

# Visual Studio

## Copyright Notice

SFL Services LLC has prepared this document for use only by their staff, agents, customers and prospective customers. Companies, names and data used as examples in this document are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of SFL Services LLC, who reserve the right to change specifications and other information contained herein without prior notice. The reader should consult SFL Services LLC to determine whether any such changes have been made.

## Licensing and Warranty

The terms and conditions governing the licensing of SFL Services LLC software consist solely of those set forth in the written contracts between SFL Services LLC and its customers. Except as expressly provided for in the warranty provisions of those written contracts, no representation or other affirmation of fact contained in this document, including but not limited to statements regarding capacity, suitability for use or performance of products described herein, shall be deemed to be a warranty by SFL Services LLC for any purpose, or give rise to any liability of SFL Services LLC whatsoever.

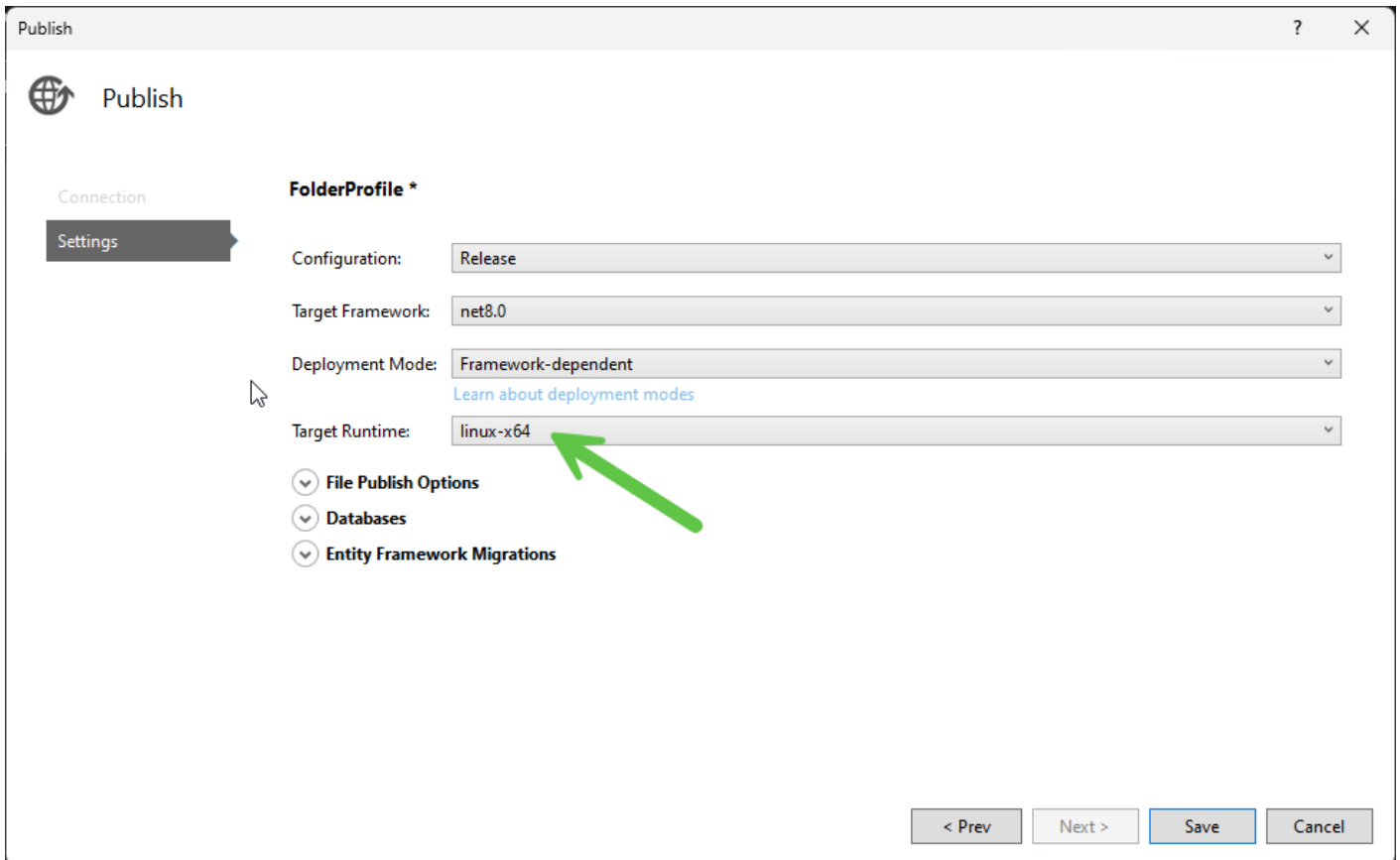
## Liability

In no event shall SFL Services LLC be liable for any incidental, indirect, special or consequential damages whatsoever (including but not limited to lost profits) arising out of or related to this document or the information contained in it, even if SFL Services LLC had been advised, knew or should have known of the possibility of such damages, and even if they had acted negligently.

- [Visual Studio - Deploy to Linux](#)
- [Visual Studio - Entity Framework Tips](#)
- [Visual Studio - Bag of Tricks](#)
- [Service - Add Installer and Setup](#)
- [Log4Net - Wrapper](#)
- [DataGrid - Working with Cells and Rows](#)

# Visual Studio - Deploy to Linux

The only thing that you need to do to deploy to Linux is to change the target runtime setting.



This allows for the deployment to add the specific files to the folder.

# Visual Studio - Entity Framework Tips

```
dotnet tool install --global dotnet-ef  
dotnet ef migrations add CreateInitial  
dotnet ef database update
```

[https://www.youtube.com/watch?v=Fbf\\_ua2t6v4](https://www.youtube.com/watch?v=Fbf_ua2t6v4)

## Scaffold

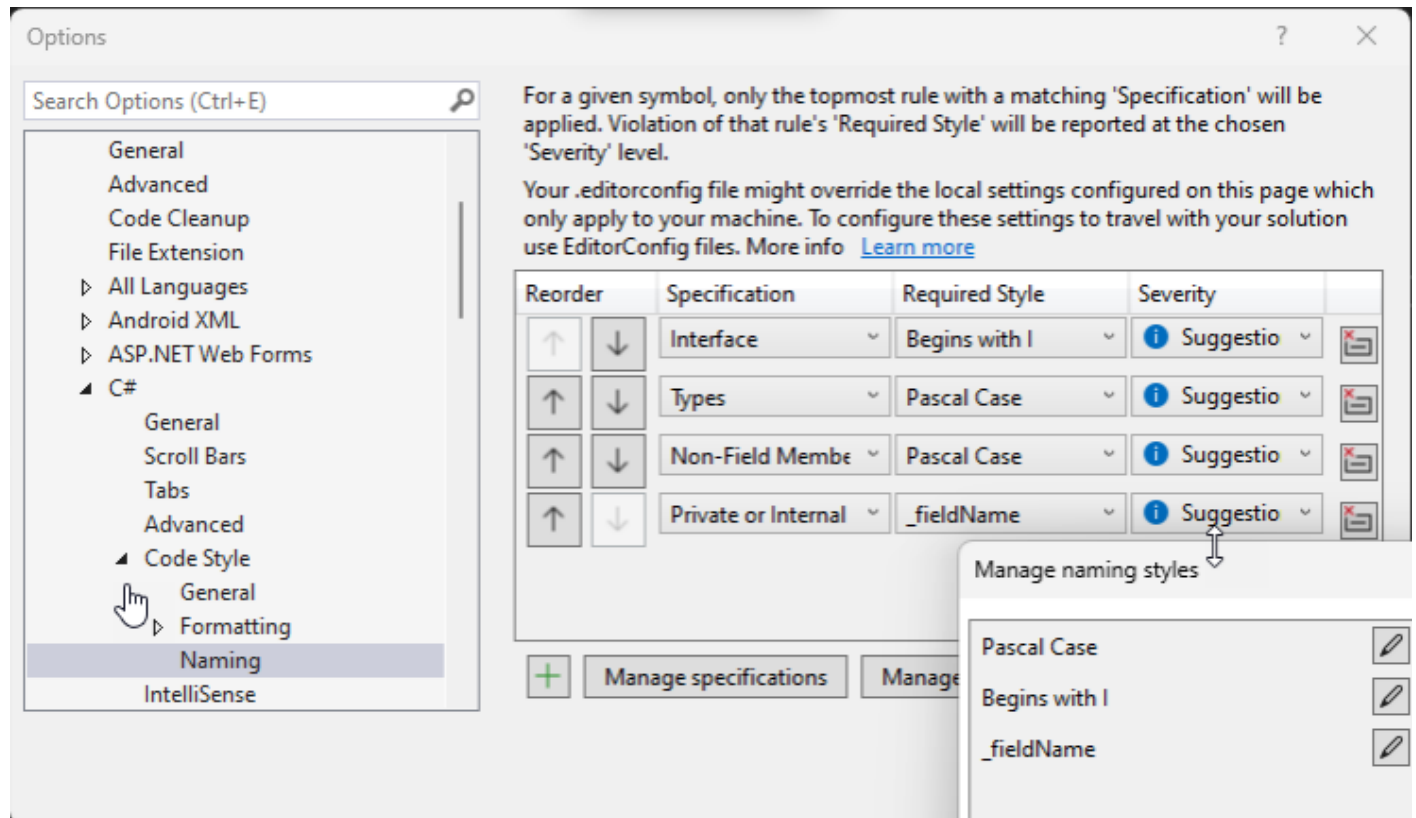
```
dotnet ef dbcontext scaffold "Data Source=mfb-us-sql-001;Initial  
Catalog=MyFFLBookAPI;TrustServerCertificate=True;User Id=XXXXXX;Password=XXXXXX"  
Microsoft.EntityFrameworkCore.SqlServer -c DataContext --context-dir Data -o Data -f
```

## Redo

<https://stackoverflow.com/questions/36741793/ef-7-migration-to-existing-database>

# Visual Studio - Bag of Tricks

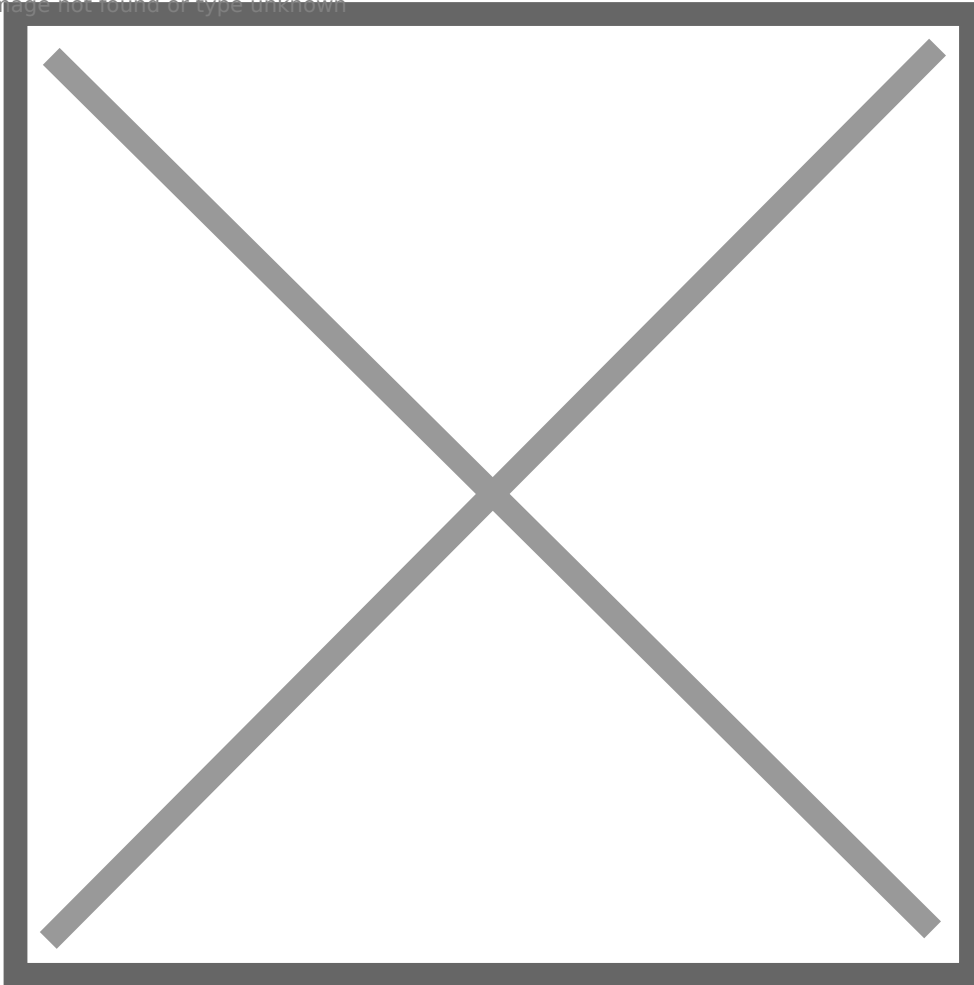
To Add underscore to fieldNames



# Service - Add Installer and Setup

Go to Solution Explorer and right-click on your project solution then go to the view designer then look at the design view then right-click on the design page and click on “add installer” and it will look like this:

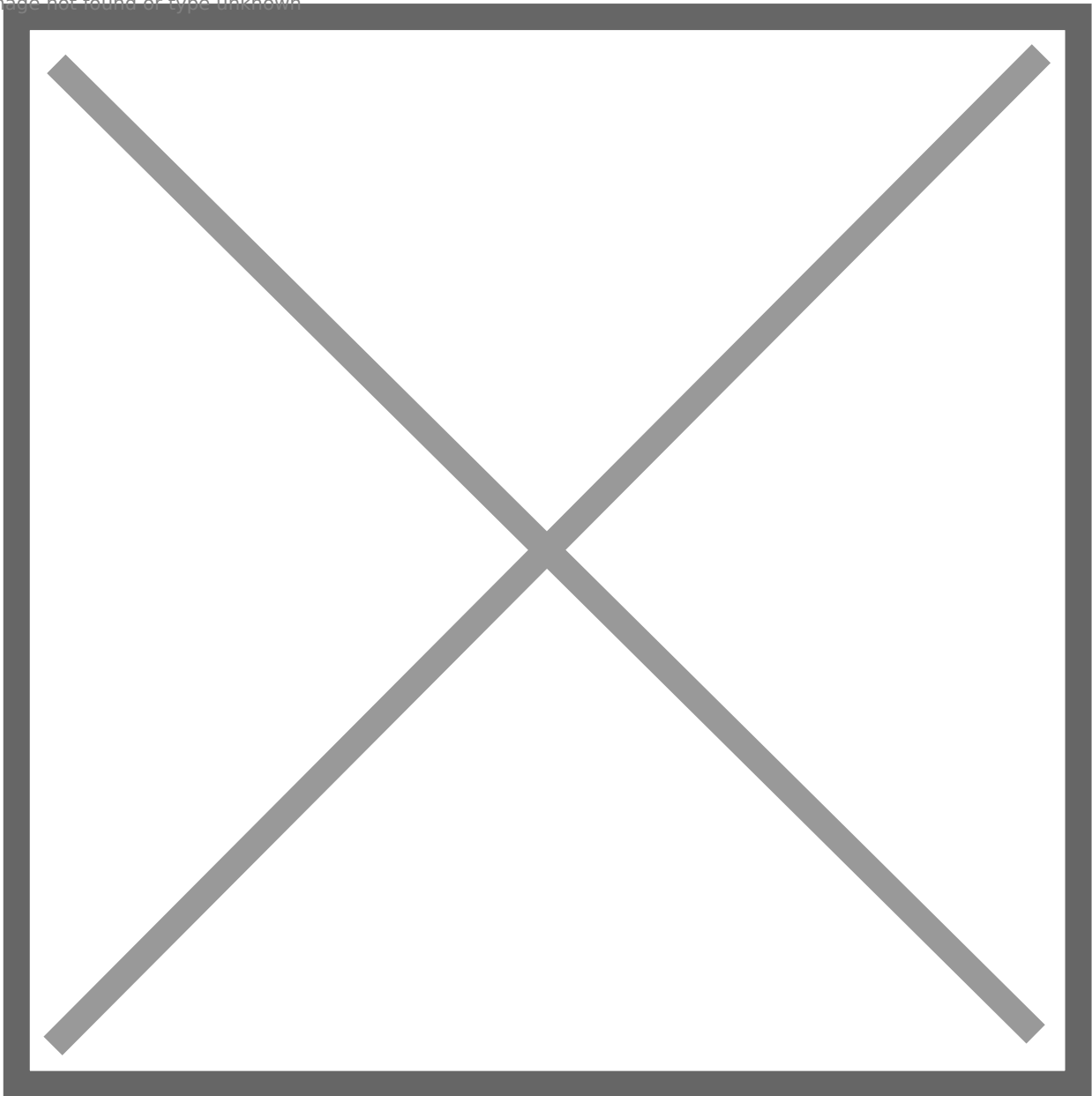
Image not found or type unknown



After adding the ProjectInstaller.cs you have 2 components in the design view of the ProjectInstaller.cs (serviceProcessInstaller1 and serviceInstaller1). You should then set up the properties as you need.

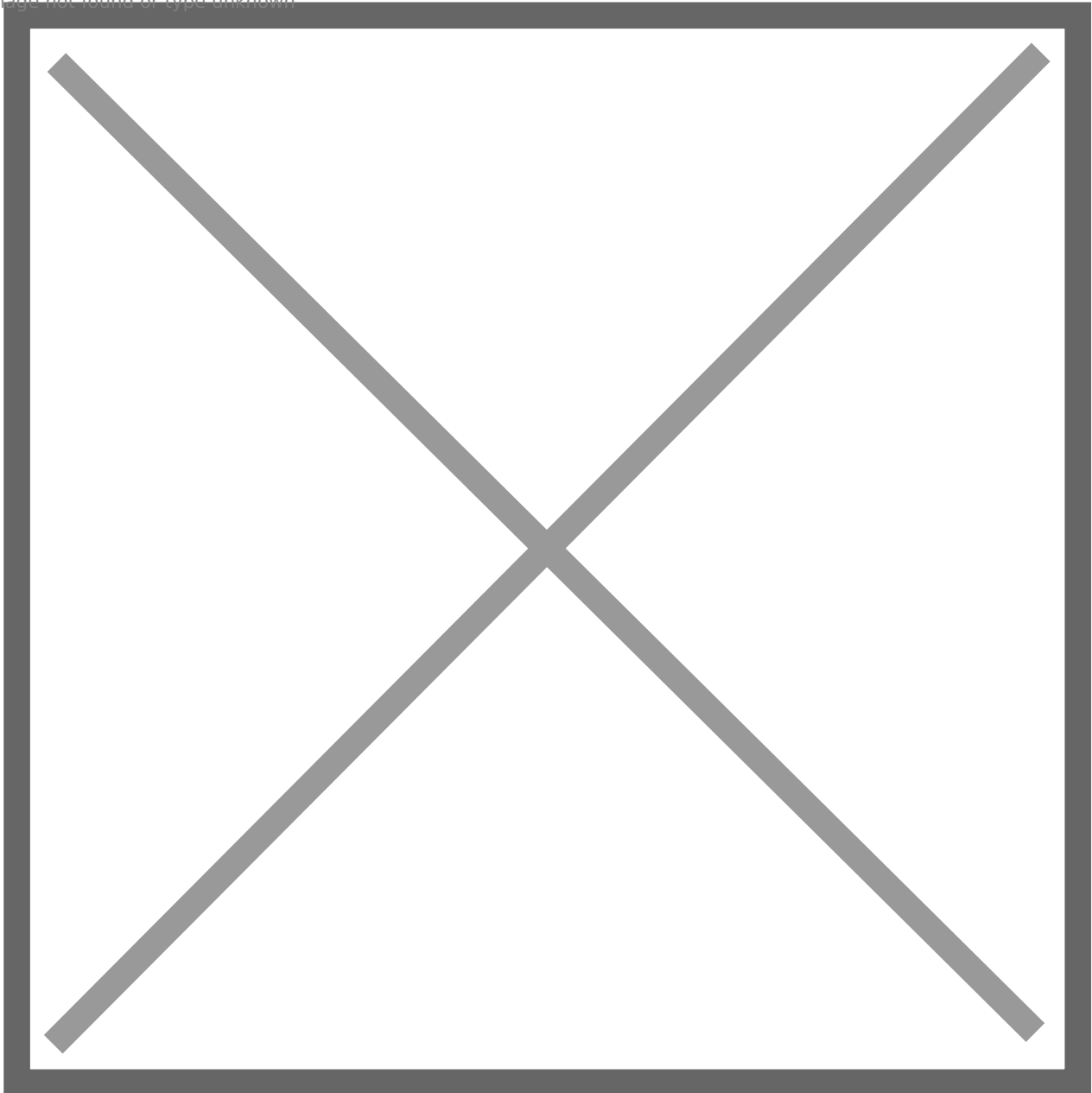
Click on serviceinstaller1 and go to the properties in the right pane and edit the service name Service1 to your project solution as in the following:

Image not found or type unknown

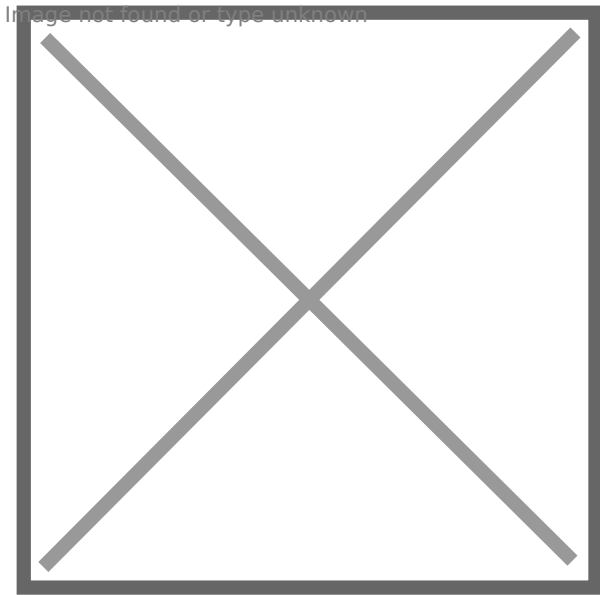


Click on serviceProcessInstaller1 and go to the properties in the right pane and select Account and choose LocalService from the dropdown and save. See the following:

Image not found or type unknown



You have two assemblies under References as in the following highlighted (1) System.Configuration.Install and (2) System.ServiceProcess.



## **Create Setup Project**

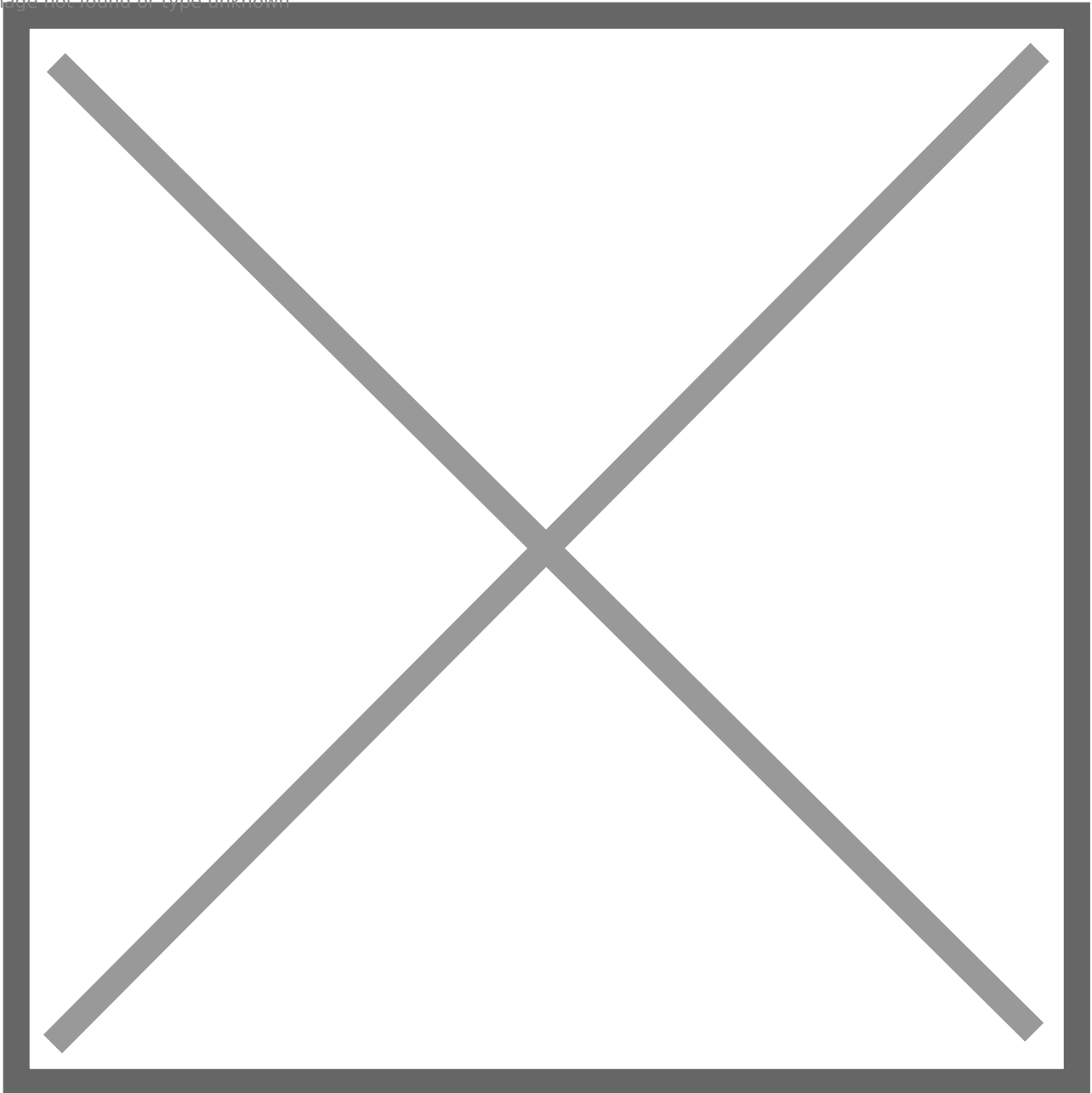
Build your service project, you need to set up the project to install the build/compiled project and run the installers to run the Windows service. Create a new project as a “Setup Project” and you need to add project output.

## **Create Setup Project for Window Service**

Go to the Solution Explorer and right-click on the solution, go to Add > New Project.

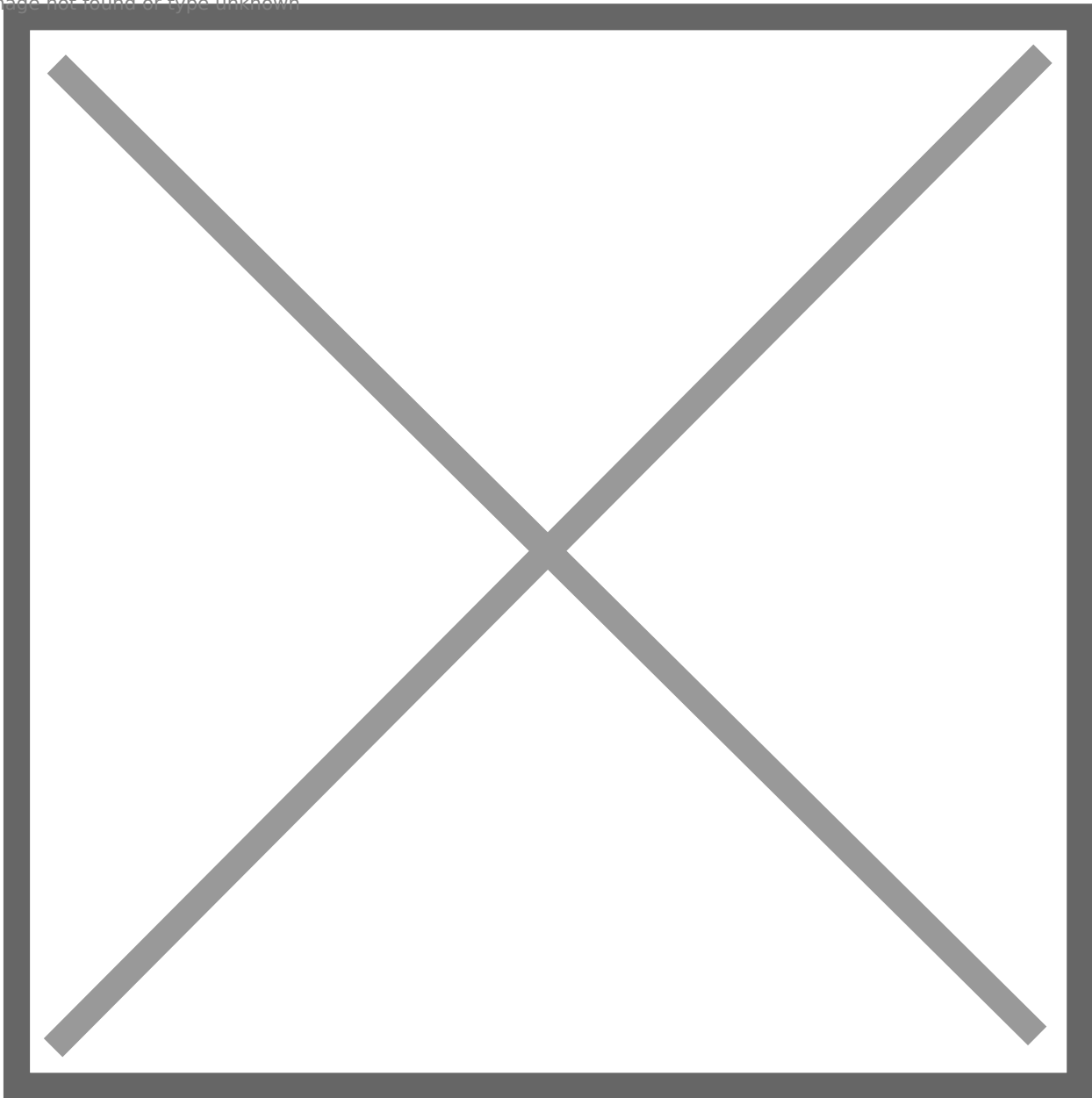


Image not found or type unknown



Open a dialog box, go to left pane under Installed Templates > Other Project Types > Setup and Deployment > Visual Studio Installer and go to the right pane and select the project as a “Setup Project” and click on the OK button.

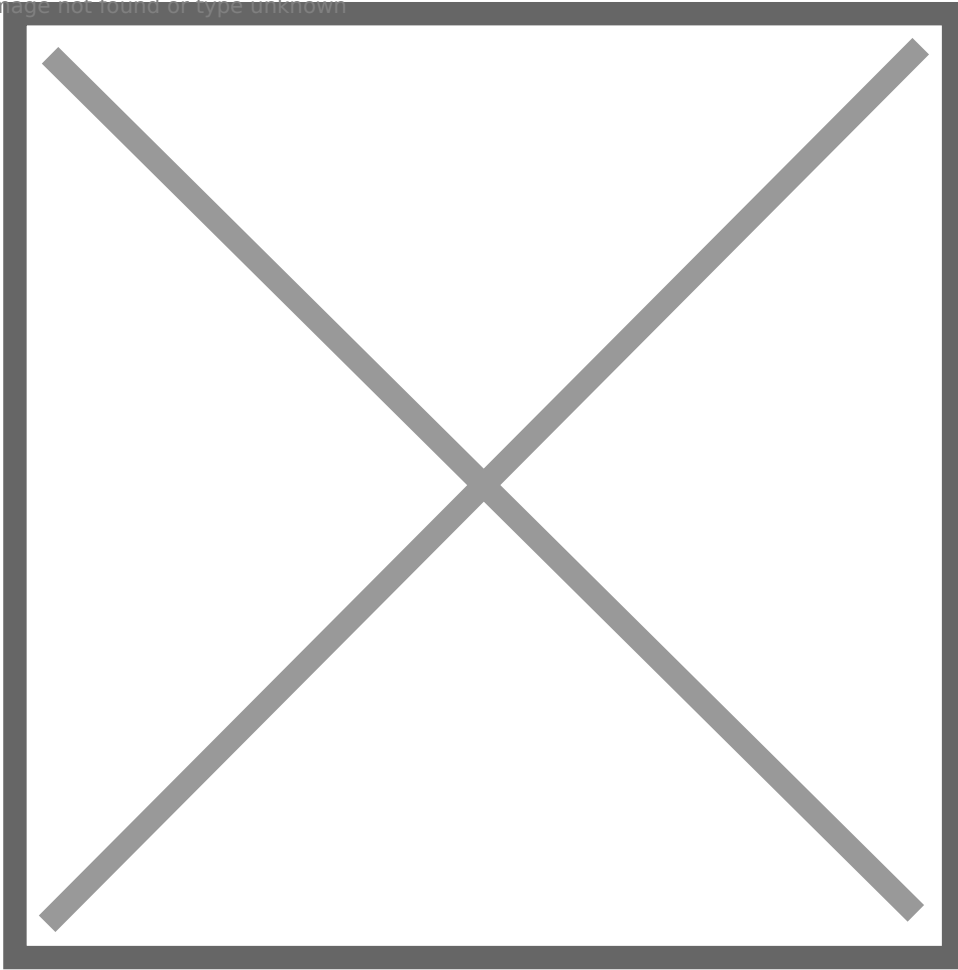
Image not found or type unknown



### **Custom actions add to setup project**

Go to the solution, right-click on the setup project then select View > Custom action.

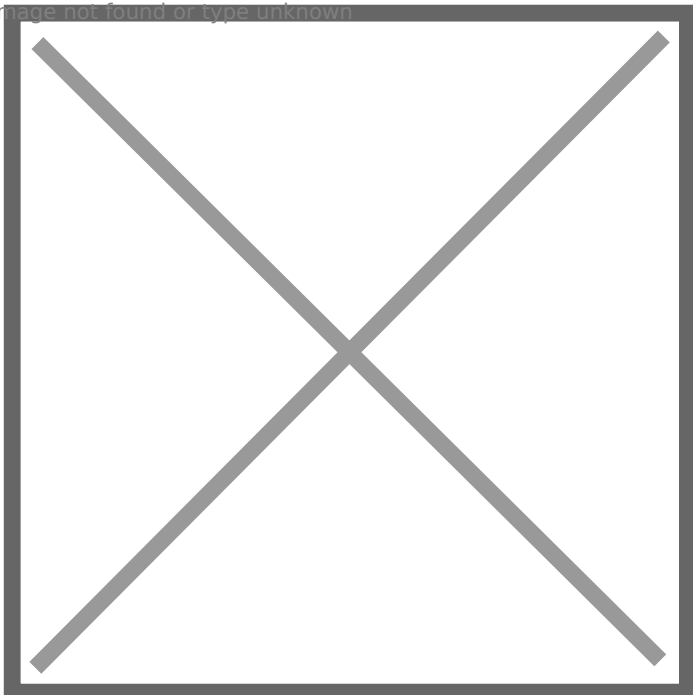
Image not found or type unknown



The “Custom action” editor appears.

Right-click on “Custom action > Add Custom action” and add the "Primary Output" from (Active)" to every custom action

Image not found or type unknown



Open a dialog box “Select Item In Project” then double-click on “Application Folder” then click on the “Add Output” button. Open a dialog box, choose your project (Window service) and “Primary Output” from (Active) and click on the OK button.

Image not found or type unknown

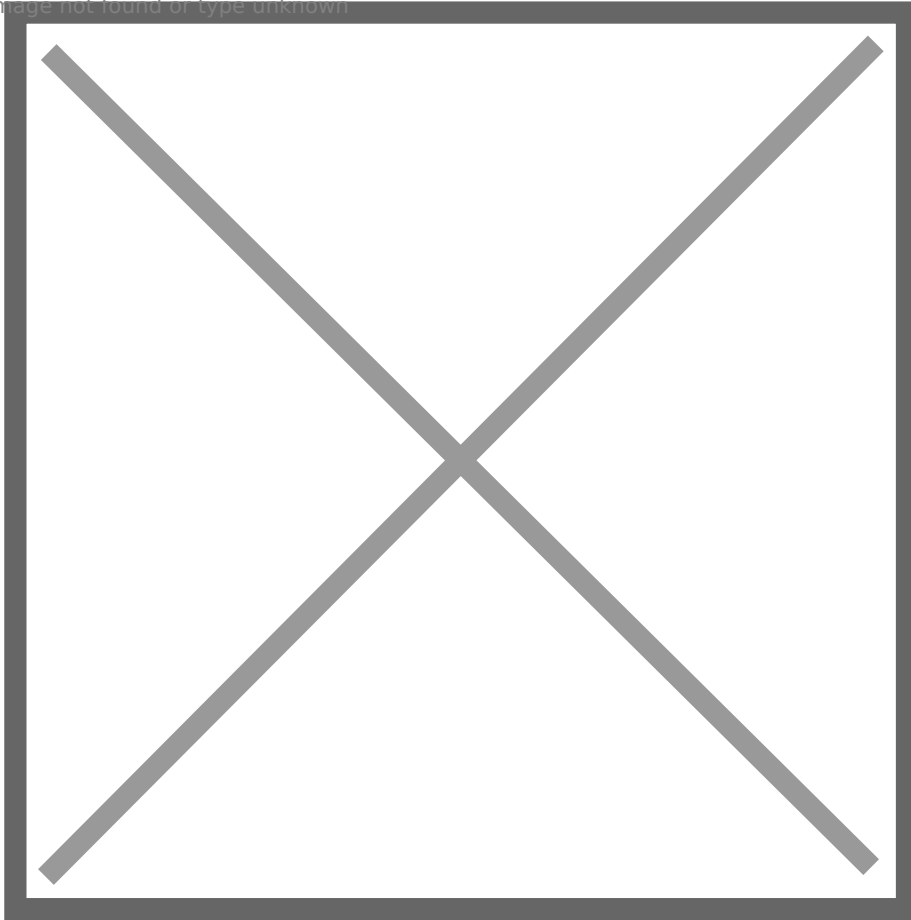


Image not found or type unknown

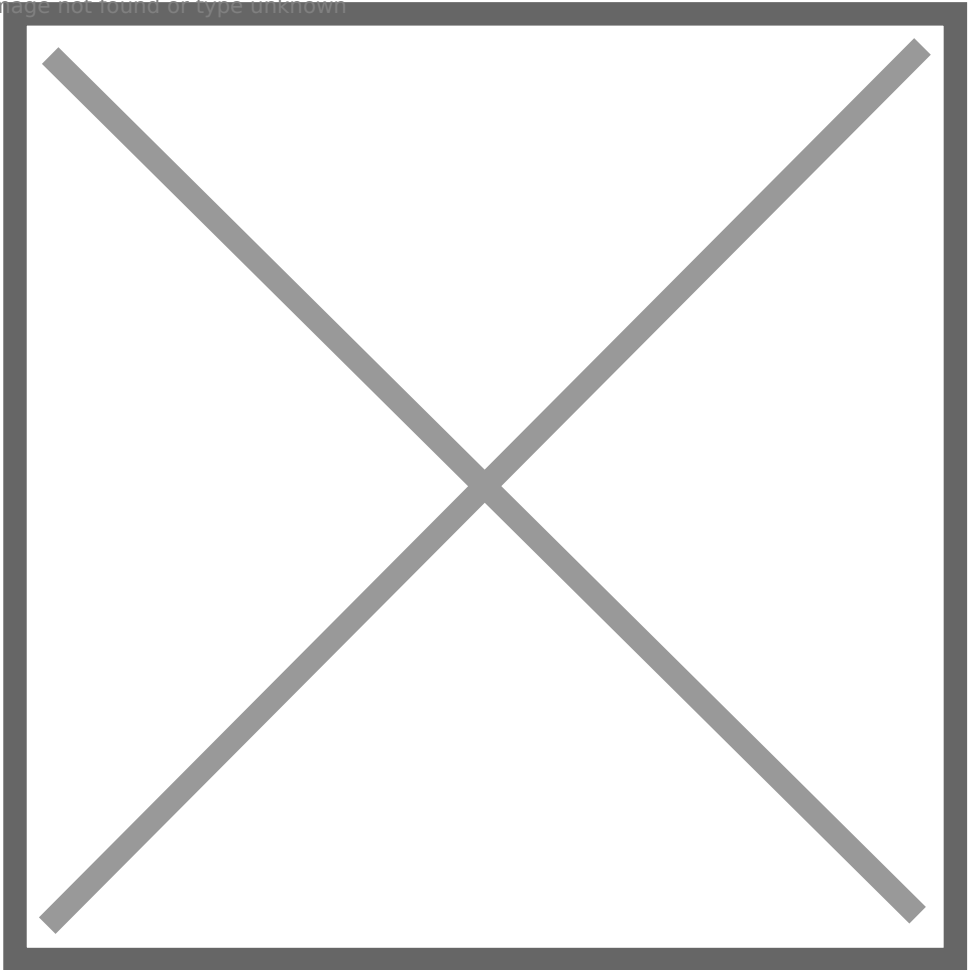
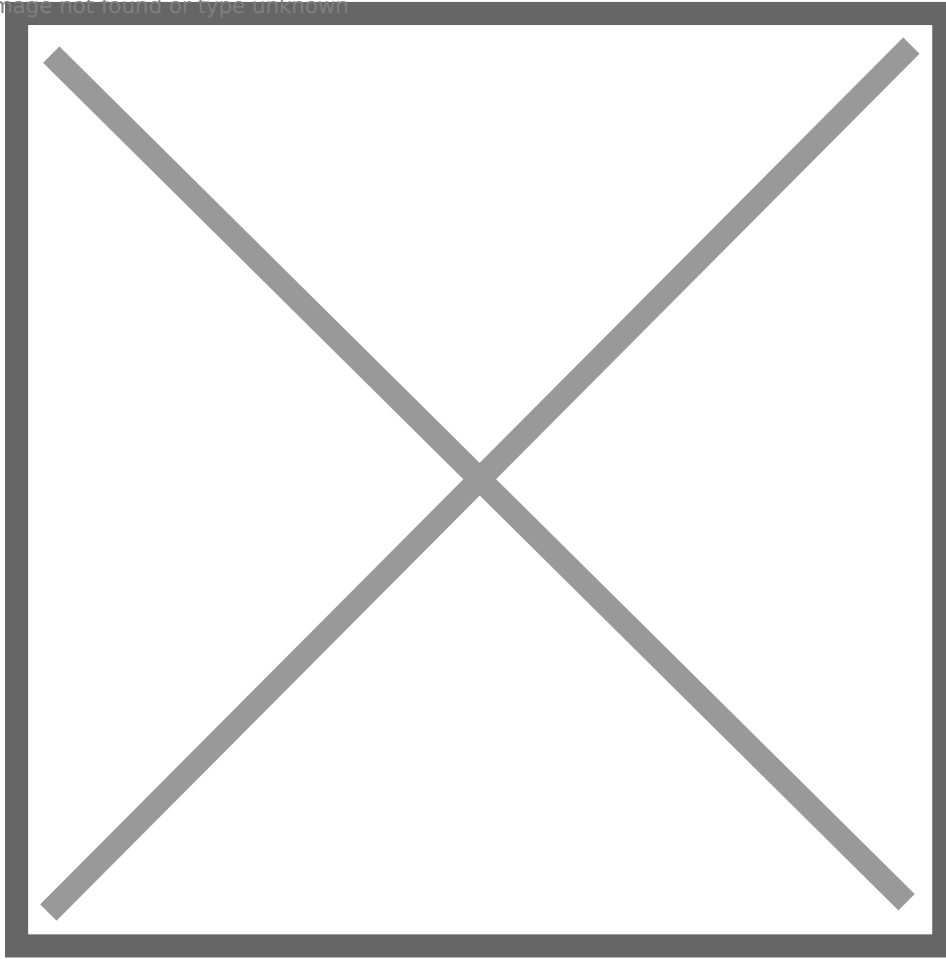


Image not found or type unknown



Build your setup project and install the Windows service.

### **Install And Start The Service**

Go to the Solution Explorer then select Build for both projects (Windows Service and Setup Project) then right-click on the Setup Project and click on the Install option and follow the setup wizard procedure. Finally your service is ready for being started and stopped. If you want to start the Windows server then go to Start > Administrator tools > Service > Find Your Service then right-click on the Service Name then select Start. Otherwise go to Start > Computer (My Computer) > Manage > Click on Service And Application from the left pane > Services.

# Log4Net - Wrapper

<https://stackoverflow.com/questions/1028375/how-do-you-configure-and-enable-log4net-for-a-stand-alone-class-library-assembly>

# DataGrid - Working with Cells and Rows

```
void dataGridView_CellValidating(object sender, DataGridViewCellValidatingEventArgs e)
{
    // Ignore cell if it's not dirty
    if (dataGridView.IsCurrentCellDirty)
        return;

    // Validate current cell.
}

void dataGridView_RowValidating(object sender, DataGridViewCellCancelEventArgs e)
{
    // Ignore Row if it's not dirty
    if (!dataGridView.IsCurrentRowDirty)
        return;

    // Validate all cells in the current row.
}

void dataGridView_CellEndEdit(object sender, DataGridViewCellEventArgs e)
{
    // Validate all cells in the current row and return if any are invalid.

    // If they are valid, save changes to the database

    // This is when I would expect dataGridView.IsCurrentRowDirty to be false.
    // When this row loses focus it will trigger RowValidating and validate all
    // cells in this row, which we already did above.
}
```

<https://stackoverflow.com/questions/2043210/datagridview-row-is-still-dirty-after-committing-changes>



<https://stackoverflow.com/questions/35977042/how-to-filter-data-using-entity-framework-in-a-way-that-datagridview-be-editable/35978408#35978408>