App Settings Tab

1 CONTENTS

1	Ger	nerat	ed Projects	3
	1.1	We	b Application Project	4
	1.1.	1	Files That Will Be Written Once	4
	1.1.	2	Files That Will Always Be Overwritten	5
	1.2	Cla	ss Library Project	6
	1.2.	1	Files That Will Be Written Once	6
	1.2.	2	Files That Will Always Be Overwritten	7
	1.3	We	b API Project	7
	1.3.	1	Files That Will Be Written Once	7
	1.3.	2	Files That Will Always Be Overwritten	7
	1.4	Wh	y is it Important to Know Which Files Will be Written Once or Always Overwritten	8
	1.4.	1	Files That Will Be Written Once	8
	1.4.	2	Files That Will Always Be Overwritten	9
2	Oth	er Se	ettings	9
	2.1	Aut	omatically Open Selected Tables or Selected Views Tab	9
	2.2	Ар	o Files Directory	10
	2.3	Res	tore All Settings to Default (Button)	10
	2.4	Che	eck for Updates (Button)	10

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.

🚺 AspCoreGen 3	.0 Razor Profession	nal+		-	-	\times
Selected Table	s Selected Views	Database Settings	Code Settings	UI Settings	App Settings	
Web App	lication Project -					
Files th	aat will be written one	<u>ce</u>	Files that will	always be ove	erwritten	
Class Lib	rary Project					
<u>Files th</u>	nat will be written one	<u>ce</u>	Files that will	always be ove	erwritten	
Web API	Project					
<u>Files th</u>	at will be written one	<u>ce</u>	Files that will	always be ove	erwritten	
2 Ann F	tomatically Open Se les Directory:	elected Tables or Sele	ected Views tab			
C:\Program	i Files (x86)\Junnark	Com\AspCoreGen 3	.0 Razor Profess	ional Plus\Ap	pFiles\	
					browse	
Res	tore All Settings to E	Default	Che	ck for Update:	s	
About Clo	se	Generate	Code for All	Tables	Can	cel

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)

Name:	МуАрр
Directory:	C:\inetpub\wwwroot\ browse
Generat	e Code Examples Over Server Port: 27229
usiness Layer	and Data Layer API
Name:	MyAppAPI
Name: Directory:	MyAppAPI C:\inetpub\www.root\MyApp\MyAppAPI
Name: Directory:	MyAppAPI C:\inetpub\wwwroot\MyApp\MyAppAPI
Name: Directory:	MyAppAPI C:\inetpub\wwwroot\MyApp\MyAppAPI
Name: Directory: eb AP1	MyAppAPI C:\inetpub\wwwroot\MyApp\MyAppAPI
Name: Directory: eb API	MyAppAPI C:\inetpub\wwwroot\MyApp\MyAppAPI
Name: Directory: birectory:	MyAppAPI C:\inetpub\wwwroot\MyApp\MyAppAPI b API MyAppWebAPI

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.

Web Application Project	
Files that will be written once	Files that will always be overwritten
Class Library Project	
Files that will be written once	Files that will always be overwritten
Web API Project	
Files that will be written once	Files that will always 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

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

1.1.2 Files That Will Always Be Overwritten

- o 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. *ProductsBase.cs* is a *Parent (Base) Class. Products.cs* is a *Child Class* because it inherits from the *ProductsBase.cs.*

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

Files Generated Once - Class Library Project (API)	×	Files Always Generated - Class Library Project (API)	×
These are the files that will be generated once and will not be overwritten the next time you generate code for the same project. Feel free to add your own code to these files. 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		These are the files that will always be generated or overwritten when they already exist. Do not add code to these files. BusinessObject (Folder) - Base (Folder) - All generated files DataLayer (Folder) - Base (Folder) - All generated files Models (Folder) - Base (Folder) - All generated files	
ОК		ОК	

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
- o 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
- o appsettings.json
- o 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. *ProductsBase.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 *ProductsBase*.cs, mostly an empty class but have all the functionality of the *ProductsBase.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: *ProductsBase*.InsertAsync(), *ProductsBase*.UpdateAsycn(), *ProductsBase*.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.

Selected Tables	App Settings Code Settings UI Settings App Settings Selected Tables or Selected Views tab Database Objects to Generate From Image: Code Settings UI Settings App Settings Selected Wews Database Settings Code Settings UI Settings App Settings Selected Wews Database Settings Code Settings UI Settings App Settings Selected Wews Database Settings Code Settings UI Settings App Selected Wews Database Settings Code Settings UI Settings App Selected Views Database Settings Code Settings UI Settings App Selected Views Selected Tables Selected Views Only Selected Views Only Selected Views Only	
	Selected Tables Selected Views Database Settings Code Settings UI Settings App Settings	

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.

App Files Directory:	
C:\Program Files (x86)\Junnark.Com\AspCoreGen 3.0 Razor Professional Plus	\AppFiles\
	browse

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*.

	Restore All Set	tings to Default]
			-
AspCoreGen 3.0) Razor Professional	+	×
Are you sure yo	u want to restore all	settings to their de	fault values?
		Yes	No

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.

	Check for Updates	
AspCoreGen 3.0	Razor Professional+	×
You currently ha this time.	ve the latest version (3.0.0). There are no updates at	
	ОК	

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

		Check for Updates	
AspCo	reGen 3.0 F	Razor Professional+	
We for and in	und a new stall the up	version for your product. Do you want to do odate?	wnload
		Yes	No
	AspCore	Gen 3.0 Razor Professional+	×
	Software c:\AspCo	e Update download is completed. Please go t oreGen3RazorProPlusSetUp.msi to run the upd	o late.
		o	К

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.