**一、开发环境**

操作系统：Windows 10 X64

开发环境：VS2015

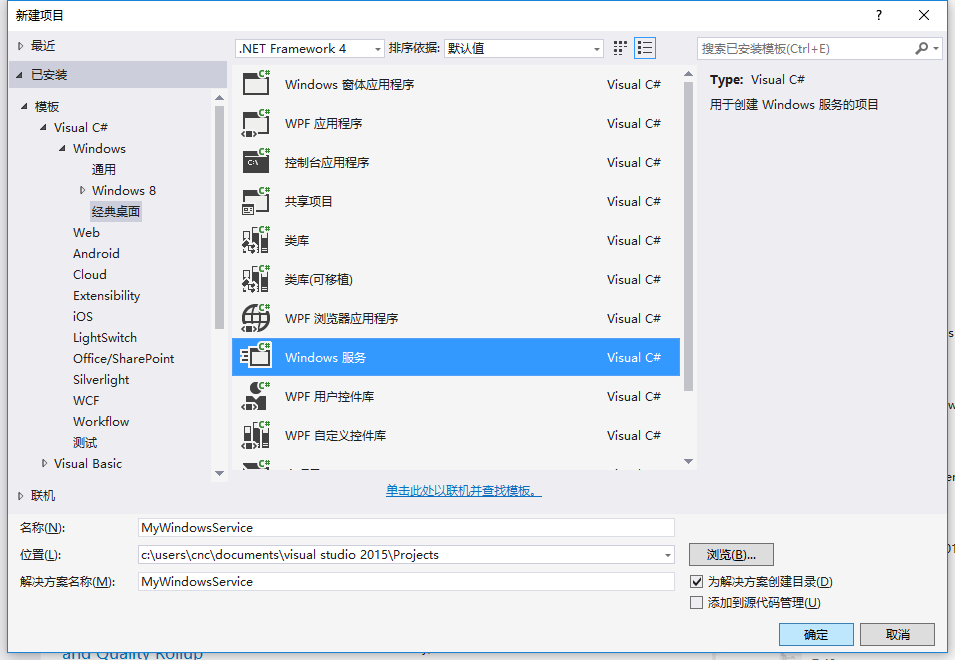
编程语言：C#

.NET版本：.NET Framework 4.0

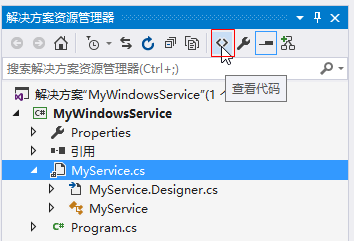
目标平台：X86

**二、创建Windows Service**

1、新建一个Windows Service，并将项目名称改为“MyWindowsService”，如下图所示：



2、在解决方案资源管理器内将Service1.cs改为MyService1.cs后并点击“查看代码”图标按钮进入代码编辑器界面，如下图所示：



3、在代码编辑器内如入以下代码，如下所示：

[IMG_258](https://www.cnblogs.com/mq0036/p/javascript:void(0);)

[IMG_259](https://www.cnblogs.com/mq0036/p/javascript:void(0);)

using System;using System.ServiceProcess;using System.IO;

namespace MyWindowsService

{

public partial class MyService : ServiceBase

{

public MyService()

{

InitializeComponent();

}

string filePath = @"D:\MyServiceLog.txt";

protected override void OnStart(string[] args)

{

using (FileStream stream = new FileStream(filePath,FileMode.Append))

using (StreamWriter writer = new StreamWriter(stream))

{

writer.WriteLine($"{DateTime.Now},服务启动！");

}

}

protected override void OnStop()

{

using (FileStream stream = new FileStream(filePath, FileMode.Append))

using (StreamWriter writer = new StreamWriter(stream))

{

writer.WriteLine($"{DateTime.Now},服务停止！");

}

}

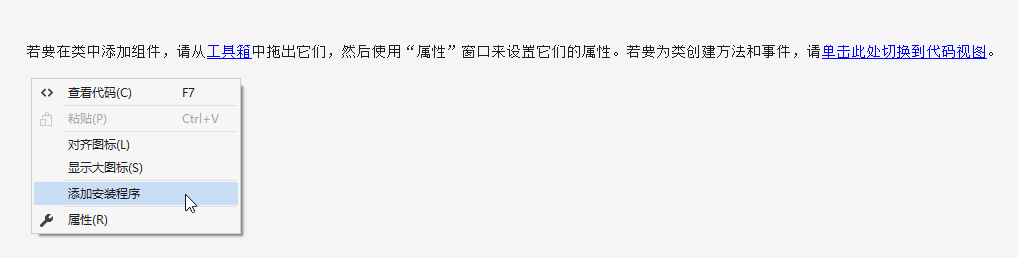
}

}

[IMG_260](https://www.cnblogs.com/mq0036/p/javascript:void(0);)

[IMG_261](https://www.cnblogs.com/mq0036/p/javascript:void(0);)

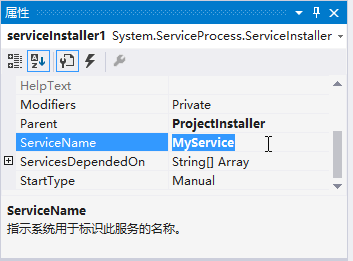
4、双击项目“MyWindowsService”进入“MyService”设计界面，在空白位置右击鼠标弹出上下文菜单，选中“添加安装程序”，如下图所示：

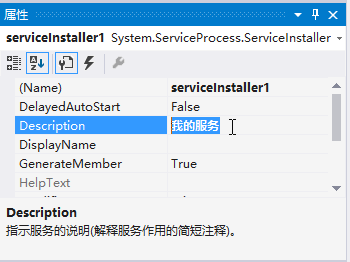


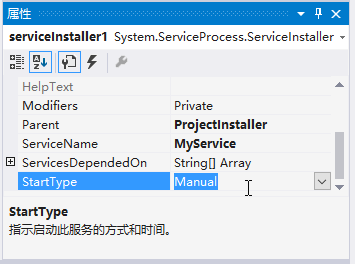
5、此时软件会生成两个组件，分别为“serviceInstaller1”及“serviceProcessInstaller1”，如下图所示：



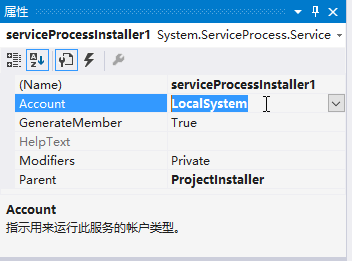
6、点击“serviceInstaller1”，在“属性”窗体将ServiceName改为MyService，Description改为我的服务，StartType保持为Manual，如下图所示：



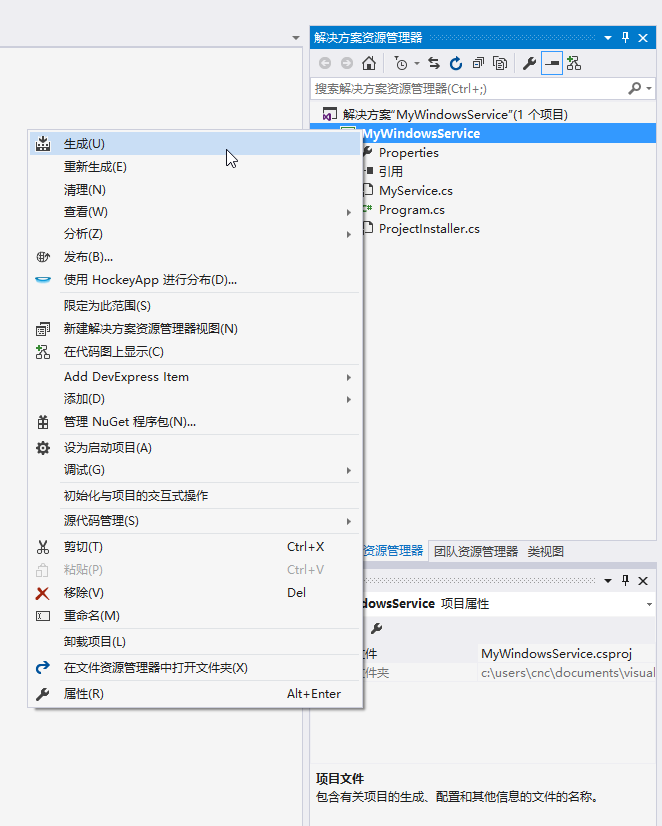




7、点击“serviceProcessInstaller1”，在“属性”窗体将Account改为LocalSystem（服务属性系统级别），如下图所示：



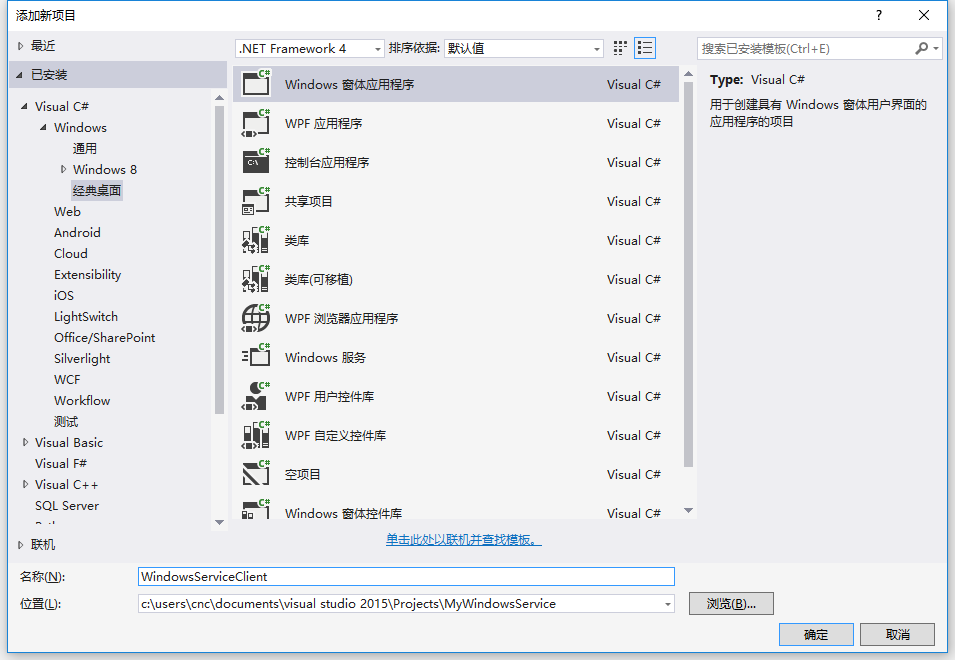
8、鼠标右键点击项目“MyWindowsService”，在弹出的上下文菜单中选择“生成”按钮，如下图所示：



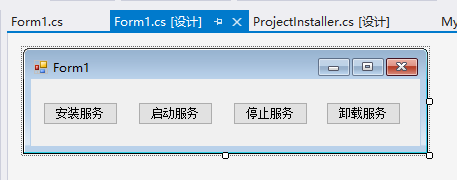
9、至此，Windows服务已经创建完毕。

**三、创建安装、启动、停止、卸载服务的Windows窗体**

1、在同一个解决方案里新建一个Windows Form项目，并命名为WindowsServiceClient，如下图所示：



2、将该项目设置为启动项目，并在窗体内添加四个按钮，分别为安装服务、启动服务、停止服务及卸载服务，如下图所示：



3、按下F7进入代码编辑界面，引用“System.ServiceProcess”及“System.Configuration.Install”，并输入如下代码：

[IMG_271](https://www.cnblogs.com/mq0036/p/javascript:void(0);)

[IMG_272](https://www.cnblogs.com/mq0036/p/javascript:void(0);)

using System;using System.Collections;using System.Windows.Forms;using System.ServiceProcess;using System.Configuration.Install;

namespace WindowsServiceClient

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

string serviceFilePath = $"{Application.StartupPath}\\MyWindowsService.exe";

string serviceName = "MyService";

//事件：安装服务

private void button1\_Click(object sender, EventArgs e)

{

if (this.IsServiceExisted(serviceName)) this.UninstallService(serviceName);

this.InstallService(serviceFilePath);

}

//事件：启动服务

private void button2\_Click(object sender, EventArgs e)

{

if (this.IsServiceExisted(serviceName)) this.ServiceStart(serviceName);

}

//事件：停止服务

private void button4\_Click(object sender, EventArgs e)

{

if (this.IsServiceExisted(serviceName)) this.ServiceStop(serviceName);

}

//事件：卸载服务

private void button3\_Click(object sender, EventArgs e)

{

if (this.IsServiceExisted(serviceName))

{

this.ServiceStop(serviceName);

this.UninstallService(serviceFilePath);

}

}

//判断服务是否存在

private bool IsServiceExisted(string serviceName)

{

ServiceController[] services = ServiceController.GetServices();

foreach (ServiceController sc in services)

{

if (sc.ServiceName.ToLower() == serviceName.ToLower())

{

return true;

}

}

return false;

}

//安装服务

private void InstallService(string serviceFilePath)

{

using (AssemblyInstaller installer = new AssemblyInstaller())

{

installer.UseNewContext = true;

installer.Path = serviceFilePath;

IDictionary savedState = new Hashtable();

installer.Install(savedState);

installer.Commit(savedState);

}

}

//卸载服务

private void UninstallService(string serviceFilePath)

{

using (AssemblyInstaller installer = new AssemblyInstaller())

{

installer.UseNewContext = true;

installer.Path = serviceFilePath;

installer.Uninstall(null);

}

}

//启动服务

private void ServiceStart(string serviceName)

{

using (ServiceController control = new ServiceController(serviceName))

{

if (control.Status == ServiceControllerStatus.Stopped)

{

control.Start();

}

}

}

//停止服务

private void ServiceStop(string serviceName)

{

using (ServiceController control = new ServiceController(serviceName))

{

if (control.Status == ServiceControllerStatus.Running)

{

control.Stop();

}

}

}

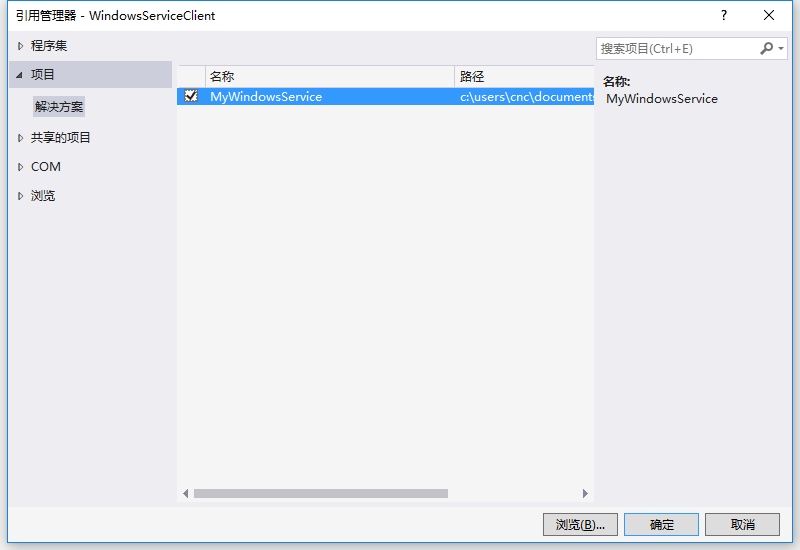
}

}

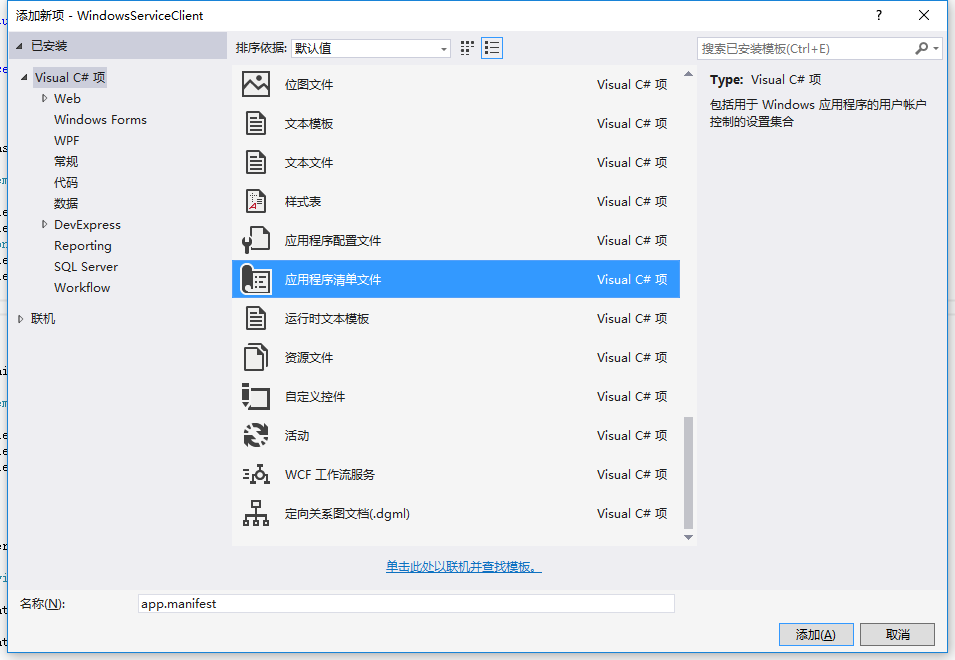
[IMG_273](https://www.cnblogs.com/mq0036/p/javascript:void(0);)

[IMG_274](https://www.cnblogs.com/mq0036/p/javascript:void(0);)

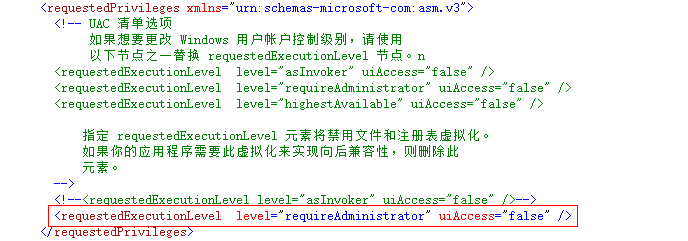
4、为了后续调试服务及安装卸载服务的需要，将已生成的MyWindowsService.exe引用到本Windows窗体，如下图所示：



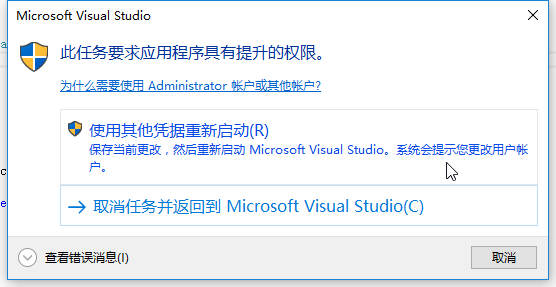
5、由于需要安装服务，故需要使用UAC中Administrator的权限，鼠标右击项目“WindowsServiceClient”，在弹出的上下文菜单中选择“添加”->“新建项”，在弹出的选择窗体中选择“应用程序清单文件”并单击确定，如下图所示：



6、打开该文件，并将<requestedExecutionLevel level="asInvoker" uiAccess="false" />改为<requestedExecutionLevel  level="requireAdministrator" uiAccess="false" />，如下图所示：

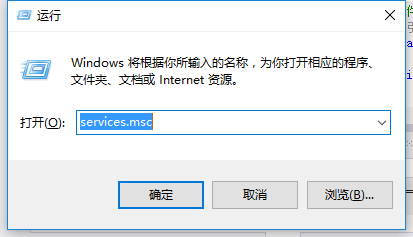


7、IDE启动后，将会弹出如下所示的窗体（有的系统因UAC配置有可能不显示），需要用管理员权限打开：

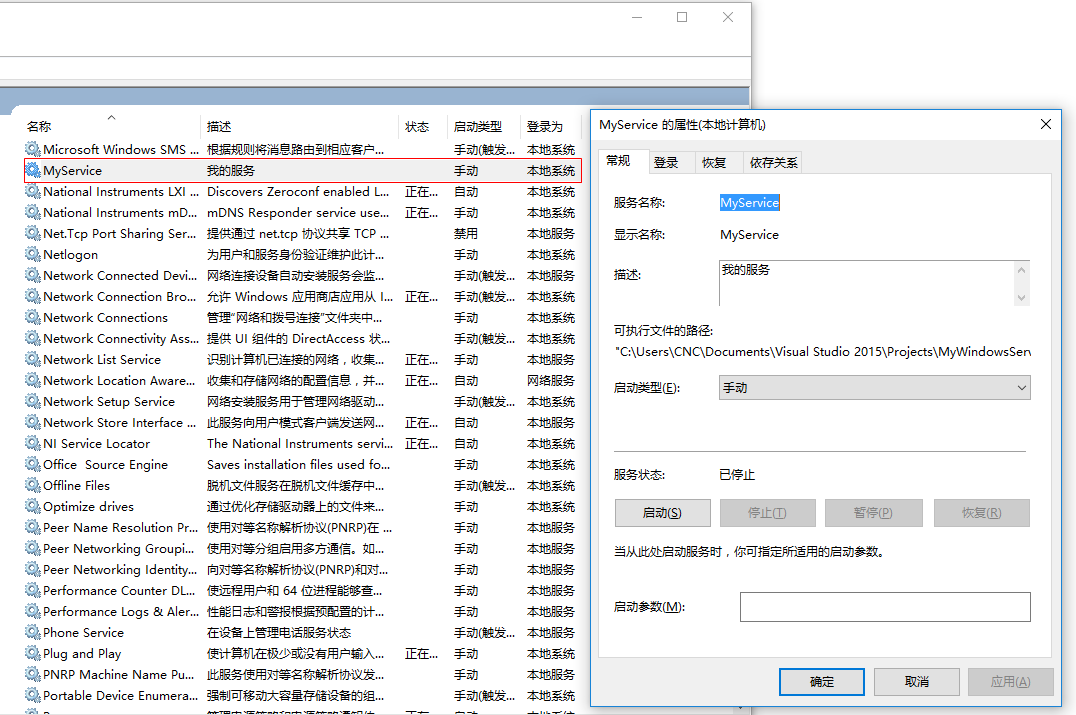


8、重新打开后，在IDE运行WindowsServiceClient项目；

9、使用WIN+R的方式打开运行窗体，并在窗体内输入services.msc后打开服务，如下图所示：



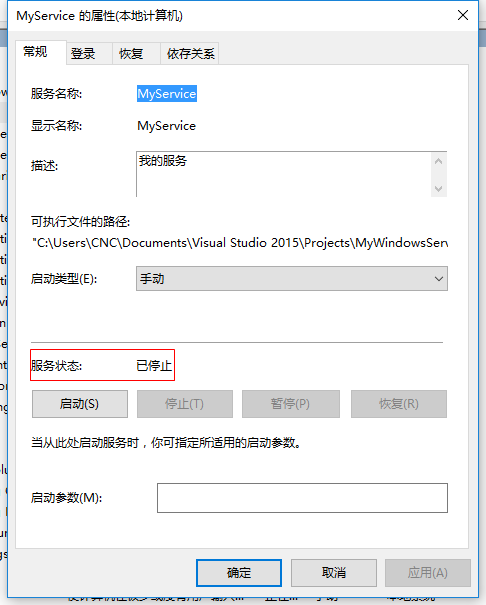
10、点击窗体内的“安装服务”按钮，将会在服务中出现MyService，如下图所示：



11、点击“运行服务”按钮，将启动并运行服务，如下所示：

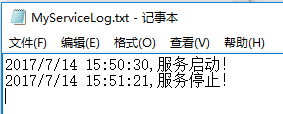


12、点击“停止服务”按钮，将会停止运行服务，如下图所示：



13、点击“卸载服务”按钮，将会从服务中删除MyService服务。

14、以上启动及停止服务将会写入D:\MyServiceLog.txt，内容如下所示：

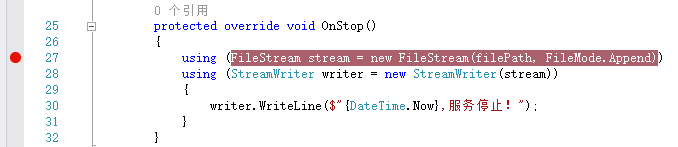


源代码下载：

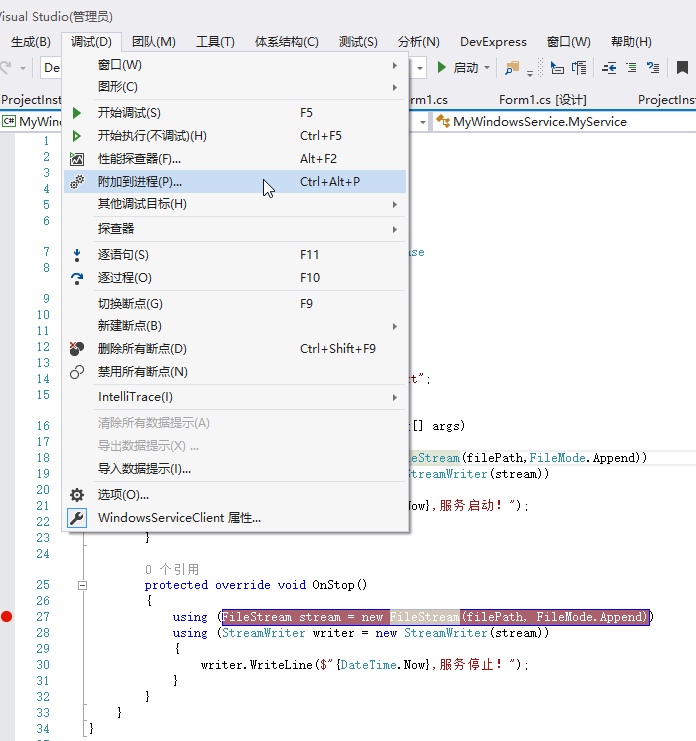
[http://pan.baidu.com/s/1kVza3Bp](http://pan.baidu.com/s/1kVza3Bp" \t "https://www.cnblogs.com/mq0036/p/_blank)

**补充：如何调试服务**

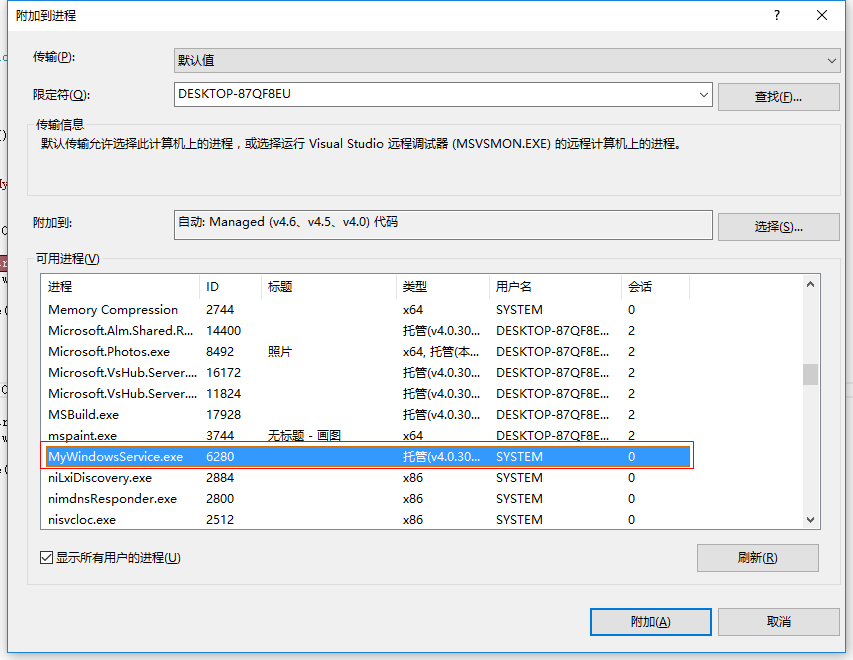
1、要调试服务，其实很简单，如需将服务附加进程到需要调试的项目里面即可，假如要调试刚才建的服务，现在OnStop事件里设置断点，如下所示：



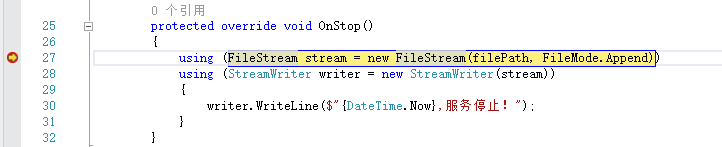
2、启动“WindowsServiceClient”项目，在“调试”菜单中选择“附件到进程”（服务必须事先安装），如下所示：



3、找到“MyWindowsService.exe”，点击“附加”按钮，如下图所示：



4、点击“停止服务”按钮，程序将会在设置断点的地方中断，如下图所示：



<https://www.cnblogs.com/mq0036/p/7875864.html>