

公告

昵称：Joye.Net  
园龄：5年8个月  
粉丝：1028  
关注：34  
+加关注

最新随笔

1. C#队列Queue实现一个简单的电商网站秒杀程序
2. C# 递归省市区三级树结构
3. System.InvalidOperationException:“ 程间操作无效: 从不是创建控件“txtPortName02”的线程访问它。”
4. 消息队列 RabbitMQ
5. .NET NPOI导出Excel详解
6. C#发展历程以及C#6.0新特性
7. C#中字典集合HashTable、Dictionary、ConcurrentDictionary三者区别
8. C#开发Windows服务
9. C# 值类型和引用类型
10. 【干货分享】JPager.Net MVC超好用轻量级分页控件

随笔分类(101)

- ABP(13)
- ASP.NET MVC(4)
- C# .NET(15)
- DDD(13)
- Design Patterns(9)
- eShopOnContainers
- HTML5(2)

随笔-75 文章-0 评论-683

C#开发Windows服务

Microsoft Windows 服务（即，以前的 NT 服务）使您能够创建在它们自己的 Windows 会话中可长时间运行的可执行应用程序。

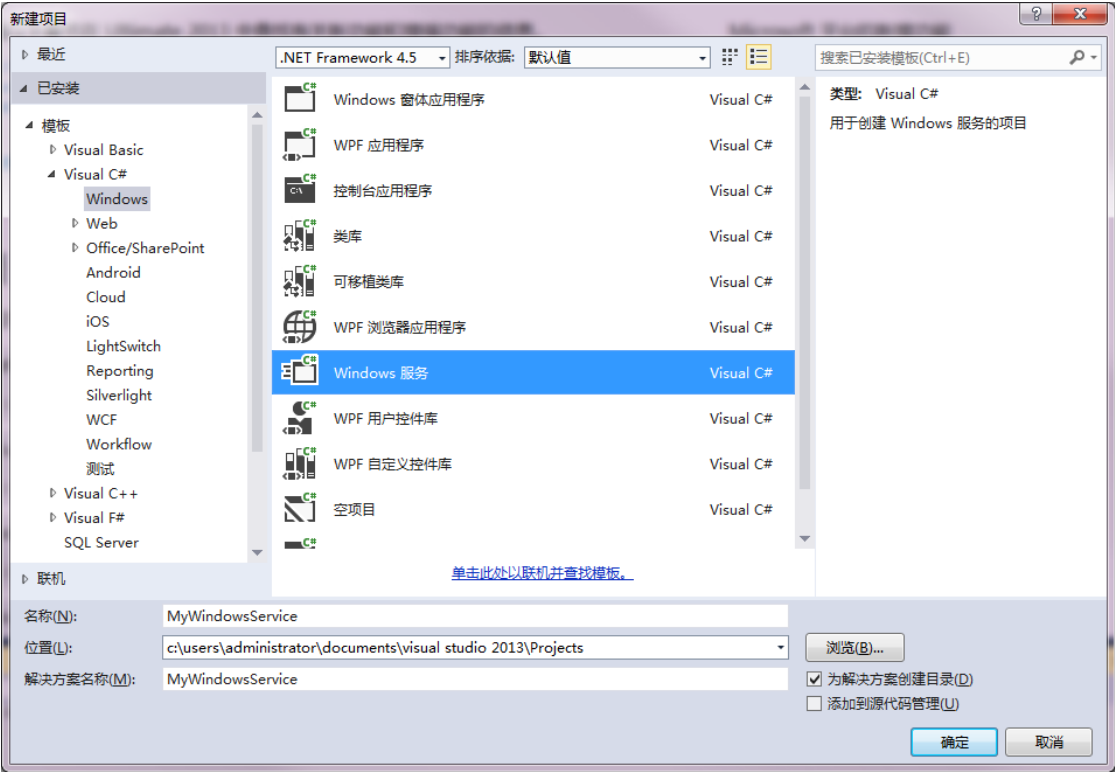
服务可以在计算机启动时自动启动，可以暂停和重新启动而且不显示任何用户界面。

服务非常适合在服务器上使用，或任何时候，为了不影响在同一台计算机上工作的其他用户，需要长时间运行功能时使用。

还可以在不同于登录用户的特定用户帐户或默认计算机帐户的安全上下文中运行服务

一、创建服务

1、VS新建一个Windows服务项目，命名MyWindowsService



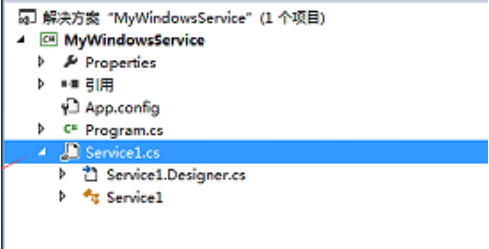
- IOC/DI(2)
- JQuery(3)
- Memcached(3)
- MongoDB(1)
- MSIL(2)
- MSSQL(2)
- NET Core
- Redis(1)
- Scrum
- 程序人生(5)
- 多线程(2)
- 负载均衡
- 架构设计(13)
- 解决方案(11)

积分与排名

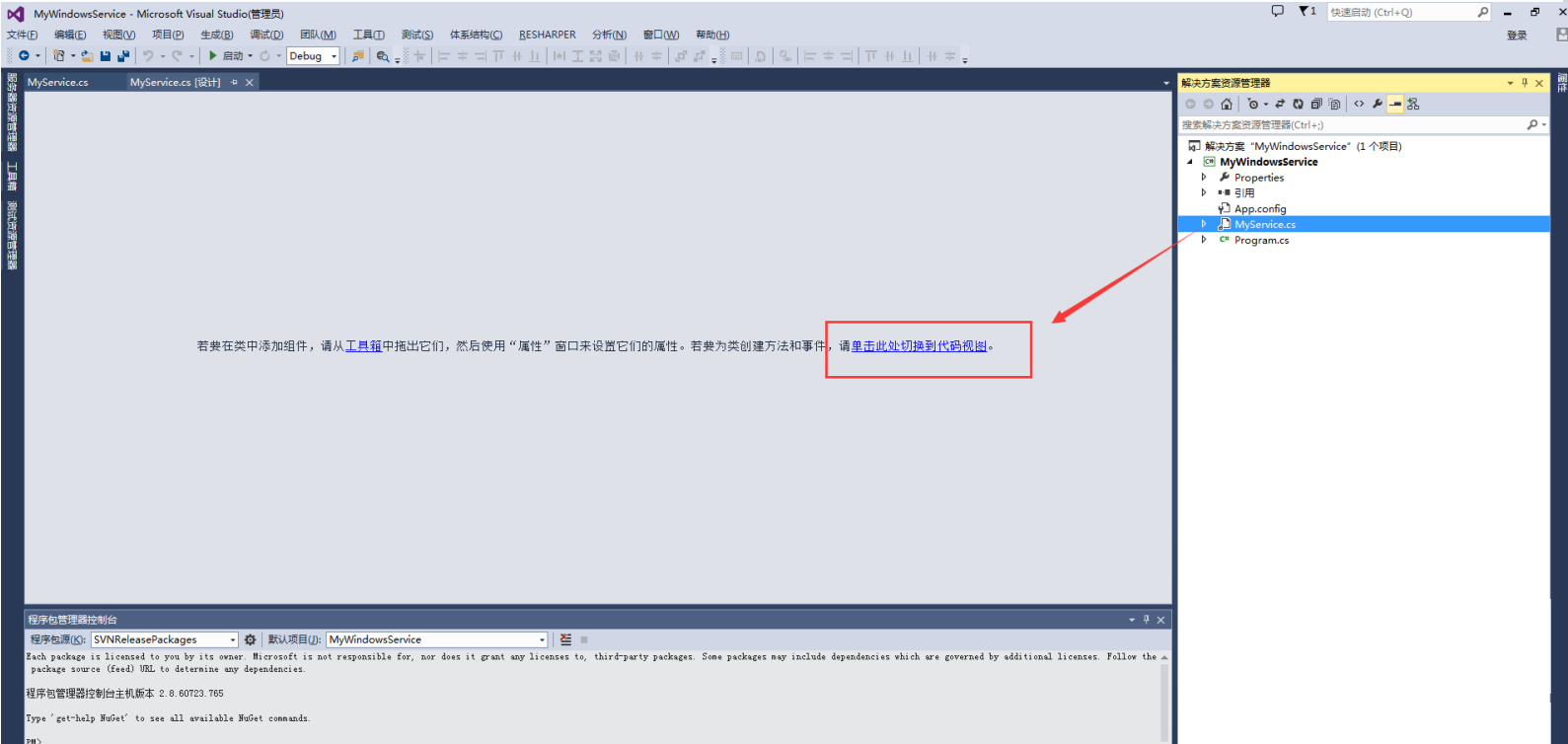
积分 - 139647  
排名 - 2788

最新评论

- 1. Re:C#队列Queue实现一个简单的电商网站秒杀程序  
internal class ThreadPar { private  
static readonly int cnt = 10;  
  
private static rea.....  
--漠里
- 2. Re:C#队列Queue实现一个简单的电商网站秒杀程序  
1、你这个是一边在抢一边在队列里移除，这样永远都在抢；  
2、判断秒杀的个数那逻辑也不对；  
--漠里
- 3. Re:.NET领域最为流行的IOC框架之一Autofac  
找了一堆文章，就这篇文章写得简单明了，其他的文章写得太乱了  
--wushuitaolove



2、修改Service1为MyService，F7进入代码页面（点击切换代码视图），编写操作逻辑代码



3、OnStart用于执行服务事件，一般采用线程方式执行方法，便于隔一段事件执行一回



```
using System;  
using System.IO;  
using System.ServiceProcess;  
using System.Threading.Tasks;  
  
namespace MyWindowsService  
{  
    public partial class MyService : ServiceBase
```

4. Re:WebAPI2使用Autofac实现IOC  
属性注入完美解决方案  
支持

--~雨落忧伤~

5. Re:.NET领域最为流行的IOC框架之一Autofac  
支持

--~雨落忧伤~

6. Re:.NET NPOI导出Excel详解  
mark 标记一下 正在学习

--秦时明月123456

7. Re:C#发展历程以及C#6.0新特性  
--金一，梦开始的地方

8. Re:C#队列Queue实现一个简单的电  
商网站秒杀程序  
貌似有问题哦。  
没有人抢到商品哦。  
全部打印“商品抢光了”

--自由的鱼

9. Re:详解.NET IL代码  
@ Joye.Net 楼主,请教一下,IL语言中的  
IL\_0000,IL\_0001,IL\_0002代表的是  
什么,而且编号还不是顺序的,如果把这个  
文件用UE打开后,怎么找到对应IL代码的  
行数

--FoxNeon

10. Re:详解.NET IL代码  
楼主,请教一下,IL语言中的  
IL\_0000,IL\_0001,IL\_0002代表的是  
什么,而且编号还不是顺序的,如果把这个  
文件用UE打开后,怎么找到对应IL代码的  
行数

--FoxNeon

阅读排行榜

1. .NET NPOI导出Excel详解(20891)

```
{
    public MyService()
    {
        InitializeComponent();
    }
    //服务开启
    protected override void OnStart(string[] args)
    {
        Task.Factory.StartNew(Handle);
    }
    //服务关闭
    protected override void OnStop()
    {

    }

    //需要定时执行的代码段
    private void Handle()
    {
        while (true)
        {
            try
            {
                var path = AppDomain.CurrentDomain.BaseDirectory + "service.log";
                var context = "MyWindowsService: Service Stopped " + DateTime.Now + "\n";
                WriteLogs(path, context);
            }
            catch (Exception)
            {
                throw;
            }
        }
    }

    public void WriteLogs(string path, string context)
    {
        var fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Write);
        var sw = new StreamWriter(fs);
        sw.BaseStream.Seek(0, SeekOrigin.End);
        sw.WriteLine(context);
    }
}
```

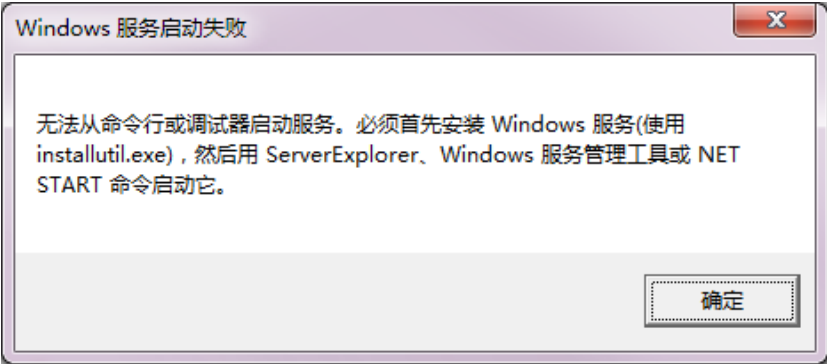
- 2. 谁让APP工程师产生了泡沫？(14368)
- 3. ASP.NET Session 简单超实用使用总结(12190)
- 4. WebAPI2使用Autofac实现IOC属性注入完美解决方案(9906)
- 5. C#中数组Array、ArrayList、泛型List<T>的比较(9607)
- 6. ASP.NET MVC5 插件化机制简单实现(8817)
- 7. 详解.NET IL代码(8382)
- 8. .NET中常用的几种解析JSON方法(7924)
- 9. 公司新加了一台友宝自动售货机引发的思考-适配器模式(7446)
- 10. .NET基于Redis缓存实现单点登录SSO的解决方案(7216)

评论排行榜

- 1. 谁让APP工程师产生了泡沫？(205)
- 2. .NET基于Redis缓存实现单点登录SSO的解决方案(53)
- 3. 程序员为什么要时刻保持危机感？(52)
- 4. Autofac 依赖注入 ASP.NET MVC5 插件机制中插件的简单实现(30)
- 5. 基于DDD的.NET开发框架 - ABP初探(29)
- 6. 公司新加了一台友宝自动售货机引发的思考-适配器模式(26)
- 7. 程序员如何应对北上广高房价示例解说-建造者模式(22)
- 8. WebAPI2使用Autofac实现IOC属性注入完美解决方案(22)
- 9. ASP.NET MVC5 插件化机制简单实现(20)

```
sw.Flush();
sw.Close();
fs.Close();
}
}
}
```

服务不能直接运行，F5会出现



二、创建服务安装程序

- 1、选中MyService.cs，右键-->查看设计器；在设计器中右键-->添加安装程序

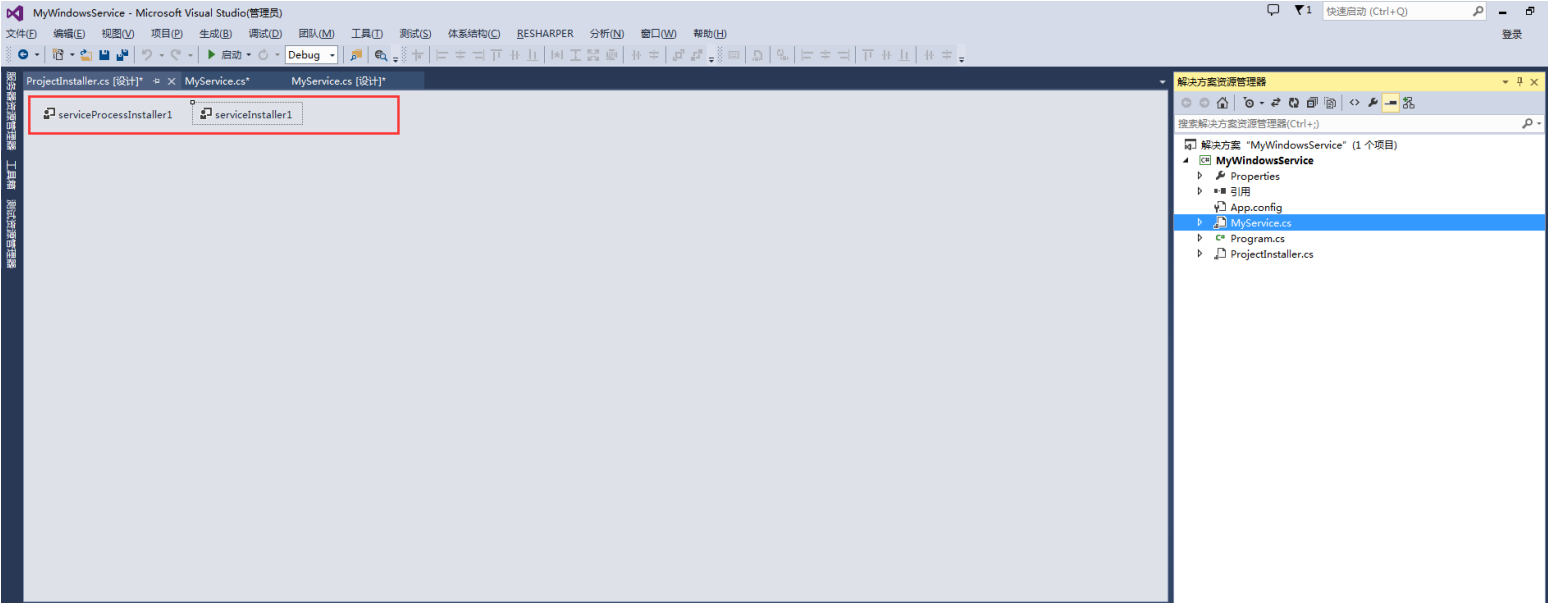


10. 【干货分享】JPager.Net MVC超好用轻量级分页控件(18)

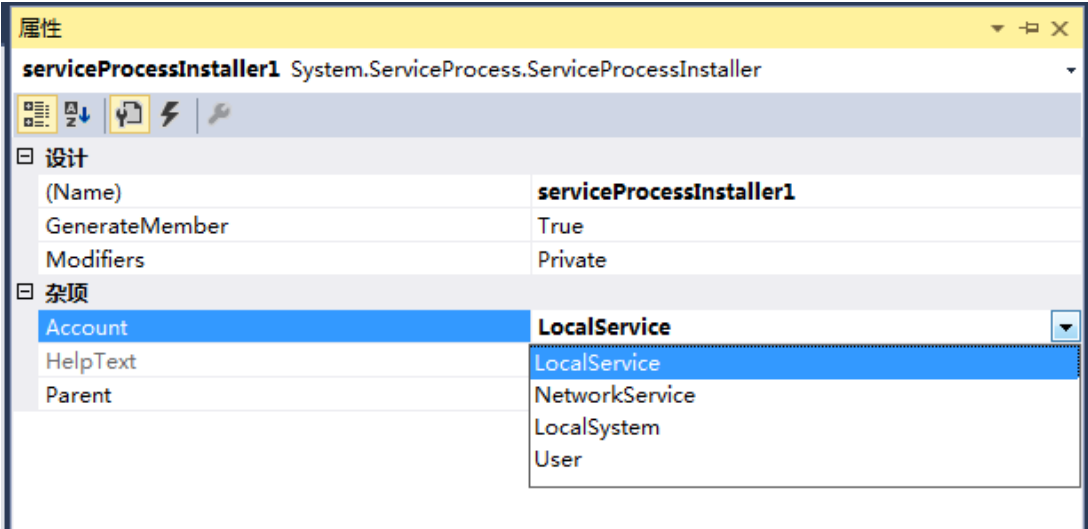
推荐排行榜

- 1. 谁让APP工程师产生了泡沫？(98)
- 2. 程序员为什么要时刻保持危机感？(64)
- 3. .NET基于Redis缓存实现单点登录SSO的解决方案(59)
- 4. 基于DDD的.NET开发框架 - ABP初探(45)
- 5. 只因路不同 —— 我选择了做一个程序员(36)
- 6. 基于DDD的.NET开发框架 - ABP的Entity设计思想(34)
- 7. .NET领域最为流行的IOC框架之一Autofac(34)
- 8. C#中数组Array、ArrayList、泛型List<T>的比较(29)
- 9. 基于DDD的.NET开发框架 - ABP分层设计(29)
- 10. 基于DDD的.NET开发框架 - ABP模块设计(28)

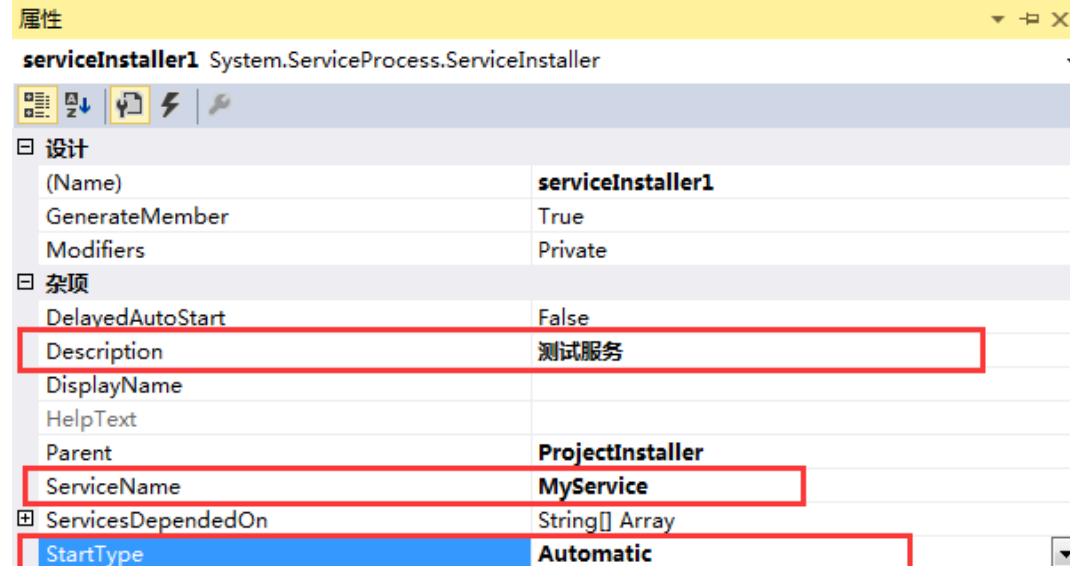
2、项目中添加了ProjectInstaller.cs文件，该文件中视图会自动会添加两个组件serviceProcessInstaller1和serviceInstaller1



3、选中serviceProcessInstaller1组件，查看属性，设置account为LocalSystem，这样能保证无论是什么用户登录都能执行。



4、选中serviceInstaller1组件，查看属性设置如下



ServiceName：表示在系统服务中的名称

StartType：Manual：手动启动，默认停止，Automatic：自动启动

Description：服务描述

### 三、安装、卸载服务

安装服务需要InstallUtil.exe，一般在电脑这个目录找到C:\Windows\Microsoft.NET\Framework64\v4.0.30319

此处需要注意的是，在C:\Windows\Microsoft.NET\Framework目录下有很多类似版本，具体去哪个目录要看项目的运行环境，例如果是.net framework2.0则需要输入 `cd C:\Windows\Microsoft.NET\Framework\v2.0.50727`

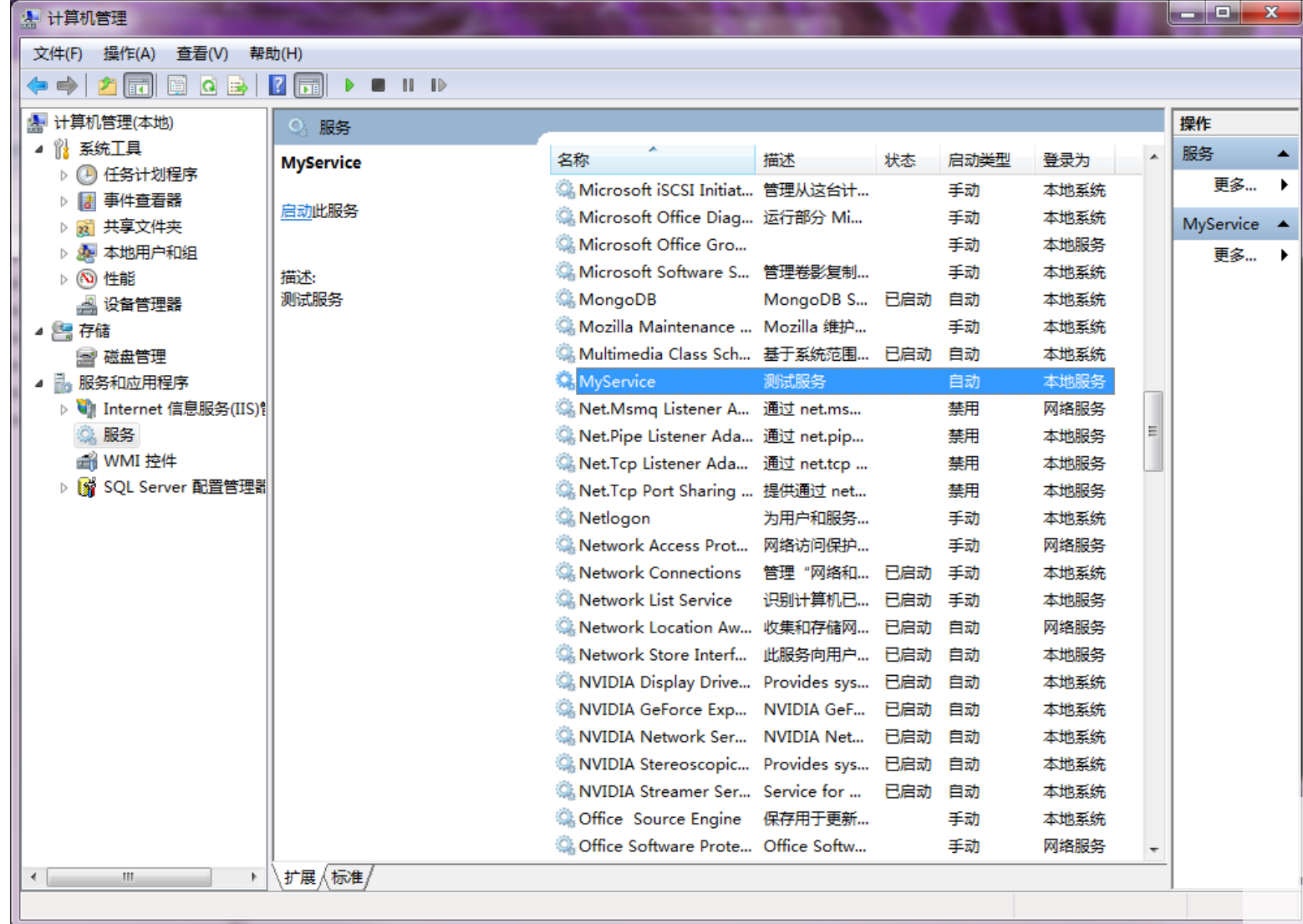
- 1、重新生成MyWindowsService项目
- 2、生成的bin目录考到要安装的目录（服务文件已经在附件中）：例如：拷贝到C:\（服务路径根据实际情况安装，卸载）
- 3、窗口+R 打开cmd

```
cd C:\Windows\Microsoft.NET\Framework64\v4.0.30319
```

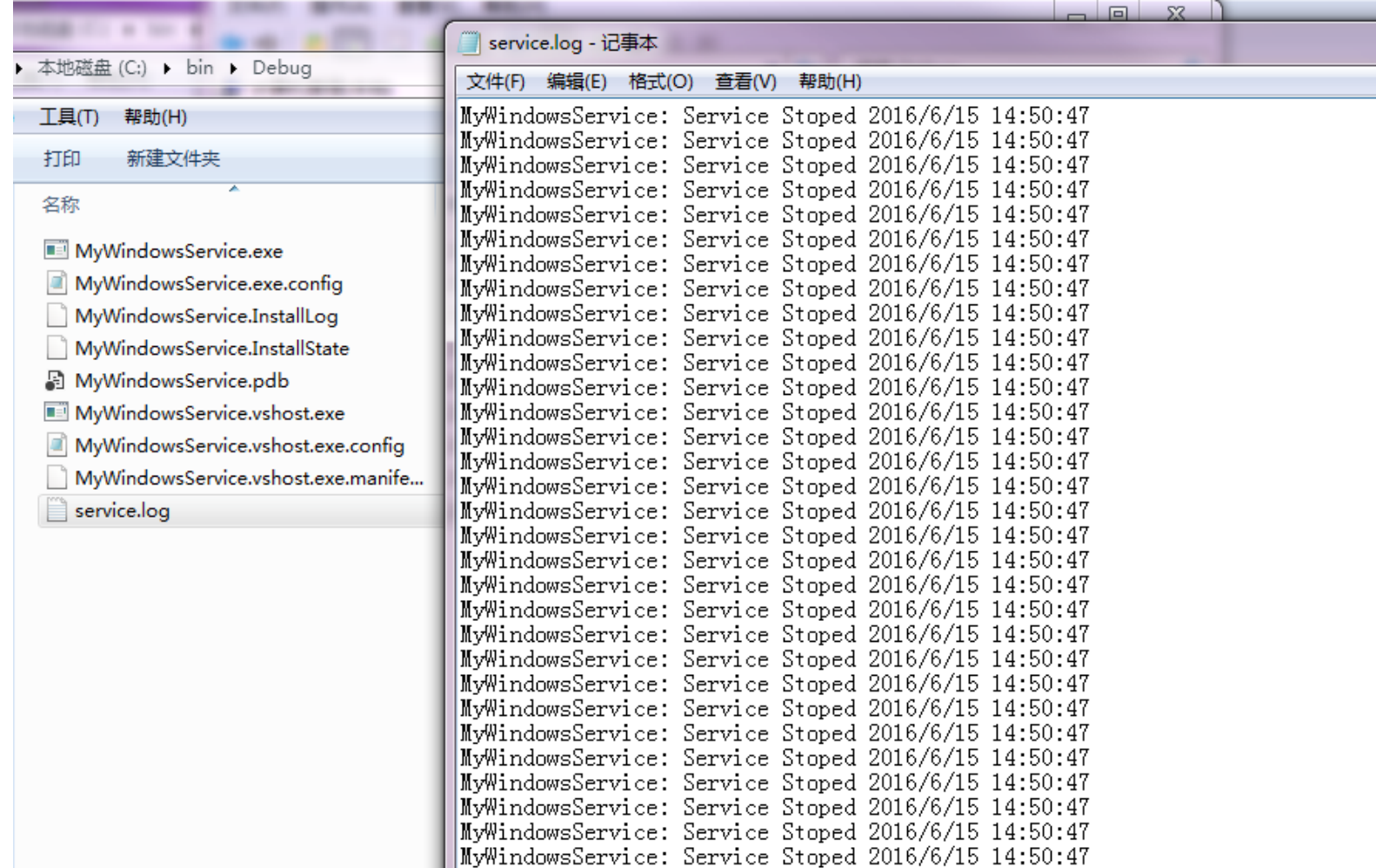
```
安装服务： Installutil C:\bin\Debug\MyWindowsService.exe
```

```
卸载服务： Installutil /u C:\bin\Debug\MyWindowsService.exe
```

- 4、我的电脑-->右键管理-->服务和应用程序-->服务 找到我们的服务



启动服务后会一直执行Handle()方法写日志

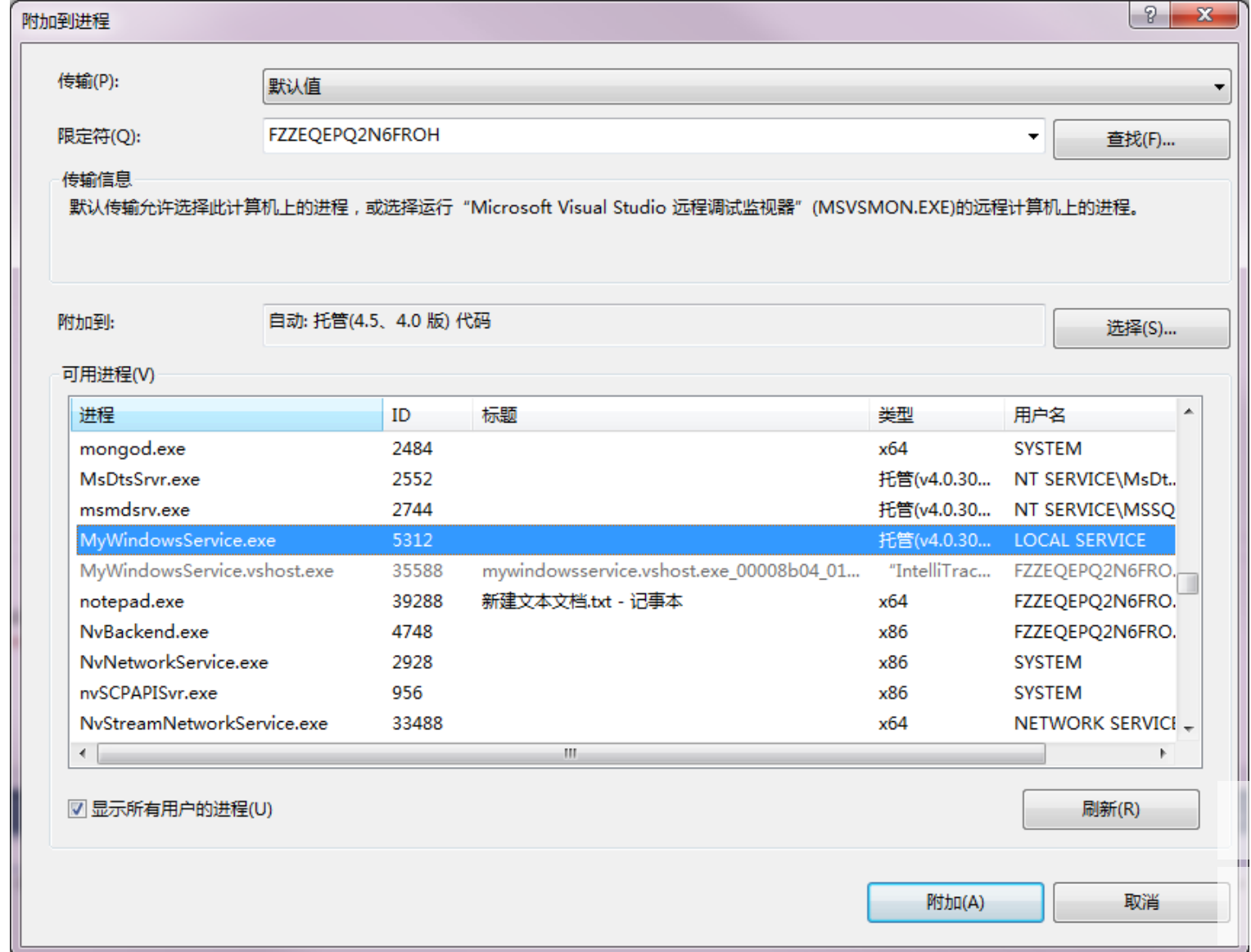


#### 四、调试服务

首先确保服务已经安装成功，并且处于已启动

VS-->调试-->附加到进程 找到我们服务名称MyWindowsService确定





然后断点就进代码了

```
ProjectInstaller.cs  Program.cs  ProjectInstaller.cs [设计]  MyService.cs  MyService.cs [设计]
MyWindowsService.MyService  Handle0

7  {
    3 个引用
8  -----public partial class MyService : ServiceBase
9  -----{
    1 个引用
10 -----public MyService()
11 -----{
12 -----InitializeComponent();
13 -----}
14 -----// 服务开启
    0 个引用
15 -----protected override void OnStart(string[] args)
16 -----{
17 -----Task.Factory.StartNew(Handle);
18 -----}
19 -----// 服务关闭
    0 个引用
20 -----protected override void OnStop()
21 -----{
22 -----}
23 -----}
24 -----
25 -----// 需要定时执行的代码段
    1 个引用
26 -----private void Handle()
27 -----{
28 -----while (true)
29 -----{
30 -----try
31 -----{
32 -----var path = AppDomain.CurrentDomain.BaseDirectory + "service.log";
33 -----var context = "MyWindowsService: Service Stopped" + DateTime.Now + "\n";
34 -----WriteLogs(path, context);
35 -----}
36 -----catch (Exception)
37 -----{
38 -----throw;
39 -----}
40 -----}
41 -----}
```

如果出现MyWindowsService.exe无法复制或进程占用的问题

打开进程管理器，关闭MyWindowsService.exe进程就可以了。