

```
Administrator: Windows PowerShell
PS C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug> InstallUtil.exe .\WalkThroughService.exe
Microsoft (R) .NET Framework Installation utility Version 4.0.30319.33440
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:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.exe assembly's progress.
The file is located at C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.InstallLog.
Installing assembly 'C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.exe'.
Affected parameters are:
  logtoconsole =
  logfile = C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.InstallLog
  assemblypath = C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.exe
Installing service MyNewService...
Service MyNewService has been successfully installed.
Creating EventLog source MyNewService in log Application...

The Install phase completed successfully, and the Commit phase is beginning.
See the contents of the log file for the C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.exe assembly's progress.
The file is located at C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.InstallLog.
Committing assembly 'C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.exe'.
Affected parameters are:
  logtoconsole =
  logfile = C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.InstallLog
  assemblypath = C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.exe

The Commit phase completed successfully.

The transacted install has completed.
PS C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug>
```

Computer Management

File Action View Help

Computer Management (Local)

- System Tools
 - Task Scheduler
 - Event Viewer
 - Shared Folders
 - Local Users and Groups
 - Performance
 - Device Manager
- Storage
 - Disk Management
- Services and Applications
 - Services
 - WMI Control
 - SQL Server Configuration I

Services

MyNewService

[Stop the service](#)
[Restart the service](#)

Name	Description	Status	Startup Type	Log On As
KtmRm for Distributed Tran...	Coordinates...		Manual (Trig...	Network S...
Link-Layer Topology Discov...	Creates a N...		Manual	Local Service
Local Session Manager	Core Windo...	Running	Automatic	Local Syste...
Media Center Extender Servi...	Allows Med...		Disabled	Local Service
Microsoft Account Sign-in ...	Enables use...		Manual (Trig...	Local Syste...
Microsoft iSCSI Initiator Ser...	Manages In...		Manual	Local Syste...
Microsoft Keyboard Filter	Controls ke...		Disabled	Local Syste...
Microsoft Office Diagnostic...	Run portio...		Manual	Local Syste...
Microsoft Software Shadow...	Manages so...		Manual	Local Syste...
Microsoft Storage Spaces S...	Host service...		Manual	Network S...
Mozilla Maintenance Service	The Mozilla ...		Manual	Local Syste...
Multimedia Class Scheduler	Enables rela...	Running	Automatic	Local Syste...
MyNewService		Running	Manual	Local Syste...
MySQL		Running	Automatic	Local Syste...
Net.Msmq Listener Adapter	Receives act...		Disabled	Network S...
Net.Pipe Listener Adapter	Receives act...	Running	Automatic	Local Service
Net.Tcp Listener Adapter	Receives act...	Running	Automatic	Local Service
Net.Tcp Port Sharing Service	Provides abi...	Running	Manual	Local Service
Netlogon	Maintains a ...		Manual	Local Syste...
Network Access Protection ...	The Networ...		Manual	Network S...
Network Connected Device...	Network Co...		Manual (Trig...	Local Service
Network Connection Broker	Brokers con...	Running	Manual (Trig...	Local Syste...
Network Connections	Manages o...		Manual	Local Syste...
Network Connectivity Assis...	Provides Dir...		Manual (Trig...	Local Syste...
Network List Service	Identifies th...	Running	Manual	Local Service
Network Location Awareness	Collects an...	Running	Automatic	Network S...
Network Store Interface Ser...	This service ...	Running	Automatic	Local Service
Office Source Engine	Saves install...		Manual	Local Syste...

Extended Standard

Actions

- Services
 - More Actions
- MyNewService
 - More Actions

Computer Management

File Action View Help

Computer Management (Local)

- System Tools
 - Task Scheduler
 - Event Viewer
 - Custom Views
 - Windows Logs
 - Applications and Services Logs
 - ACEEventLog
 - Hardware Events
 - Internet Explorer
 - Key Management Service
 - Media Center
 - Microsoft
 - Microsoft Office Diagnostics
 - Microsoft Office Search
 - Microsoft-SQLServer
 - Microsoft-SQLServer
 - MyNewLog
 - PreEmptive
 - Windows Azure
 - Windows PowerShell
 - Subscriptions
 - Shared Folders
 - Local Users and Groups
 - Performance
 - Device Manager
 - Storage
 - Disk Management
 - Services and Applications
 - Services
 - WMI Control
 - SQL Server Configuration Tools

Level	Date and Time	Source	Event ID	Task Category
Information	4/4/2014 10:20:20 PM	MySource	0	None
Information	4/4/2014 10:08:49 PM	MySource	0	None
Information	3/30/2014 2:44:30 PM	MySource	0	None
Information	3/30/2014 2:44:26 PM	MySource	0	None

Event 0, MySource

General Details

In OnStart!

Log Name: MyNewLog
Source: MySource
Event ID: 0
Level: Information
User: N/A
OpCode:
More Information: [Event Log Online Help](#)

Logged: 3/30/2014 2:44:26 PM
Task Category: None
Keywords: Classic
Computer: OwenGamePC

Actions

MyNewLog

- Open Saved Log...
- Create Custom View...
- Import Custom View...
- Clear Log...
- Filter Current Log...
- Properties
- Find...
- Save All Events As...
- Attach a Task To this L...
- View
- Refresh
- Help

Event 0, MySource

- Event Properties
- Attach Task To This Ev...
- Copy
- Save Selected Events...
- Refresh
- Help

```
Administrator: Windows PowerShell
PS C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug> installutil.exe -u .\WalkThroughService.exe
Microsoft (R) .NET Framework Installation utility Version 4.0.30319.33440
Copyright (C) Microsoft Corporation. All rights reserved.

The uninstall is beginning.
See the contents of the log file for the C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.exe assembly's progress.
The file is located at C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.InstallLog.
Uninstalling assembly 'C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.exe'.
Affected parameters are:
    logtoconsole =
    logfile = C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.InstallLog
    assemblypath = C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug\WalkThroughService.exe
Removing EventLog source MyNewService.
Service MyNewService is being removed from the system...
Service MyNewService was successfully removed from the system.

The uninstall has completed.
PS C:\Users\The Boss\Documents\Visual Studio 2012\Projects\WalkThroughService\WalkThroughService\bin\Debug>
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.ServiceProcess;
using System.Text;
using System.Threading.Tasks;

namespace WalkThroughService
{
    //Deriving from ServiceBase allows us to write services.
    public partial class MyNewService : ServiceBase
    {
        public MyNewService()
        {
            InitializeComponent();
            //Make sure that the event log exists, but don't try to recreate it
            //if it does.
            if (!EventLog.SourceExists("MySource")){
                EventLog.CreateEventSource("MySource",
                    "MyNewLog");
            }
            //Attach the event log that we have to the one that was created
            //above
            eventLog1.Source = "MySource";
            eventLog1.Log = "MyNewLog";
        }

        //This method runs when your service starts, and accepts command line
        //parameters.
        protected override void OnStart(string[] args)
        {
            //Write an entry to the event log.
            eventLog1.WriteEntry("In OnStart!");
        }

        //This method runs when the service is stopped for any reason.
        protected override void OnStop()
        {
            eventLog1.WriteEntry("In OnStop!");
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.ServiceProcess;
using System.Text;
using System.Threading.Tasks;

namespace WalkThroughService
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        static void Main()
        {
            ServiceBase[] ServicesToRun;
            ServicesToRun = new ServiceBase[]
            {
                new MyNewService()
            };
            ServiceBase.Run(ServicesToRun);
        }
    }
}
```