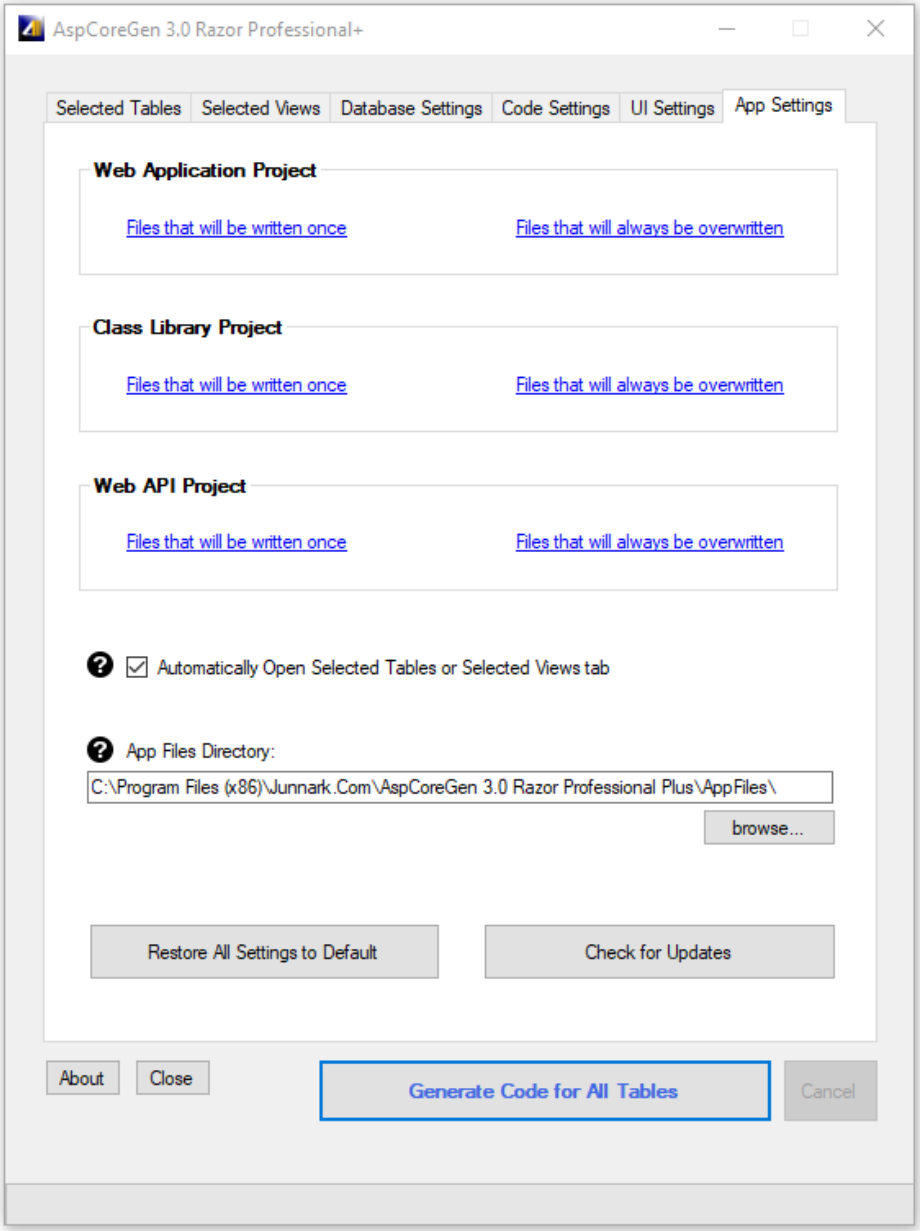
App Settings Tab

1 CONTENTS

App Settings Tab

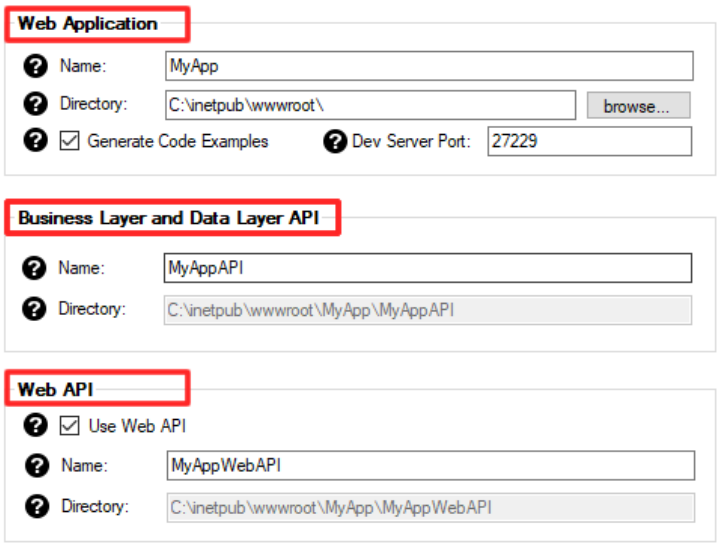
The App Settings Tab is where we set AspCoreGen 3.0 Razor application-specific information such as the directory where the *App Files* are located. Whether to open the *Selected Tables* or *Selected Views* tab when the respective option is chosen in the *Code Settings* tab. You can also *Restore All Settings to Default* from here with a click of a button. And most importantly, it shows in a list the generated *Files* that you can customize per Project and the generated *Files* that you should not add code to.



1 GENERATED PROJECTS

These are the *Generated Projects* as shown in the *Code Settings Tab*.

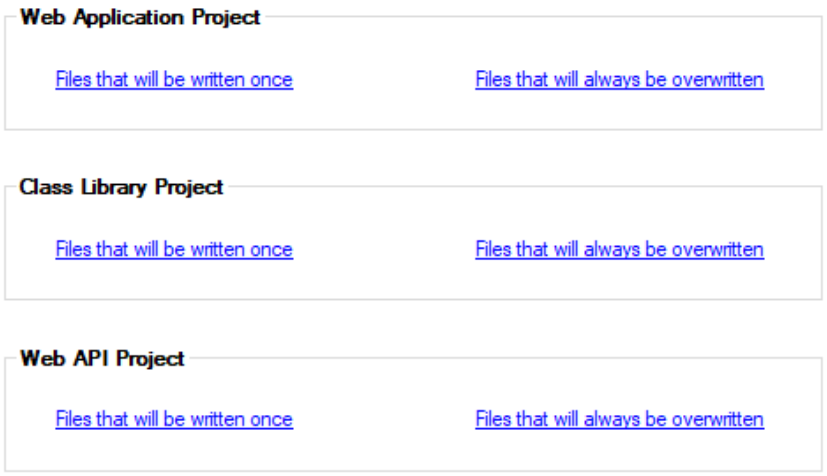
- Web Application Project (Front End)
- Business Layer and Data Layer API Project (Class Library Project – Middle and Data Tier)
- Web API Project (Optional)



Code Settings Tab – Generated Projects

Each application/project is comprised of generated *Files*. You can add your own code in some of the generated *Files* per project, and when you generate code over the same projects again these *Files* will NOT be overwritten. There are also *Files* that you should not add your own code to.

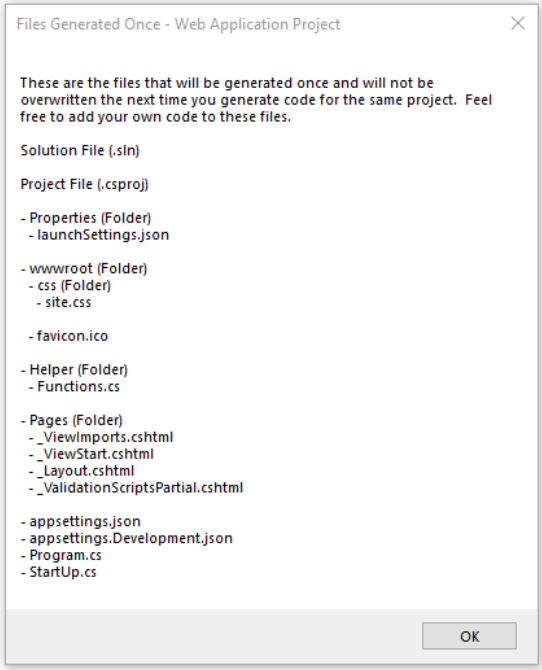
- **Files that will be written once:** You can add your own custom here.
- **Files that will always be overwritten:** Do not add your code here, it will be overwritten.

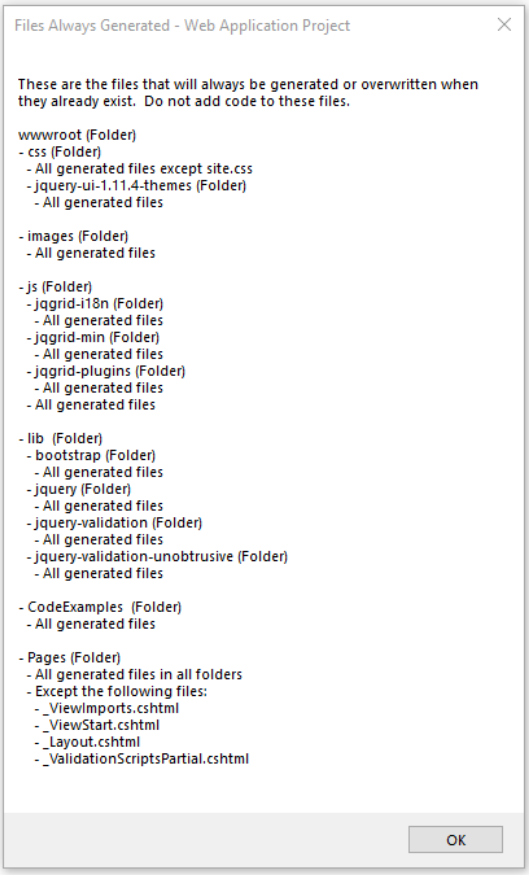


App Settings Tab – Files Written Once, Files Always Overwritten

1.1 WEB APPLICATION PROJECT

There are 2 links under the *Web Application Project*. A pop-up box pops-up when you click the respective link.





1.1.1 Files That Will Be Written Once

- *Solution File (.sln)*
- *Project File (.csproj)*
- Properties (Folder)
  - *launchSettings.json*
- wwwroot (Folder)
  - css (Folder)
    - *site.css*
  - *favicon.ico*
- Controllers (Folder)
  - *All Generated Child Classes*
- Helper (Folder)
  - *Functions.cs*
- Pages (Folder)
  - *_ViewImports.cshtml*
  - *_ViewStart.cshtml*
  - *_Layout.cshtml*
  - *_ValidationScriptsPartial.cshtml*

- *appsettings.json*
- *appsettings.Development.json*
- *Program.cs*
- *StartUp.cs*

1.1.2 Files That Will Always Be Overwritten

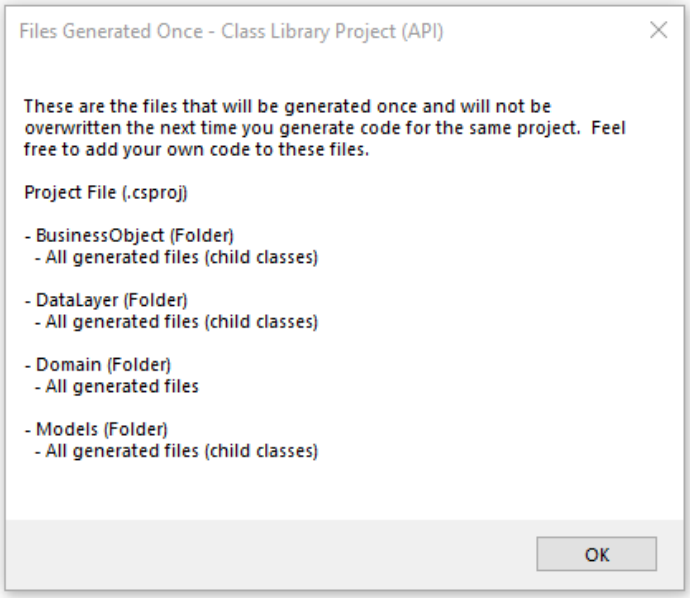
- wwwroot (Folder)
  - css (Folder)
    - *All generated files* **except site.css**
  - jquery-ui-1.11.4-themes (Folder)
    - *All generated files*
  - images (Folder)
    - *All generated files*
  - js (Folder)
    - jqgrid-i18n (Folder)
      - *All generated files*
    - jqgrid-min (Folder)
      - *All generated files*
    - jqgrid-plugins (Folder)
      - *All generated files*
    - *All generated files*
  - lib (Folder)
    - bootstrap (Folder)
      - *All generated files*
    - jquery (Folder)
      - *All generated files*
    - jquery-validation (Folder)
      - *All generated files*
    - jquery-validation-unobtrusive (Folder)
      - *All generated files*
- CodeExamples (Folder)
  - *All generated files*
- Pages (Folder)
  - *All generated files in all folders*
  - **Except the following files:**
    - _ViewImports.cshtml
    - _ViewStart.cshtml
    - _Layout.cshtml
    - _ValidationScriptsPartial.cshtml

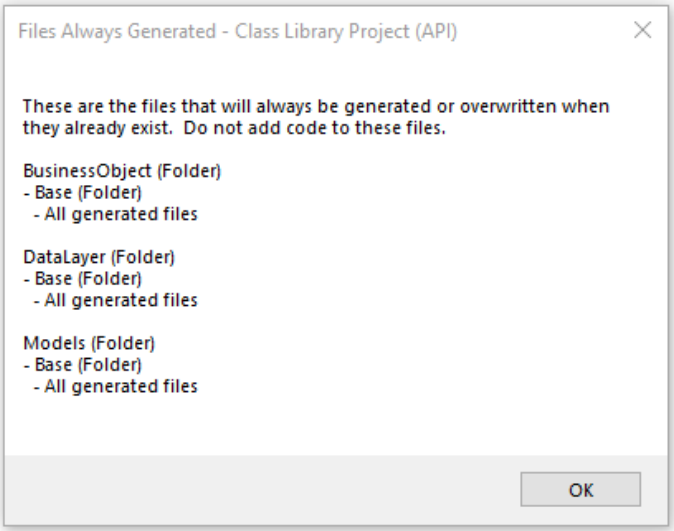
1.2 CLASS LIBRARY PROJECT

Note: *Child Classes* are class files that inherit from a *Parent (Base) Class*. For example, there is a *Base* folder in the *BusinessObject* folder. All of the classes in the *Base* folder are considered to be *Parent (Base) Classes* because each one those *Class* is **inherited by** the respective *Child Class* located in the *BusinessObject* folder.

E.g. *Products**Base**.cs* is a *Parent (Base) Class*. *Products.cs* is a *Child Class* because it inherits from the *Products**Base**.cs.*

There are 2 links under the *Class Library Project*. A pop-up box pops-up when you click the respective link.





1.2.1 Files That Will Be Written Once

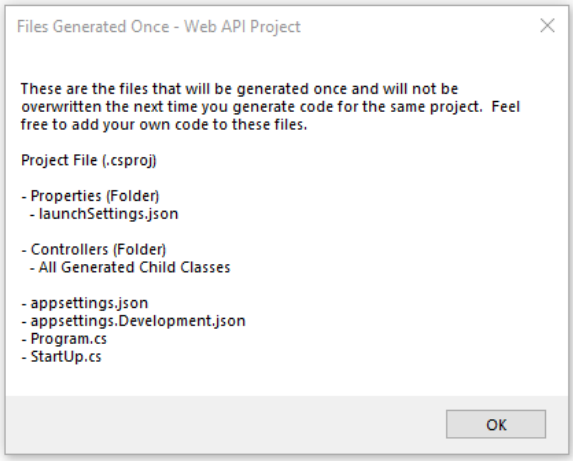
- *Project File (.csproj)*
- BusinessObject (Folder)
  - *All generated files (child classes)*
- DataLayer (Folder)
  - *All generated files (child classes)*
- Domain (Folder)
  - *All generated files*
- Models (Folder)
  - *All generated files (child classes)*

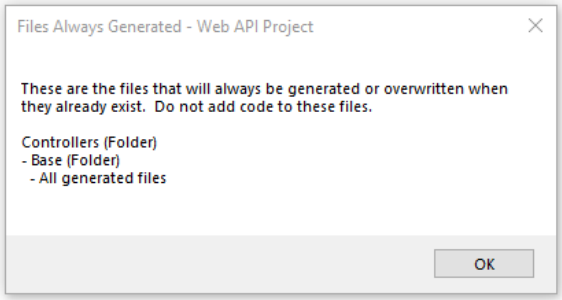
1.2.2 Files That Will Always Be Overwritten

- BusinessObject (Folder)
  - Base (Folder)
    - *All generated files*
- DataLayer (Folder)
  - Base (Folder)
    - *All generated files*
- Models (Folder)
  - Base (Folder)
    - *All generated files*

1.3 Web API Project

There are 2 links under the *Web API Project*. A pop-up box pops-up when you click the respective link.





1.3.1 Files That Will Be Written Once

- *Project File (.csproj)*
- Properties (Folder)
  - *launchSettings.json*
- Controllers (Folder)
  - *All Generated Child Classes*
- *appsettings.json*
- *appsettings.Development.json*
- *Program.cs*
- *StartUp.cs*

1.3.2 Files That Will Always Be Overwritten

- Controllers (Folder)
  - Base (Folder)
    - *All generated files*

1.4 WHY IS IT IMPORTANT TO KNOW WHICH FILES WILL BE WRITTEN ONCE OR ALWAYS OVERWRITTEN

Note: This part is not in the *App Settings Tab*.

So why do we have some files written just once while the others are always overwritten when we generate code for the same project?

1.4.1 Files That Will Be Written Once

For the files that are written once like the *launchSettings.json* for the *Web Application Project* and *Web API Project*, you might want to add your own code, for this instance you might want to add your own settings to the *launchSettings.json.* AspCoreGen 3.0 Razor will NOT overwrite this file if it already exist.

Another example is the *Solution File (.sln)*. You may want to add another project that is not generated by AspCoreGen 3.0 Razor to your solution, this will make sure that the project will load in the same solution even when you keep generating code for the same project over, and over again.

The same goes with the *Project File (.csproj)*, you will be able to add new Razor Pages, images, class files, even references, etc. and know that those you added will always be loaded even after regeneration.

You probably want to add your own *favicon.ico* and overwrite the one we added to the project by default. Or even add new styles to the *site.css* without fear of losing your code.

Another example is the *_Layout.cshtml*. You are probably going to want to add your own design to the overall web application. Here's your chance.

In the *Class Library Project* (Business Layer and Data Layer API project) you will see a lot of *All generated files (child classes).* For *Business Objects, Data Layer, Models,* etc. 2 classes are generated per database table. In the example (as explained in *Page 6*), a *Parent (Base) Class* and a *Child Class*.

For the *Products* database table 2 *Business Object* (*Middle-Tier*) classes are generated:

1. *Products**Base**.cs* is a *Parent (Base) Class,* most of the generated code is here. **This will always be overwritten.**
2. *Products.cs* is a *Child Class* because it inherits from the *Products**Base***.cs, mostly an empty class but have all the functionality of the *Products**Base**.cs Parent (Base) Class.* **This will be written once**. So you can add your custom code here (if you want to).

AspCoreGen 3.0 Razor always references *Child Classes* by default. E.g. ***Products***.InsertAsync(), ***Products***.UpdateAsycn(), ***Products***.SelectByPrimaryKeyAsycn(), etc. So if you add your own method like "YourOwnMethod()" in the *Products.cs (Child Class)*, it will now be accessible as *Products.* ***YourOwnMethod()***, and again, this method will not be overwritten when you regenerate code for the same project.
Note: You can also reference the *Parent Class* like this: *Products**Base***.InsertAsync(), *Products**Base.***SelectByPrimaryKeyAsycn(), but our suggestion is **Don't**. Stick to referencing the *Child Class*.

1.4.2 Files That Will Always Be Overwritten

Do not add your own code in any of these files.

An example would be one of the generated Razor Page, the *ListCrudRedirect.cshtml*. Maybe you want to add a new column or remove a column in the generated JQGrid. Or maybe you want no sorting, or *Add New Record* functionality only no *Editing* or *Deleting*. For whatever reason you may want to update the generated code for this Razor Page, **Don't**.

You should instead create a new Razor Page and add it to the *Web Application Project*. Copy all the code from the *ListCrudRedirect.cshtml*, and then do all your changes there.

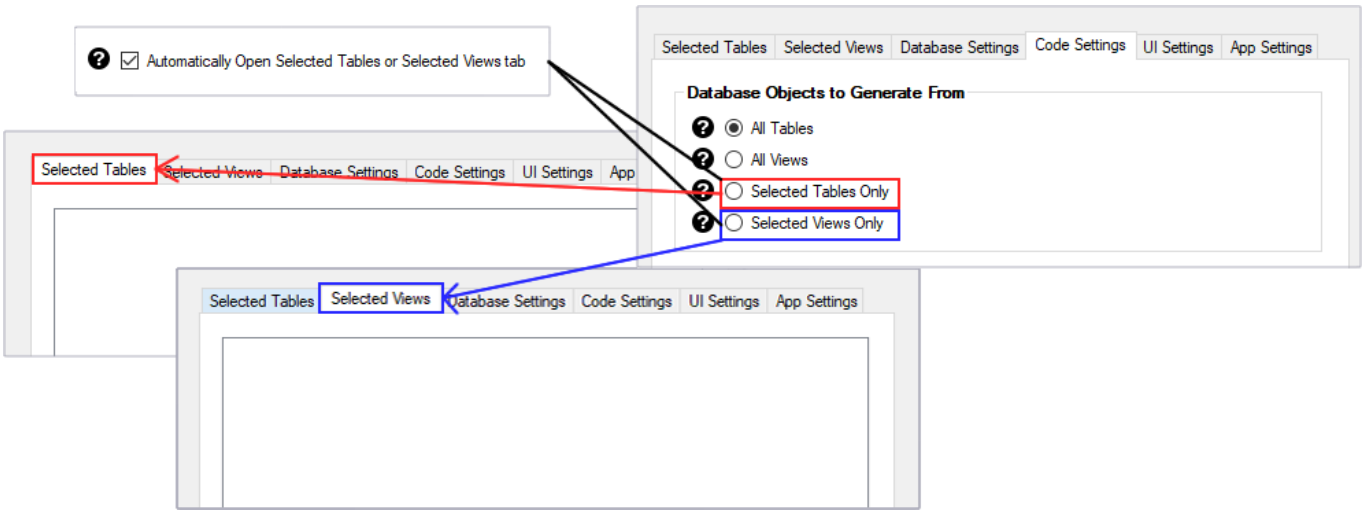
So you have *YourNewView.cshtml*, and it references a javacript file *jqgrid-listcrudredirect.js*, you want to add new javascript functions to it, **Don't**.

Instead, create a new javascript file *your-new-javascript.js*, copy all the code from *jqgrid-listcrudredirect.js* into it, and then reference *your-new-javascript.js* from *YourNewView.cshtml*.

2 Other Settings

2.1 Automatically Open Selected Tables or Selected Views Tab

In the *Code Settings Tab*, when you click the *Selected Tables Only* or *Selected Views Only* option under the *Database Object to Generate From* and this (*Automatically Open Selected Tables or Selected Views Tab*) is checked, AspCoreGen 3.0 Razor will automatically open the *Selected Tables Tab* or *Selected Views Tab* respectively.

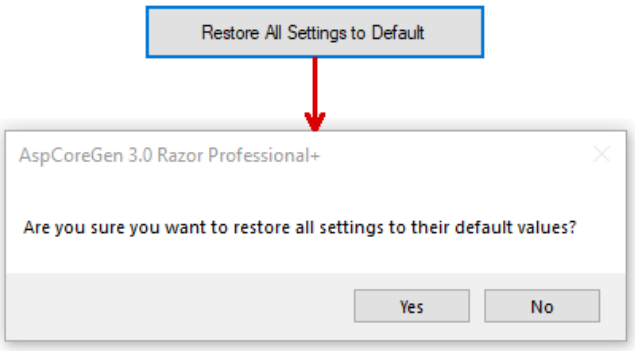


2.2 App Files Directory

The *App Files Directory* is the location of the *App Files* folder is in your computer. The default directory is: *C:\Program Files (x86)\Junnark.Com\AspCoreGen 3.0 Razor Professional Plus\AppFiles*. AspCoreGen 3.0 Razor uses the *App Files Directory* for various settings required by the application.

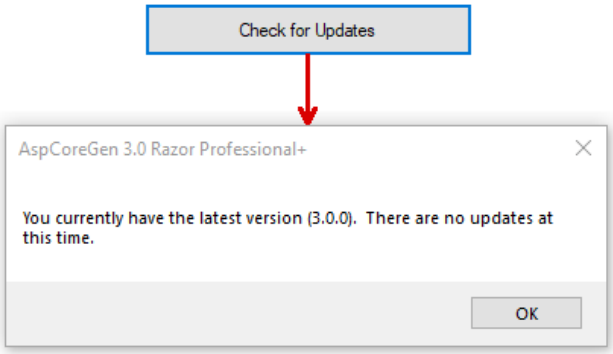
2.3 Restore All Settings to Default (Button)

This button will restore AspCoreGen 3.0 Razor's settings to their default values just like when you first installed it. This will not make you reactivate AspCoreGen 3.0 Razor. It will also pop-up a confirmation dialog to before completely restoring of all settings. Clicking *Yes* will *Restore All Settings to Default*.

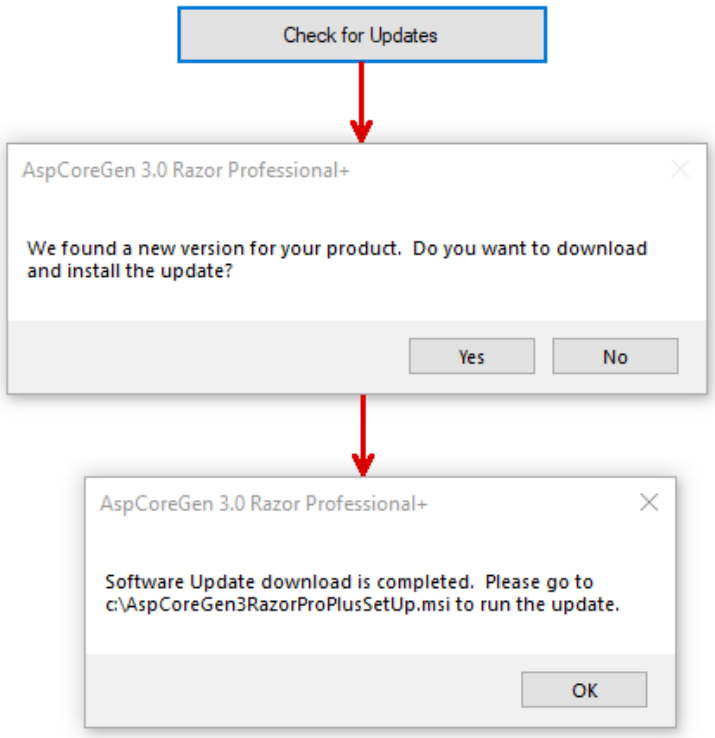


2.4 Check for Updates (Button)

This button will check if there is an available Update for AspCoreGen 3.0 Razor. If there is none, a message will pop-up letting you know that you have the latest version.



The update will be downloaded to a local folder, or it will start the installation when an update is available.



You can read end-to-end tutorials on more subjects on using AspCoreGen 3.0 Razor Professional Plus that came with your purchase. These tutorials are available to customers and are included in a link on your invoice when you purchase AspCoreGen 3.0 Razor Professional.

Note: Some features shown here are not available in the Express Edition.

End of tutorial.