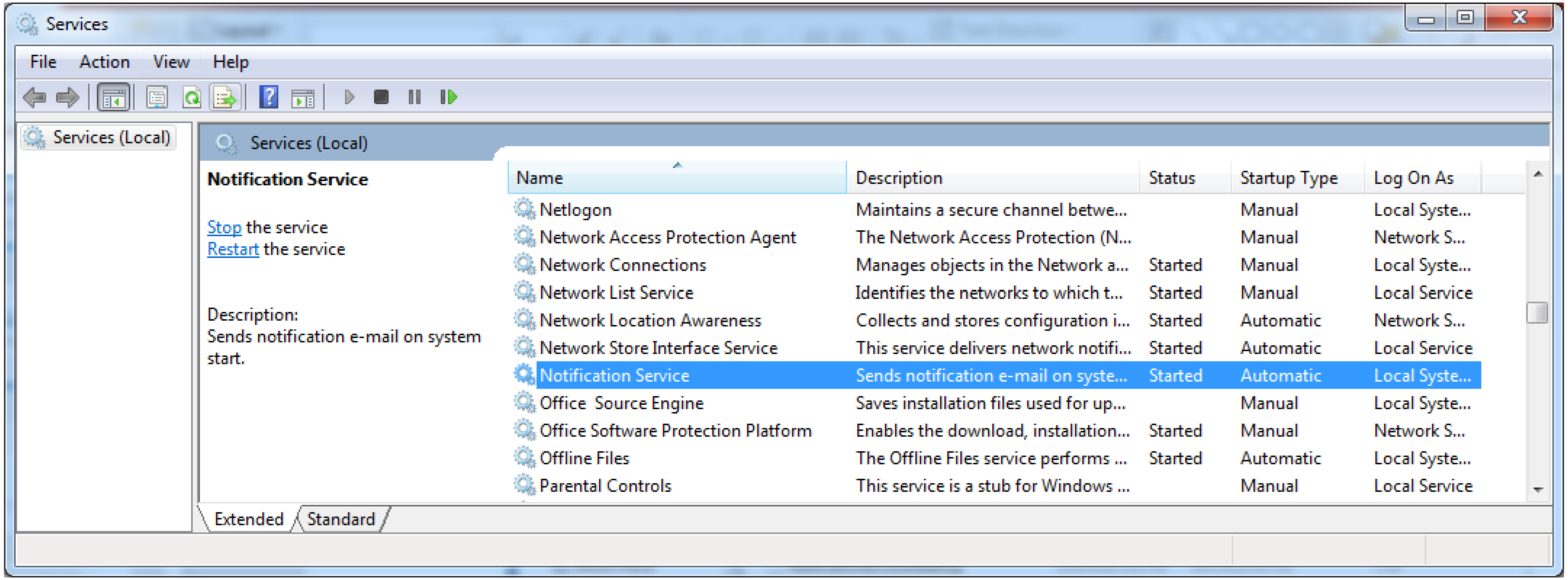
9 items

Administrator: Developer Command Prompt for VS2015

```
C:\Windows\system32>cd c:\customservices\notification service

c:\CustomServices\Notification Service>installutil.exe NotificationService.exe
Microsoft (R) .NET Framework Installation utility Version 4.7.3062.0
Copyright (C) Microsoft Corporation.  All rights reserved.


Running a transacted installation.

Beginning the Install phase of the installation.
See the contents of the log file for the c:\CustomServices\Notification Service\
NotificationService.exe assembly's progress.
The file is located at c:\CustomServices\Notification Service\NotificationServic
e.InstallLog.
Installing assembly 'c:\CustomServices\Notification Service\NotificationService.
exe'.
Affected parameters are:
   logtoconsole =
   logfile = c:\CustomServices\Notification Service\NotificationService.InstallL
og
   assemblypath = c:\CustomServices\Notification Service\NotificationService.exe
```

To uninstall a service, navigate back to the directory and run the command
`installutil.exe -u` ***`<Your EXE Name>`***

If the installation is successful, you will be able to see your service in the Windows Services panel. You can right-click on it and set some of the extra properties as well as starting and stopping it.



Notification Service

Stop the service
Restart the service

Description:
Sends notification e-mail on system start.

Name	Description	Status	Startup Type	Log On As
Netlogon	Maintains a secure channel betwe...		Manual	Local Syste...
Network Access Protection Agent	The Network Access Protection (N...		Manual	Network S...
Network Connections	Manages objects in the Network a...	Started	Manual	Local Syste...
Network List Service	Identifies the networks to which t...	Started	Manual	Local Service
Network Location Awareness	Collects and stores configuration i...	Started	Automatic	Network S...
Network Store Interface Service	This service delivers network notifi...	Started	Automatic	Local Service
Notification Service	Sends notification e-mail on syste...	Started	Automatic	Local Syste...
Office Source Engine	Saves installation files used for up...		Manual	Local Syste...
Office Software Protection Platform	Enables the download, installation...	Started	Manual	Network S...
Offline Files	The Offline Files service performs ...	Started	Automatic	Local Syste...
Parental Controls	This service is a stub for Windows ...		Manual	Local Service

For more details and another example service you can create, see the Microsoft site:
https://docs.microsoft.com/en-us/dotnet/framework/windows-services/walkthrough-creating-a-windows-service-application-in-the-component-designer