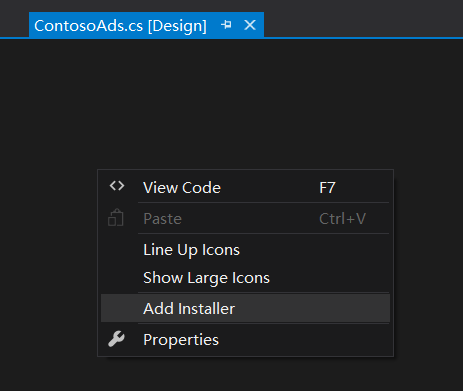
* 一．

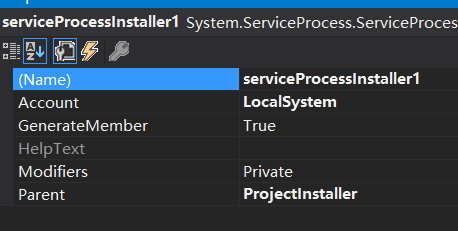
新建windows service项目，打开service类的designer，添加installer



**修改ServiceName属性为合适值**

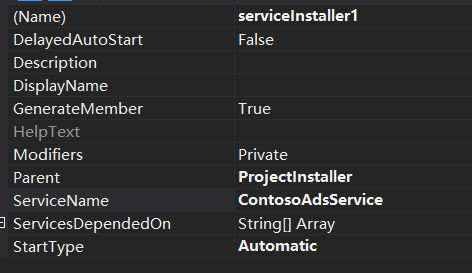
打开自动创建的projectInstallder的designer，

serviceProcessInstaller1的属性



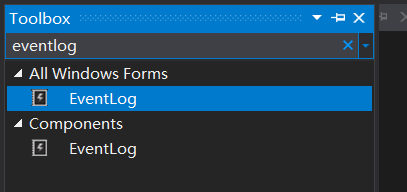
Account改为localsystem（不用写系统日志可以改为LocalService）

serviceInstaller1修改ServiceName属性为合适值

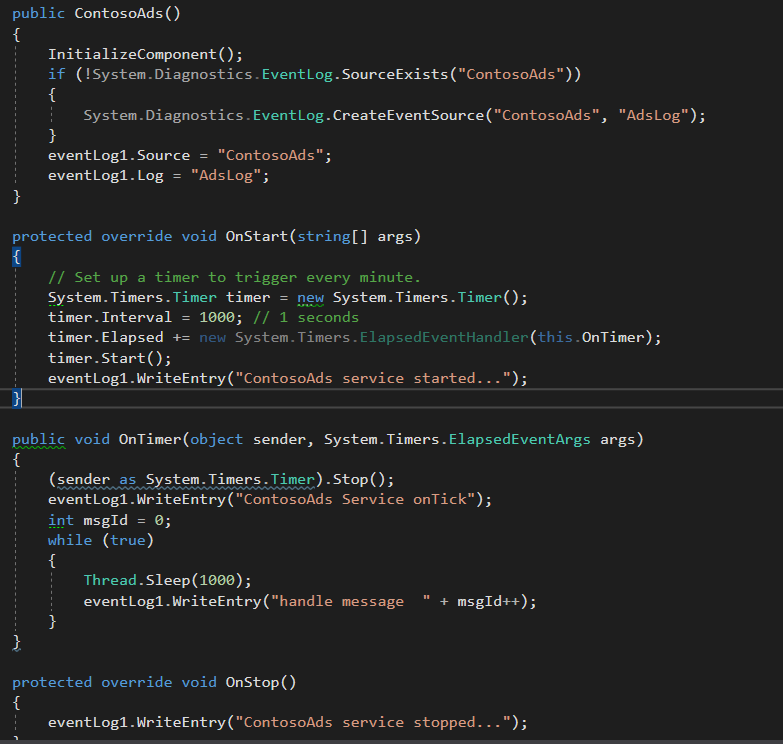


* 二．

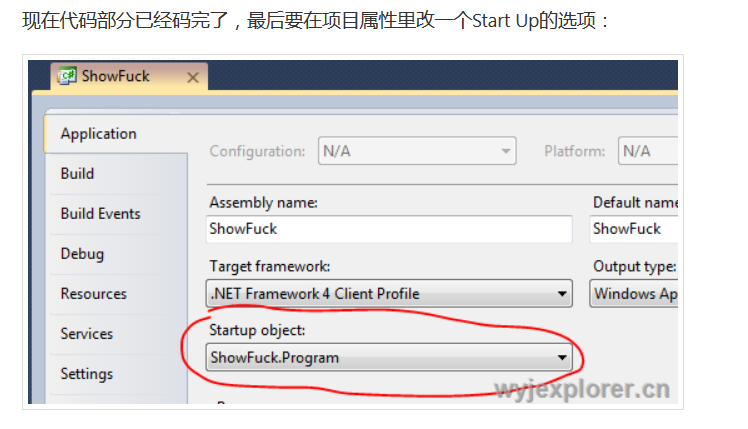
Service类的designer添加一个eventlog



由于service的OnStart方法不能执行阻塞或者长循环的内容，所以用timer来在其中触发我们需要的长操作



如此代码部分就完成了，



build项目，进入debug目录，

接下来安装部署

* 三．

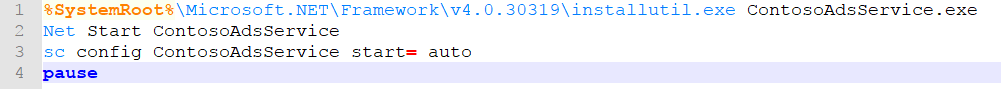
创建安装脚本install.bat

%SystemRoot%\Microsoft.NET\Framework\v4.0.30319\installutil.exe ContosoAdsService.exe

Net Start ContosoAdsService

sc config ContosoAdsService start= auto

pause



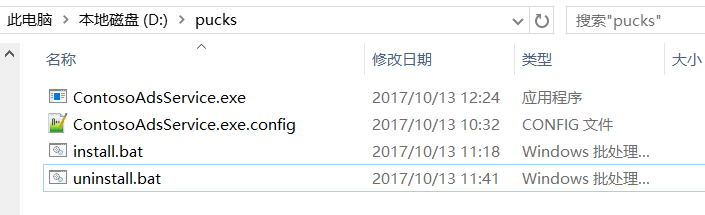
创建卸载脚本uninstall.bat

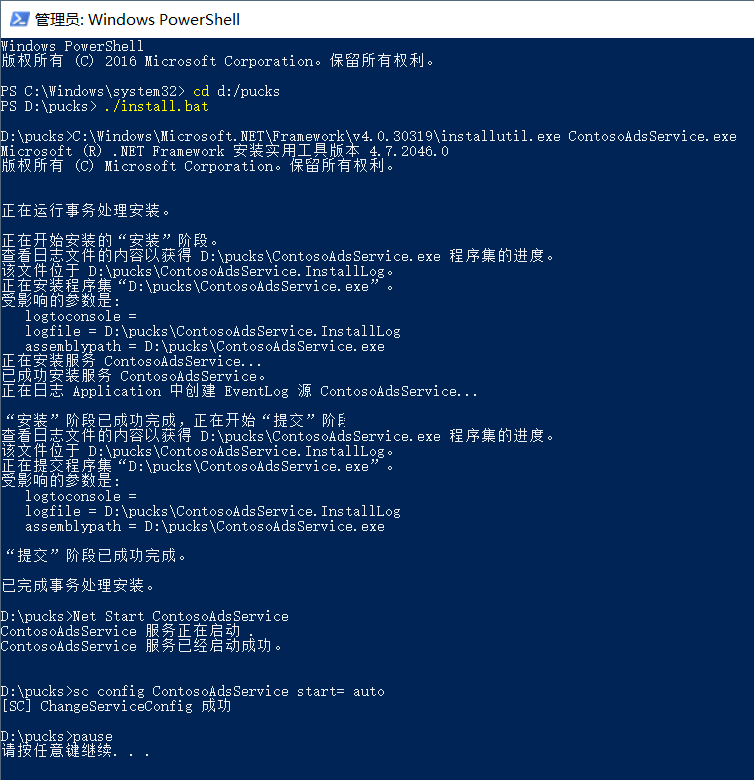
%SystemRoot%\Microsoft.NET\Framework\v4.0.30319\installutil.exe /u ContosoAdsService.exe

Pause



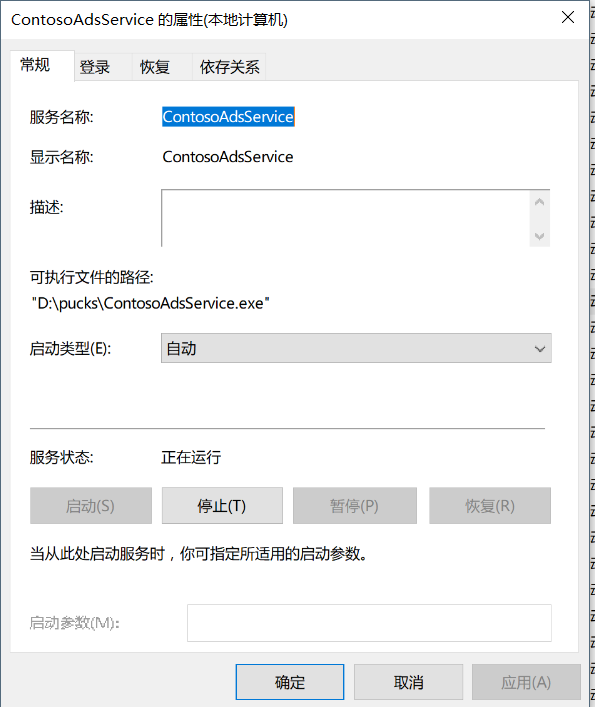
所有文件放在同一目录，管理员身份运行命令行



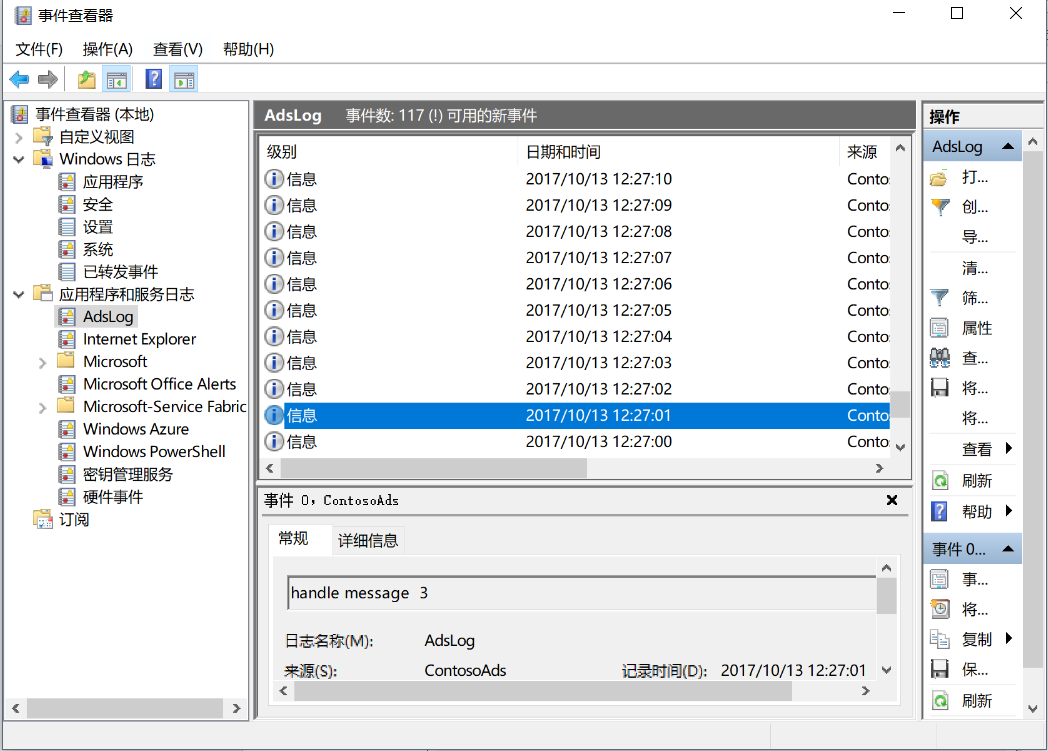


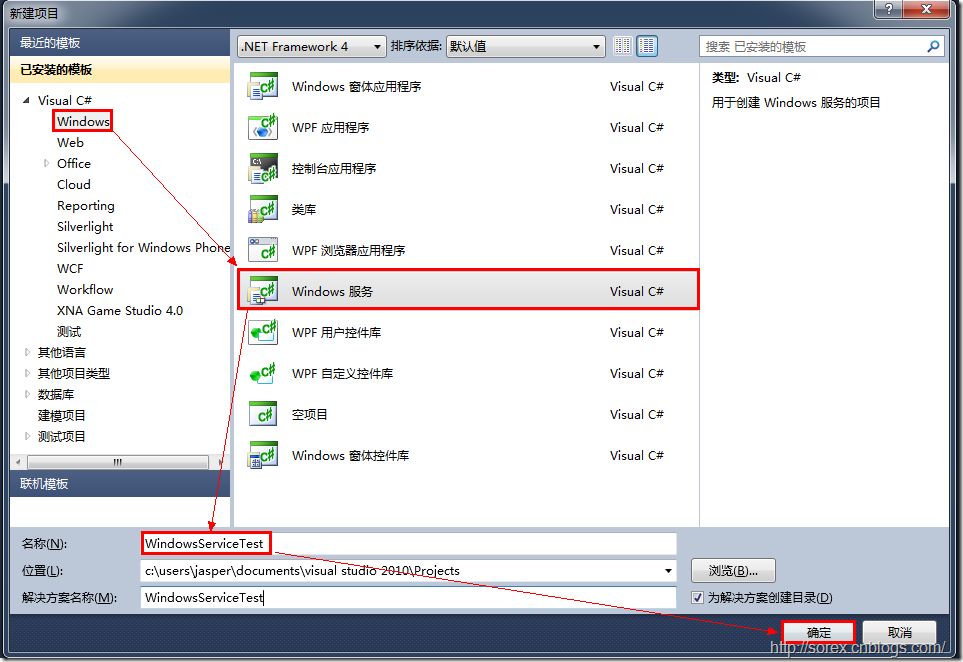
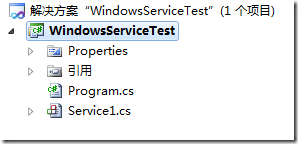
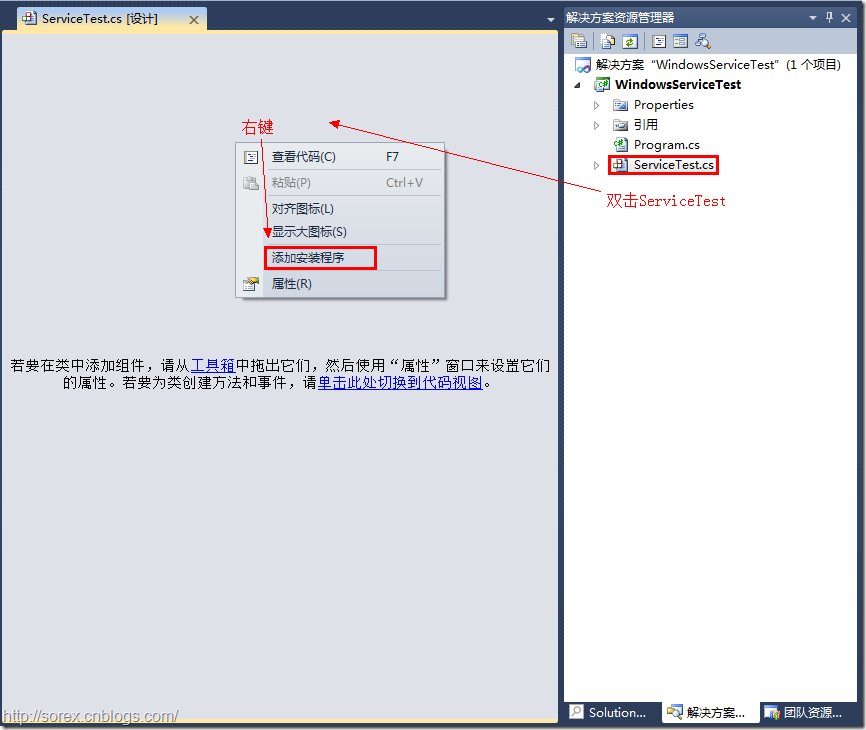
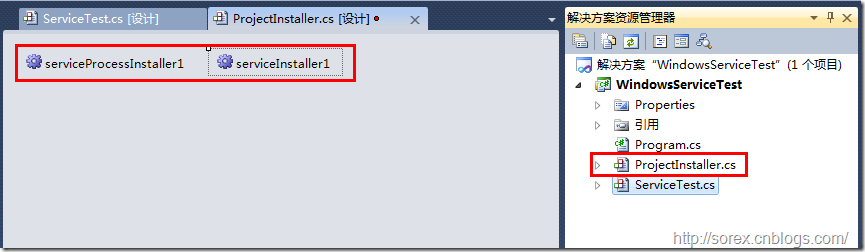
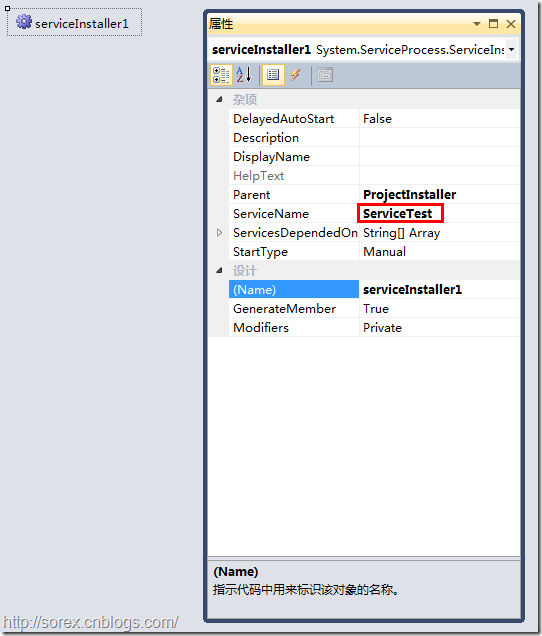
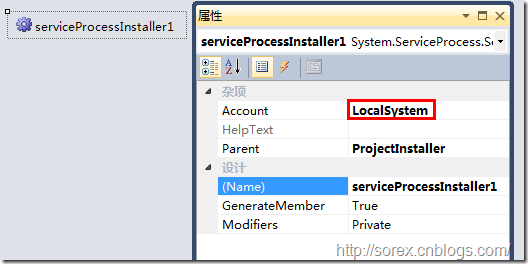
* 四．

Service面板查看



Eventlog查看



* **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*参考教程**  ：http://www.cnblogs.com/sorex/archive/2012/05/16/2502001.html
* Windows Service这一块并不复杂，但是注意事项太多了，网上资料也很凌乱，偶尔自己写也会丢三落四的。所以本文也就产生了，本文不会写复杂的东西，完全以基础应用的需求来写，所以不会对Windows Service写很深入。
* 本文介绍了如何用C#创建、安装、启动、监控、卸载简单的Windows Service 的内容步骤和注意事项。
* **一、创建一个Windows Service**
* **1）创建Windows Service项目**
* [](http://images.cnblogs.com/cnblogs_com/sorex/201205/201205151802093347.png)[](http://images.cnblogs.com/cnblogs_com/sorex/201205/201205151802143441.png)
* **2）对Service重命名**
* 将Service1重命名为你服务名称，这里我们命名为ServiceTest。
* **二、创建服务安装程序**
* **1）添加安装程序**
* [](http://images.cnblogs.com/cnblogs_com/sorex/201205/201205151802166209.png)
* [](http://images.cnblogs.com/cnblogs_com/sorex/201205/201205151802193927.png)
* 之后我们可以看到上图，自动为我们创建了ProjectInstaller.cs以及2个安装的组件。
* **2）修改安装服务名**
* 右键serviceInsraller1，选择属性，将ServiceName的值改为ServiceTest。
* [](http://images.cnblogs.com/cnblogs_com/sorex/201205/201205151802222583.png)
* **3）修改安装权限**
* 右键serviceProcessInsraller1，选择属性，将Account的值改为LocalSystem。
* [](http://images.cnblogs.com/cnblogs_com/sorex/201205/201205151802241828.png)
* **三、写入服务代码**
* **1）打开ServiceTest代码**
* 右键ServiceTest，选择查看代码。
* **2）写入Service逻辑**
* 添加如下代码：

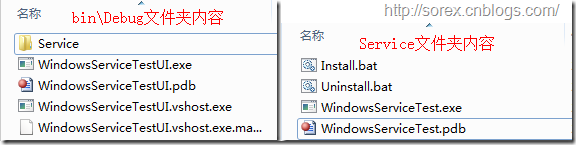
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Diagnostics;  using System.Linq;  using System.ServiceProcess;  using System.Text;    namespace WindowsServiceTest  {      public partial class ServiceTest : ServiceBase      {          public ServiceTest()          {              InitializeComponent();          }            protected override void OnStart(string[] args)          {              using (System.IO.StreamWriter sw = new System.IO.StreamWriter("C:\\log.txt", true))              {                  sw.WriteLine(DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss ") + "Start.");              }          }            protected override void OnStop()          {              using (System.IO.StreamWriter sw = new System.IO.StreamWriter("C:\\log.txt", true))              {                  sw.WriteLine(DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss ") + "Stop.");              }          }      }  } |

* 这里我们的逻辑很简单，启动服务的时候写个日志，关闭的时候再写个日志。
* **四、创建安装脚本**
* 在项目中添加2个文件如下(必须是ANSI或者UTF-8无BOM格式)：
* **1）安装脚本Install.bat**

|  |  |
| --- | --- |
| 1  2  3 | %SystemRoot%\Microsoft.NET\Framework\v4.0.30319\installutil.exe WindowsServiceTest.exe  Net Start ServiceTest  sc config ServiceTest start= auto |

* **2）卸载脚本Uninstall.bat**

|  |  |
| --- | --- |
| 1 | %SystemRoot%\Microsoft.NET\Framework\v4.0.30319\installutil.exe /u WindowsServiceTest.exe |

* **3）安装脚本说明**
* 第二行为启动服务。
* 第三行为设置服务为自动运行。
* 这2行视服务形式自行选择。
* **4）脚本调试**
* 如果需要查看脚本运行状况，在脚本最后一行加入pause
* **五、在C#中对服务进行控制**
* **0）配置目录结构**
* 简历一个新WPF项目，叫WindowsServiceTestUI，添加对System.ServiceProcess的引用。
* 在WindowsServiceTestUI的bin\Debug目录下建立Service目录。
* 将WindowsServiceTest的生成目录设置为上面创建的Service目录。
* 生成后目录结构如下图
* [](http://images.cnblogs.com/cnblogs_com/sorex/201205/201205151802256156.png)
* **1）安装**
* 安装时会产生目录问题，所以安装代码如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | string CurrentDirectory = System.Environment.CurrentDirectory;  System.Environment.CurrentDirectory = CurrentDirectory + "\\Service";  Process process = new Process();  process.StartInfo.UseShellExecute = false;  process.StartInfo.FileName = "Install.bat";  process.StartInfo.CreateNoWindow = true;  process.Start();  System.Environment.CurrentDirectory = CurrentDirectory; |

* **2）卸载**
* 卸载时也会产生目录问题，所以卸载代码如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | string CurrentDirectory = System.Environment.CurrentDirectory;  System.Environment.CurrentDirectory = CurrentDirectory + "\\Service";  Process process = new Process();  process.StartInfo.UseShellExecute = false;  process.StartInfo.FileName = "Uninstall.bat";  process.StartInfo.CreateNoWindow = true;  process.Start();  System.Environment.CurrentDirectory = CurrentDirectory; |

* **3）启动**
* 代码如下：

|  |  |
| --- | --- |
| 1  2  3  4  5 | using System.ServiceProcess;      ServiceController serviceController = new ServiceController("ServiceTest");  serviceController.Start(); |

* **4）停止**

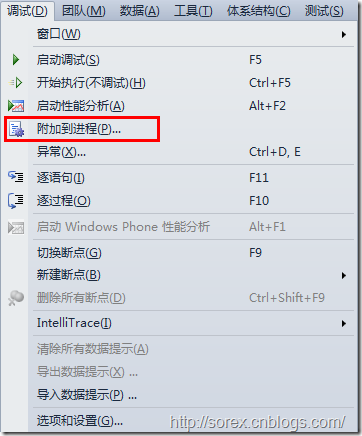
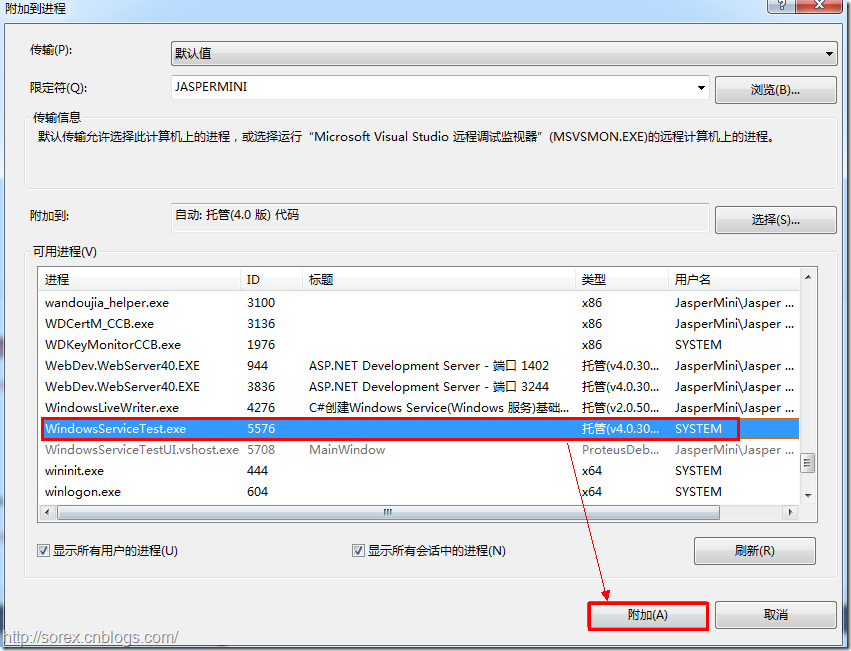
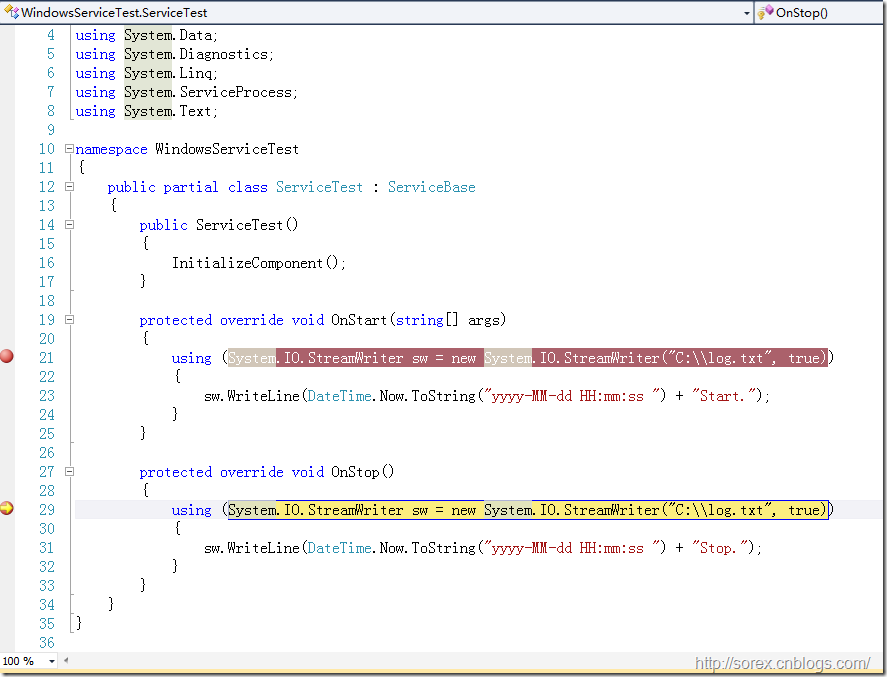
|  |  |
| --- | --- |
| 1  2  3 | ServiceController serviceController = new ServiceController("ServiceTest");  if (serviceController.CanStop)      serviceController.Stop(); |

* **5）暂停/继续**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | ServiceController serviceController = new ServiceController("ServiceTest");  if (serviceController.CanPauseAndContinue)  {      if (serviceController.Status == ServiceControllerStatus.Running)          serviceController.Pause();      else if (serviceController.Status == ServiceControllerStatus.Paused)          serviceController.Continue();  } |

* **6）检查状态**

|  |  |
| --- | --- |
| 1  2 | ServiceController serviceController = new ServiceController("ServiceTest");  string Status = serviceController.Status.ToString(); |

* **六、调试Windows Service**
* **1）安装并运行服务**
* **2）附加进程**
* [](http://images.cnblogs.com/cnblogs_com/sorex/201205/20120515180226136.png)[](http://images.cnblogs.com/cnblogs_com/sorex/201205/201205151802305628.png)
* **3）在代码中加入断点进行调试**
* [](http://images.cnblogs.com/cnblogs_com/sorex/201205/201205151802343181.png)
* **七、总结**
* 本文对Windows service的上述配置都未做详细解释，但是按上述步骤就可以制作可运行的Windows Service，从而达到了工作的需求。