



黑客派

C# 开发一个简单的 Window Service 入门

作者: [hi-asan](#)

原文链接: <https://hacpai.com/article/1519978875853>

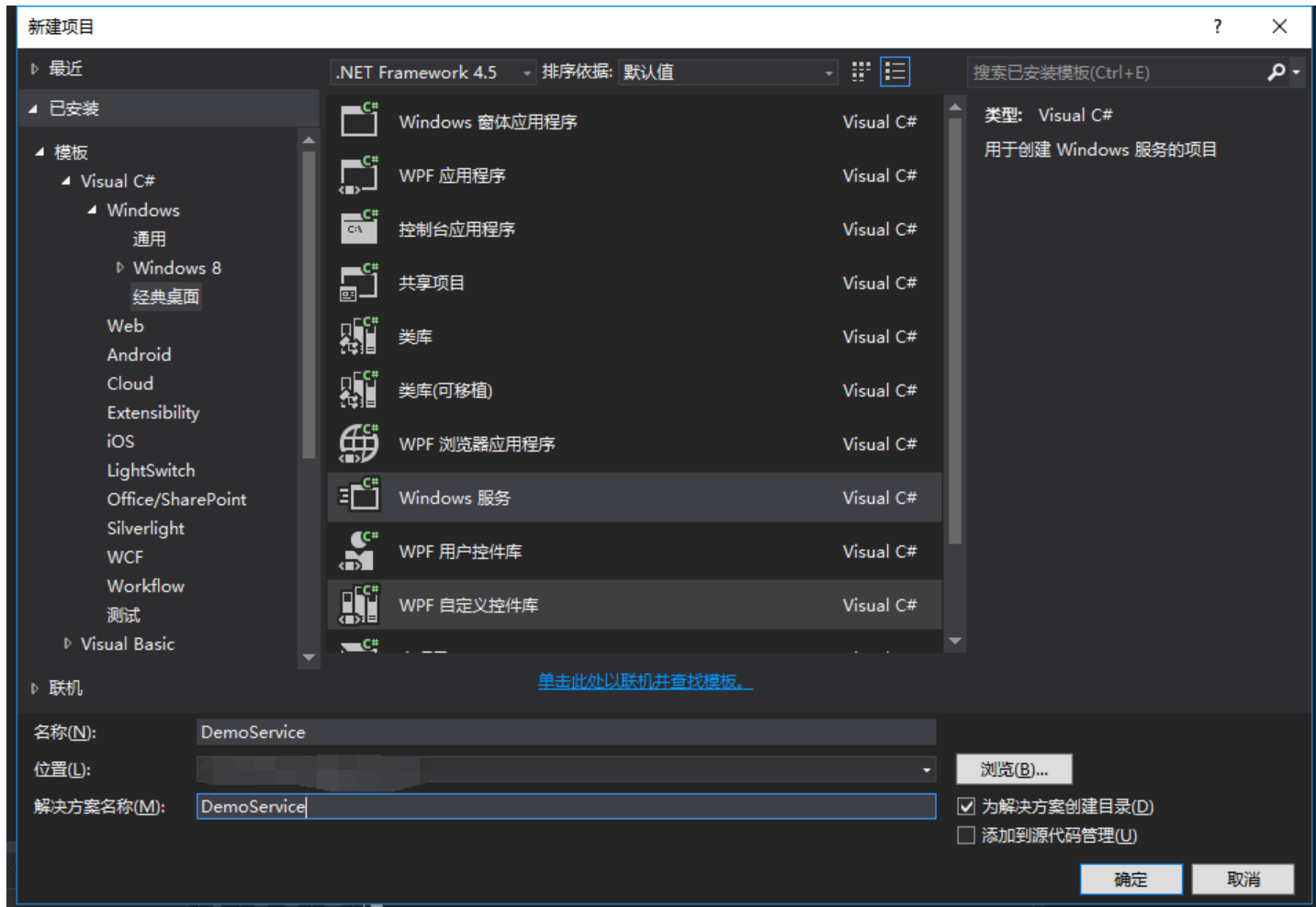
来源网站: 黑客派

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

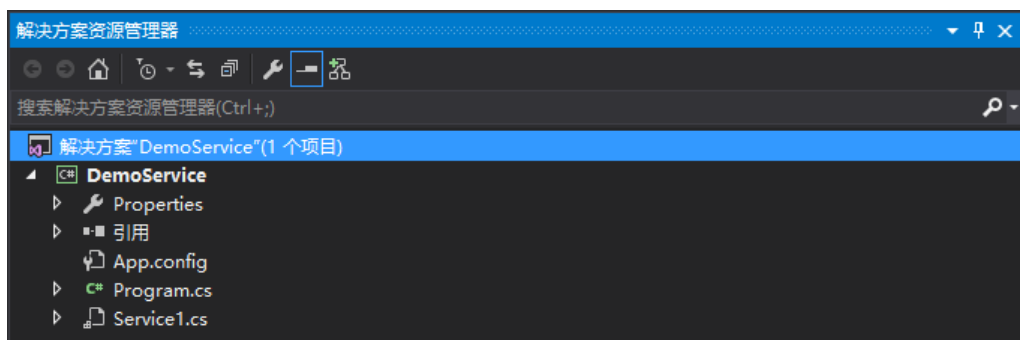
开始

创建项目

创建一个Windows服务的项目



创建成功后的目录结构



App.config 应用程序配置文件

Program.cs 程序入口文件

Service1.cs WindowService 主文件

准备代码

创建日志类

创建一个日志类**LogHeper.cs**

```
class LogHelper
{
    public static void Info(string msg)
    {
        if (!Directory.Exists("C:\\\\DemoServiceLog"))
            Directory.CreateDirectory("C:\\\\DemoServiceLog");
        string file = "C:\\\\DemoServiceLog\\" + DateTime.Now.ToString("yyyy-MM-dd") + ".log";
        if (!File.Exists(file))
        {
            StreamWriter sw = File.AppendText(file);
            sw.Close();
        }
        while (true)
        {
            try
            {
                StreamWriter sw = File.AppendText(file);
                sw.WriteLine(DateTime.Now.ToString("HH:mm:ss.s") + " " + msg);
                sw.Close();
                break;
            }
            catch (Exception e)
            {
                System.Threading.Thread.Sleep(50);
                continue;
            }
        }
    }
}
```

调整Service类

在强迫症的驱使下将**Service1.cs**改名为**Service.cs**

同时**Program.cs**类中也需要指定**Service.cs**

```
static class Program
{
    /// <summary>
    /// 应用程序的主入口点。
    /// </summary>
    static void Main()
    {
        ServiceBase[] ServicesToRun;
        ServicesToRun = new ServiceBase[]
        {
            new Service()
        }
    }
}
```

```
};  
ServiceBase.Run(ServicesToRun);  
}  
}
```

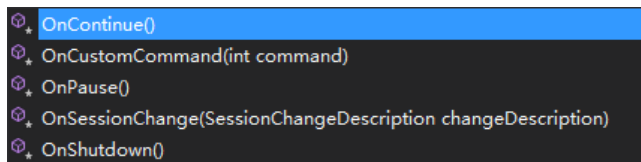
是的，它可以在ServiceBase里面指定多个服务。在这个Demo里面我们只运行一个。

编写服务程序的事件

右键**Service.cs**类查看代码

```
public partial class Service : ServiceBase  
{  
    public Service()  
    {  
        InitializeComponent();  
    }  
    protected override void OnStart(string[] args)  
    {  
        LogHelper.Info("Service is start...");  
    }  
  
    protected override void OnStop()  
    {  
        LogHelper.Info("Service is stop...");  
    }  
}
```

默认重写了两个事件，一个服务启动(OnStart)事件，一个服务停止事件(OnStart)



还支持 暂停，继续，电源关闭等事件

需要注意一点的是，在这些事件中的代码千万不能出现堵塞以及死循环的代码。否则会导致启动或停失败。甚至卸载都会出现问题。

曾不小心在启动事件中，监听了一个端口，堵塞在那里，导致启动不成功。卸载过程中提示目标服务被标记为删除，最后通过删除注册表信息才得以卸载。

服务的安装与卸载

服务的信息配置

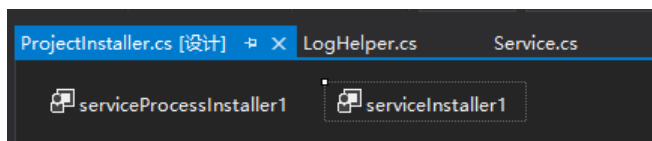
目前一个简单的WindowsService算是完成。在启动和关闭的时候将记录在日志当中。

那么怎么让他安装在用户的机器上呢？

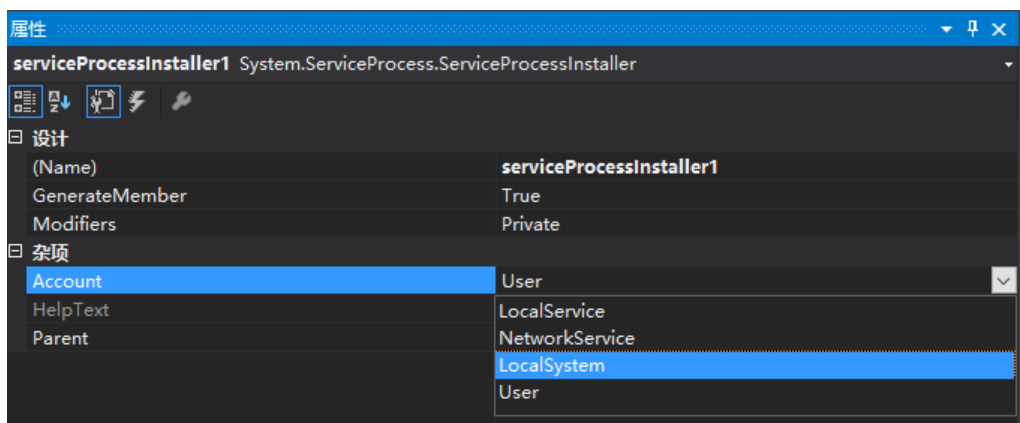
其实也很简单，首先双击**Service.cs**

在激活的标签页里面右键添加安装程序

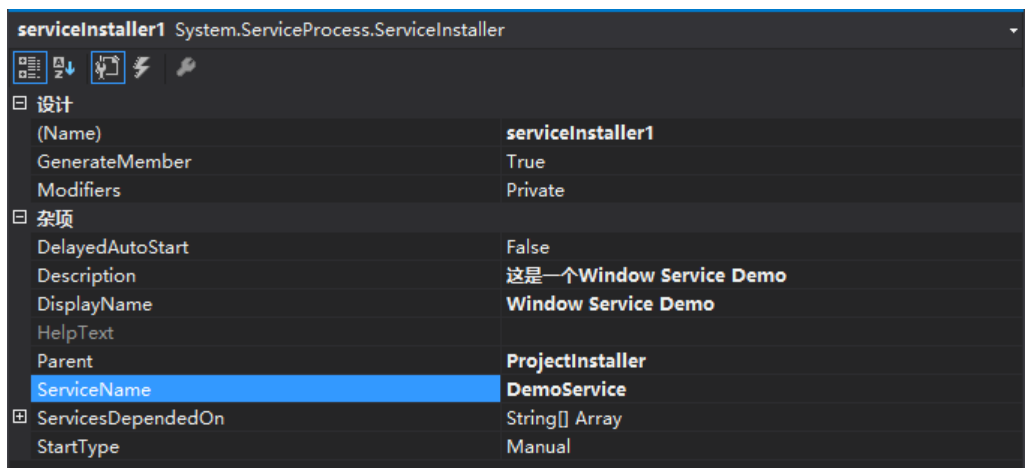
项目目录里面会多出一个**ProjectInstaller.cs**项，双击后出现



选中**serviceProcessInstaller1** 选择Account为LocalSystem



选中**serviceInstaller1** 后配置服务信息



Description 服务描述

DisplayName 给用户的一个友好名称

ServiceName 服务的名称

服务安装与卸载批处理文件

创建一个install.bat文件

@ECHO OFF

echo 准备安装服务

pause

echo 安装服务...

echo -----

%SystemRoot%\Microsoft.NET\Framework\v4.0.30319\installutil.exe %~dp0\DemoService.exe

```
net start DemoService
sc config DemoService start=auto
echo -----
echo 安装服务成功!
pause
```

调用.NET Framework 的installutil.exe程序进行安装服务

安装成功后通过net start DemoService 启动服务

sc config Demoservice start=auto 配置服务的启动类型为自动，相当于开机启动

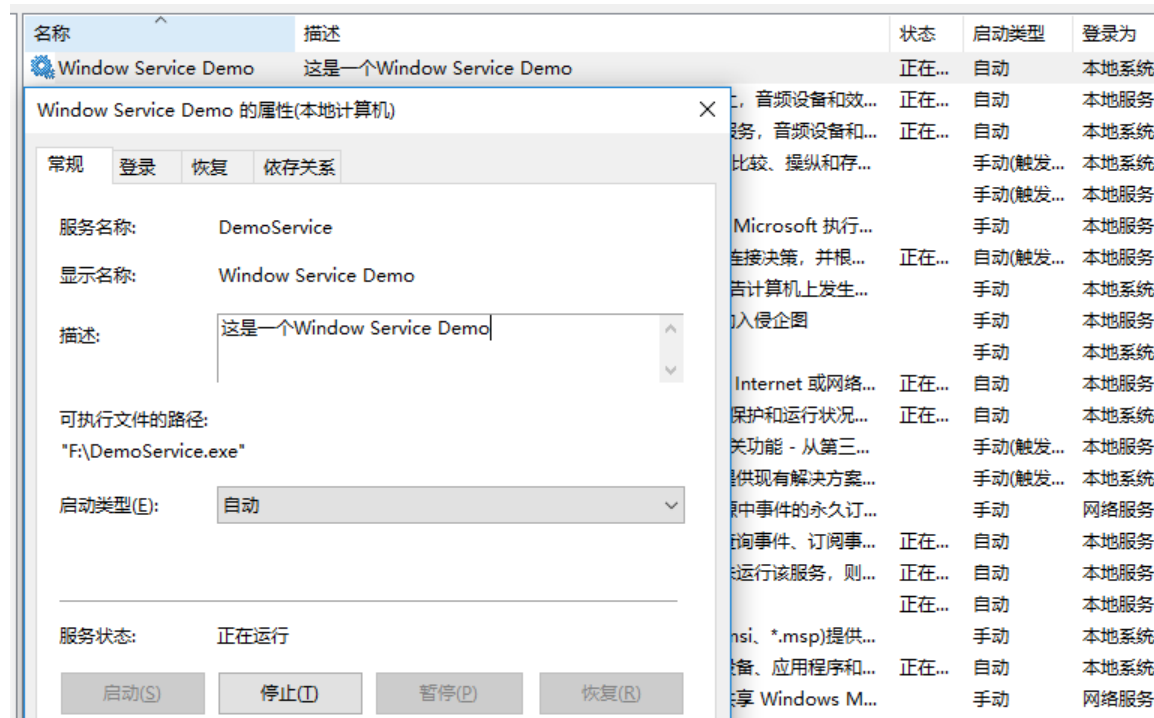
创建一个uninstall.bat文件

```
@ECHO OFF
echo 准备卸载服务
pause
echo 卸载服务...
echo -----
%SystemRoot%\Microsoft.NET\Framework\v4.0.30319\installutil.exe %~dp0\DemoService.exe
-U
sc delete DemoService
Net Stop DemoService
echo -----
echo 卸载服务成功!
pause
```

将这两个文件与项目生成出来的DemoService.exe放在同级目录下，右键已管理员方式运行即可完成装与卸载。

在C盘产生的日志文件中也记录了我们在代码中输出的信息

Win+R 输入service.msc即可找到安装成功的服务



其中 **Window Service Demo** 只是一个友好的名称而并非服务真正的名称

真正的名称在我们刚刚在 **serviceInstaller1** 配置的 **ServiceName**

结语

一个简单的 Window Service 就已经完成了。虽然他并没有完成什么特别令人有成就感的行为，也没做出什么惊天地泣鬼神的操作。

但这只是一个引子。一个 Window Service 入门的引子。通过 Window Service 我们可以做很多有意思事情。

我主要是做 JAVA WEB 开发的，大家都知道浏览器最初设计为了安全考虑，浏览器上所执行的脚本是允许和用户的机器交互的，也仅仅是缓存一些东西而已，对用户威胁不大。

举一个简单的例子，浏览器上传文件，可以说是都是用户自主触发的，都是经过用户的同意的。正常情况下是不可能自动将用户的文件上传上去。

最早浏览器与客户端交互的一个中间件叫做 **OCX** 控件或 **ActiveX** 控件，或许还有其他的控件。但是他只能在那个令人诟病的 IE 浏览器上使用。

后面我将会写一篇关于通过 Window 服务让浏览器与客户端交互的一个案例。满足一些特殊需求。