

*Steps for creating a Synergy Service in Visual Studio

- 1) Open Visual Studio (Visual Studio 2015 was used in making this tutorial) and create a new Synergy Window Service project

File -> New -> Project -> Synergy/DE -> Windows Service

In this case, the Project is named: SynNETService and the Location is: C:\Synergy.NETService

- 2) Replace all instances of the default Service1 with the desired name for your service. Either select:

Edit -> Find and Replace -> Replace in Files

or

Hit Ctrl + Shift + H

Then replace all instances of Service1 with your service name

Right click on the file Service1.dbl in Solution Explorer, choose Rename and then rename it to your service name.

In this case, replace Service1 with SynergyService

- 3) Add the code you wish executed on service start up on in the OnStart method and code to execute on service stop in the OnStop method

In this case, code to create a file (SynServiceFile.txt) in the % windir%\Temp directory and write start and stop entries to that file

- 4) In the constructor add code to create a timer and use this timer to call a method on a timed interval.

For example -

```
;Below code will create a timer which calls the OnTimer method every 60 seconds  
data timer                ,@System.Timers.Timer                ,new System.Timers.Timer()
```

```
timer.Interval = 60000                ;60 * 1000 milliseconds = 60 seconds  
timer.Elapsed += new System.Timers.ElapsedEventHandler(this.OnTimer)  
timer.Start()
```

- 5) Next add code to the constructor to set the Source and Log properties of the variable based on the EventLog object.

For example,

```
MyEventLog = new System.Diagnostics.EventLog()
```

```
data event_source        ,string                , "SynergyService"
```

```

;Check to make sure there isn't another event with the SynergyService name
if (!System.Diagnostics.EventLog.SourceExists(event_source))
begin
;By passing an empty string for the Log parameter the event will be logged as an
Application event
System.Diagnostics.EventLog.CreateEventSource(event_source,string.Empty)
end

```

MyEventLog.Source = event_source

- 6) Create the method specified in the timer's ElapsedEventHandler delegate with the signature below:

```

public method OnTimer, void
sender ,@object
args ,@System.Timers.ElapsedEventArgs
endparams

```

- 7) Add a class level variables for file channel, %DATETIME return result and to write entries into the Windows Event Viewer log.

For example,

```

chan          ,int
dt            ,a20
MyEventLog    ,@EventLog ,new EventLog()

```

- 8) Write any relevant entries to the Windows Event Viewer log with the variable based on the EventLog class's WriteEntry method

For example,

In OnStart method

```
MyEventLog.WriteEntry("SynNETService started")
```

- 9) Add an Installer Project to Solution

In the Solution Explorer right-click on the Service file (in this case SynergyService.dbl) and choose View Designer or hit Shift+F7

Right-click in the designer window and select Add Installer. This will add two installers to the project. One for the service and one for the service's associated process.

Select serviceInstaller1 and in the Properties window edit the following -

Name - to be the desired name of your service. In this case SynergyService.

Description - to be a brief description of your service. Note, this description will appear in the Services window and will help others identify the purpose of your service.

DisplayName - to be the text that appears in the Services window's Name column. Note, this name may be different than the service Name which is used by the system to interact with the service.

StartType - Set to Automatic (so the service starts automatically with the system.

Select serviceProcessInstaller1 in the designer window and in the Properties window set the **Account** property to LocalSystem.

10) In the Solution Explorer right-click on the service project and select Properties.

In the Application tab set "Startup object:" to the name of service project
In this case **SynNETService**

11) Build the service project with Build -> service project name

In this case Build -> Build SynNETService or Shift+F6

12) Install the service.

Open a Visual Studio command prompt as Administrator (installing the service will not be successful unless the command prompt is opened as Administrator)
Navigate to the folder with the service executable and run InstallUtil.exe passing the service executable -

installutil.exe SynNetService.exe

The output of InstallUtil.exe will be similar to the below -

```
C:\Synergy.NETService\SynNETService\bin\Debug>installutil SynNetService.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:\Synergy.NETService\SynNETService\bin\Debug\SynNetService.exe assembly's progress.

The file is located at C:\Synergy.NETService\SynNETService\bin\Debug\SynNetService.InstallLog.

Installing assembly 'C:\Synergy.NETService\SynNETService\bin\Debug\SynNetService.exe'.

Affected parameters are:

```
logtoconsole =
logfile = C:\Synergy.NETService\SynNETService\bin\Debug\SynNetService.InstallLog
assemblypath = C:\Synergy.NETService\SynNETService\bin\Debug\SynNetService.exe
```

Installing service SynergyService...

Service SynergyService has been successfully installed.

Creating EventLog source SynergyService in log Application...

The Install phase completed successfully, and the Commit phase is beginning.

See the contents of the log file for the C:\Synergy.NETService\SynNETService\bin\Debug\SynNetService.exe assembly's progress.

The file is located at C:\Synergy.NETService\SynNETService\bin\Debug\SynNetService.InstallLog.

Committing assembly 'C:\Synergy.NETService\SynNETService\bin\Debug\SynNetService.exe'.

Affected parameters are:

```
logtoconsole =
logfile = C:\Synergy.NETService\SynNETService\bin\Debug\SynNetService.InstallLog
assemblypath = C:\Synergy.NETService\SynNETService\bin\Debug\SynNetService.exe
```

The Commit phase completed successfully.

The transacted install has completed.

Now that the service is installed it may be started with the Service Window (Control Panel -> Administrative Tools -> Services) or in an Administrator command prompt by running sc.exe.

With the Services Window -

Locate the service (in this case SynergyService)
Double-click and click the Start button

Using sc.exe - (NOTE: sc.exe is the executable that controls the Windows Service Controller)

Open an Administrator command prompt

Navigate to the directory with the service executable

Execute the below

```
sc SynergyService.exe
```

The output of sc SynergyService.exe will be -

```
SERVICE_NAME: SynergyService
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE                : 2  START_PENDING
                           (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE       : 0  (0x0)
        SERVICE_EXIT_CODE    : 0  (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINT            : 0x7d0
        PID                  : 9160
        FLAGS                 :
```

Note the file created by the service will be in Windows\Temp and not the user's temp directory as the service is run under LocalSystem

*NOTE: these steps based on the below Microsoft MSDN page –

“Walkthrough: Creating a Windows Service Application in the Component Designer”

<https://msdn.microsoft.com/en-us/library/zt39148a%28v=vs.110%29.aspx>