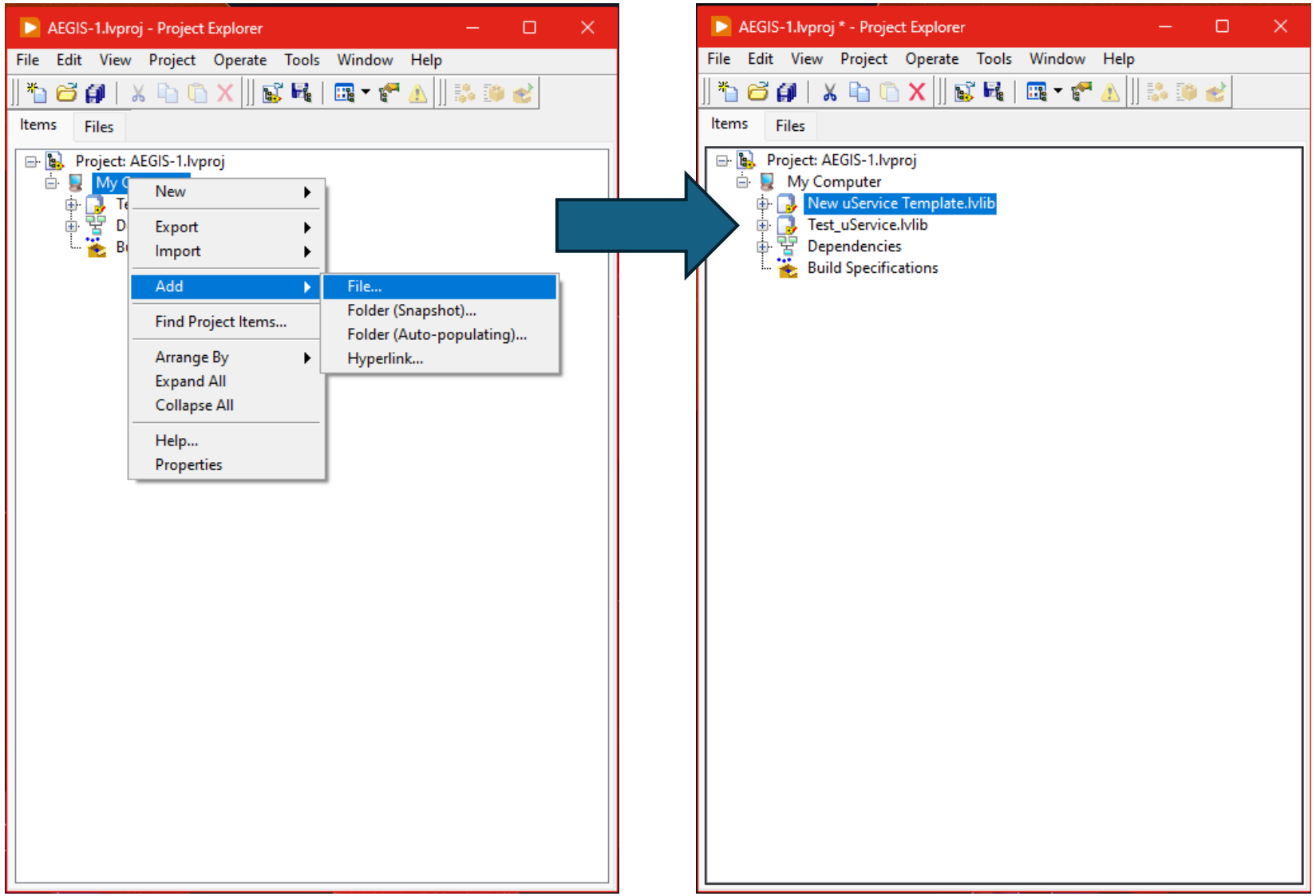
Making a new uService

An unofficial guide with pictures
by Jakub

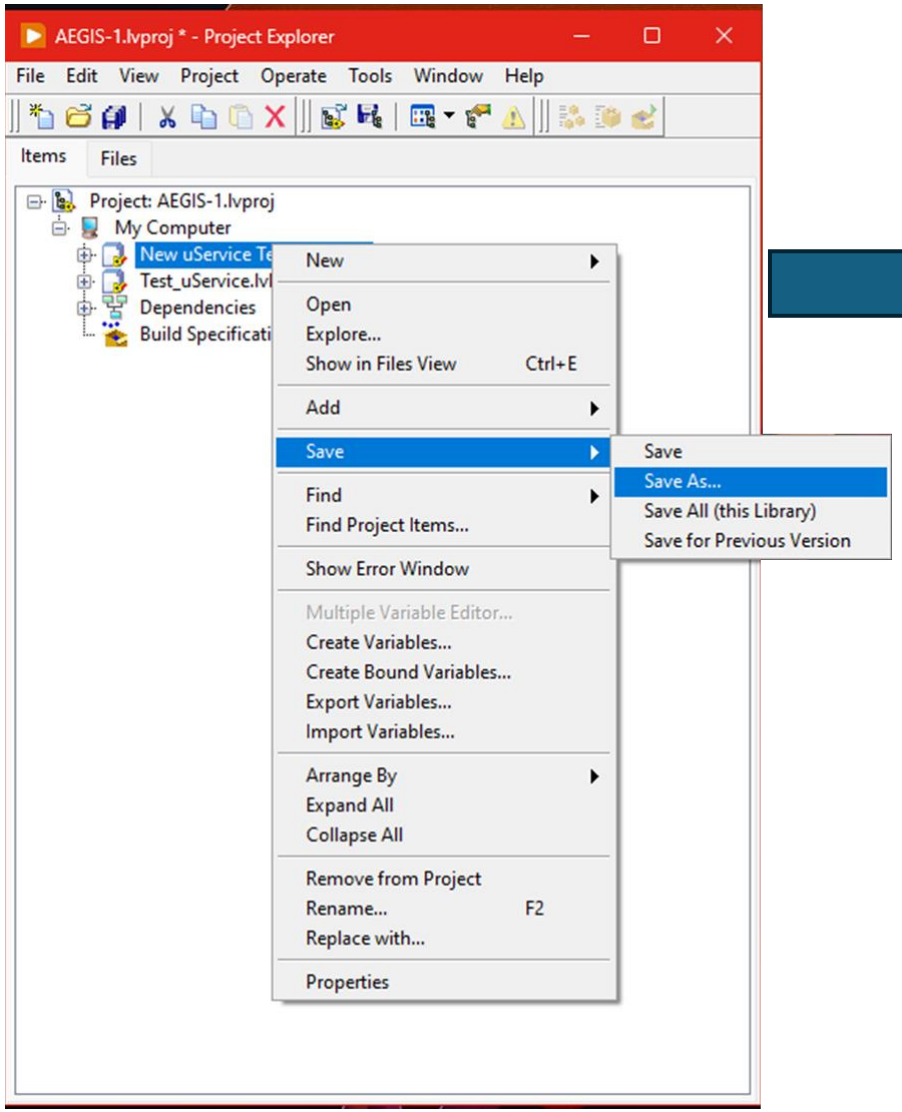
Table of content

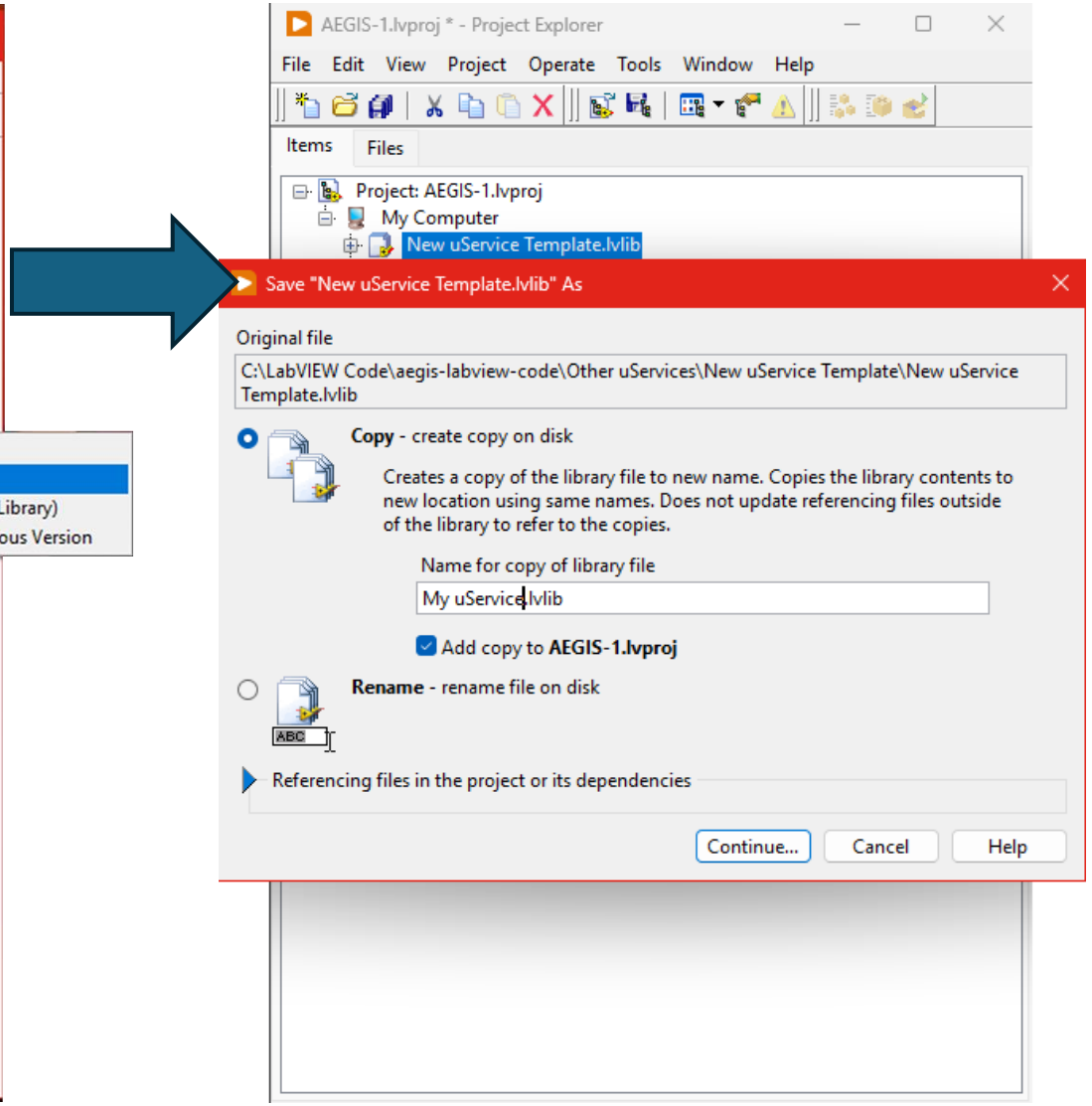
Add the New uService Template to the project



1. Right-click on the *My Computer* in the LV's Project Explorer and select *Add->File…*
2. Inside the directory of the repository (aegis-labview-code) select *Other uServices>New uService Template>New uService Template.lvlib*
   - You want to add the entire library, not only the class

Create a copy of the template

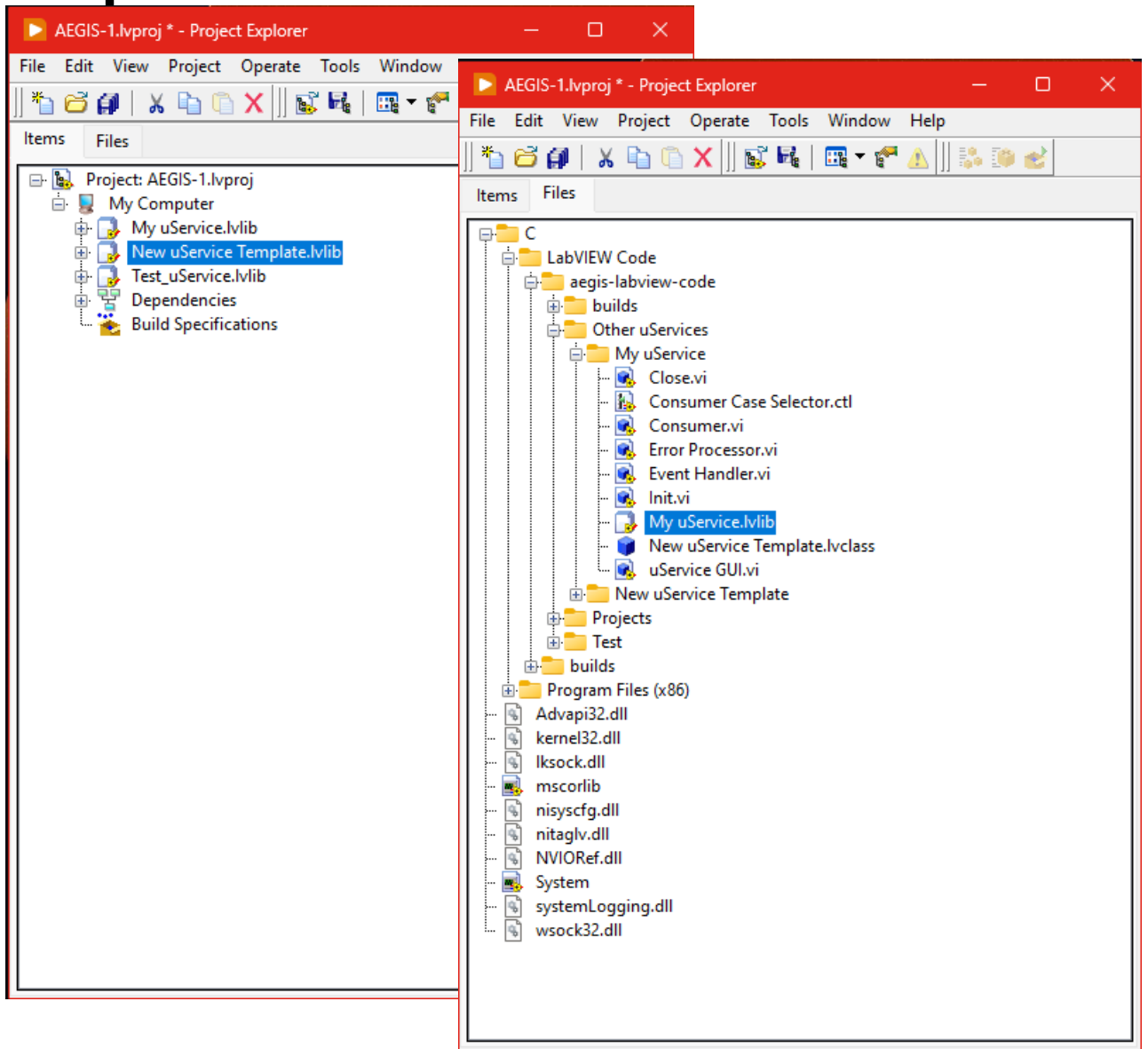




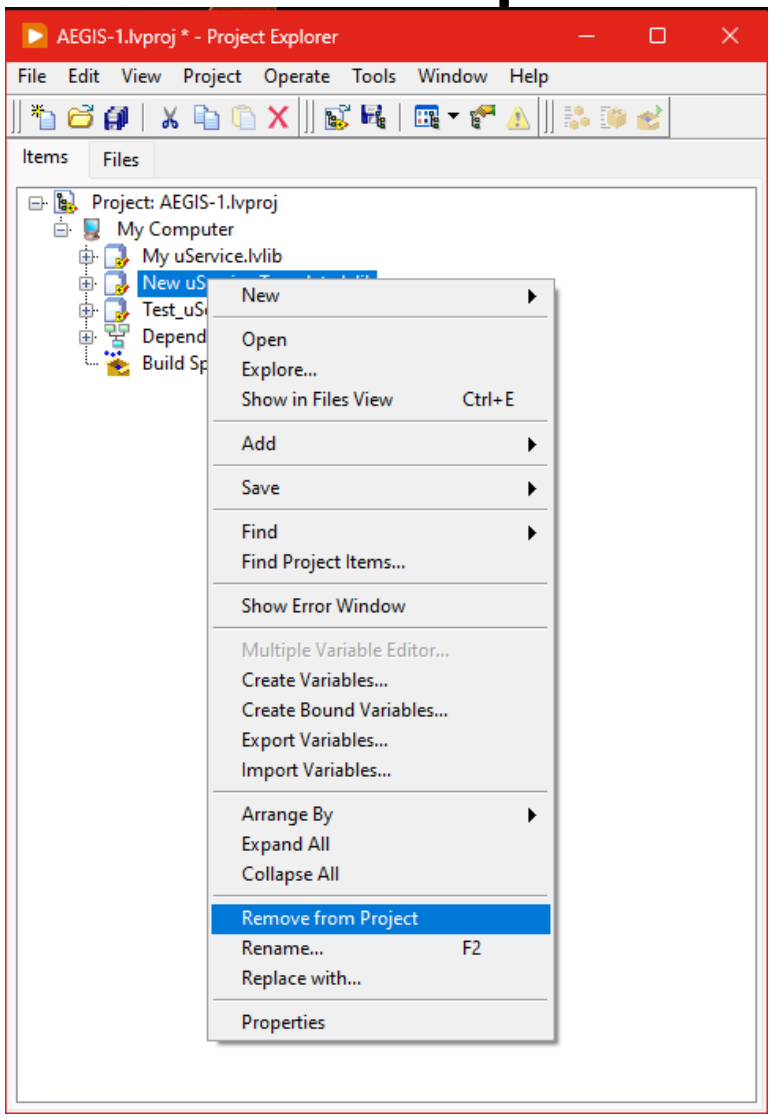
3. Right-click on *New uService Template.lvlib -> Save -> Save as...*
4. Baptise it as you wish (*My uService.lvlib* in the example on the slide)
   - Leave the flag *Add copy to <project name>.lvproj* ticked. This will add the new library to the project you are working in.
5. Press continue

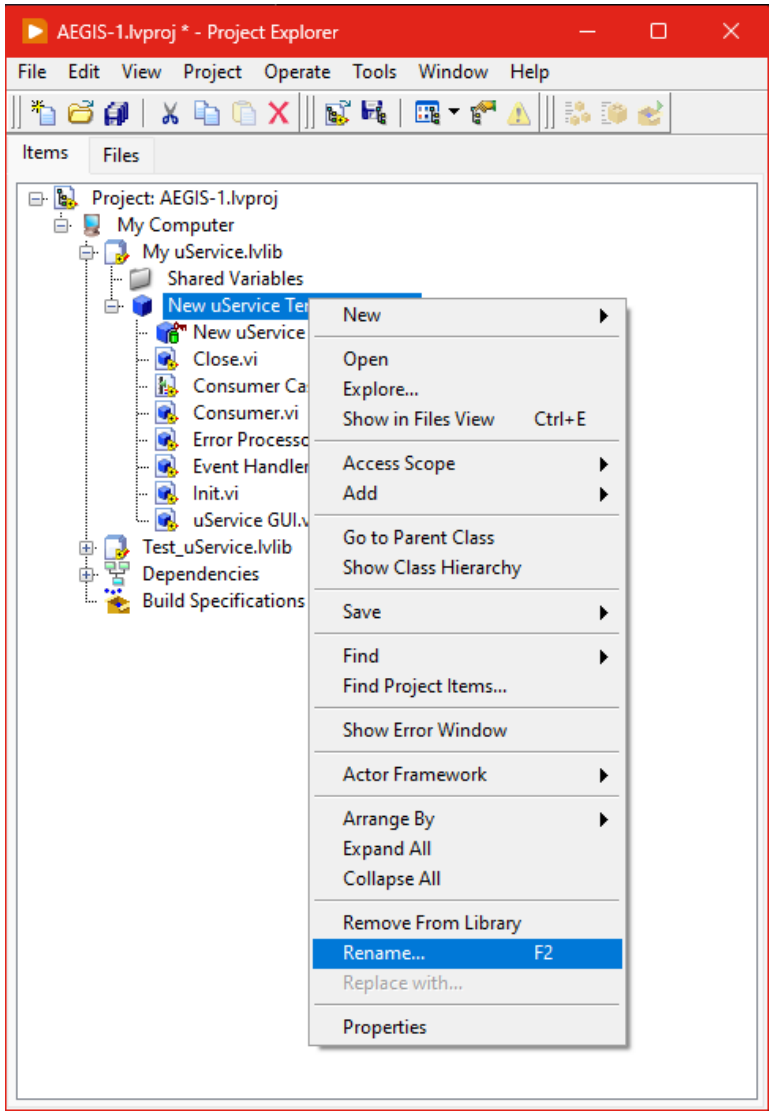
Create a copy of the template

6. Browse for the folder where you want to save your new uService. Some general rules:
   - *Controller Managers* folder -> uServices that control controllers (actuators, electro gun, Rotators, Temperature, etc.)
   - *Detector Managers* folder -> uServices that control detectors (1TCCD, 5152, Captorius, etc.)
   - *Other uServices* -> anything else (ELENA Interface, Telegram Bot)
   - The name of the folder for your library should be the same as the library (LabView will add *Folder* at the end; you can remove it to make it clean)
7. Press *Save*, it will create a new folder with the specified name and copy all the VIs inside



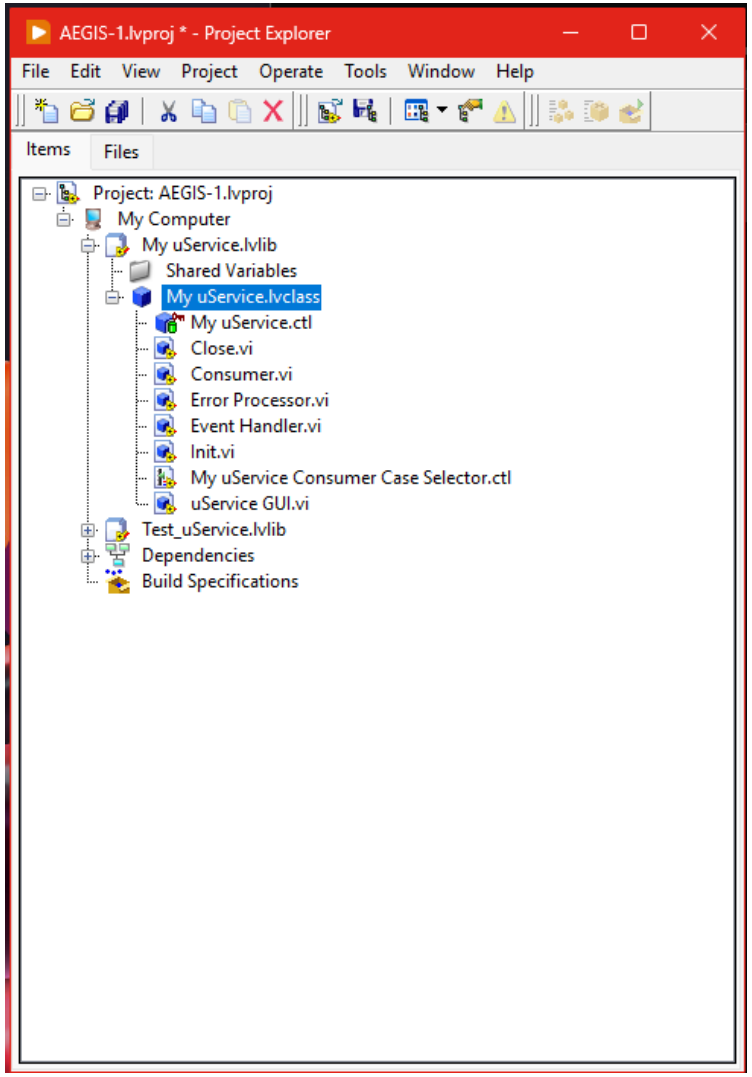
Clean up

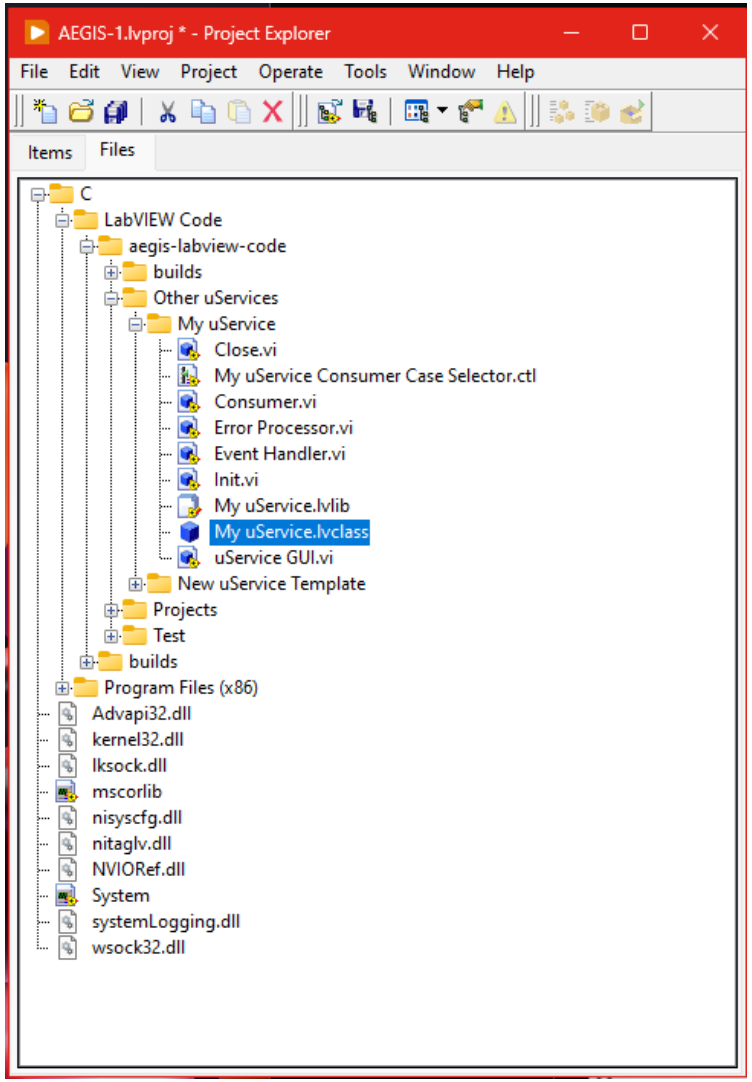




8. Remove the template
    1. Right-click on the *New uService Template.lvlip*
    2. Select *Remove from Project*
    3. Press *OK*
9. Rename the uService class
    1. Open the freshly created library
    2. Right-click on the *New uService Template.lvclass*
    3. Select *Rename*
    4. Give it the same name as the library
    5. Rename also the *Consumer Case Selector* by appending uService name in the front
    6. You can also edit the class's icon. It is recommended to at least set the *Text icon* (This can be accessed by right-clicking the class and selecting *Properties*)

Finish



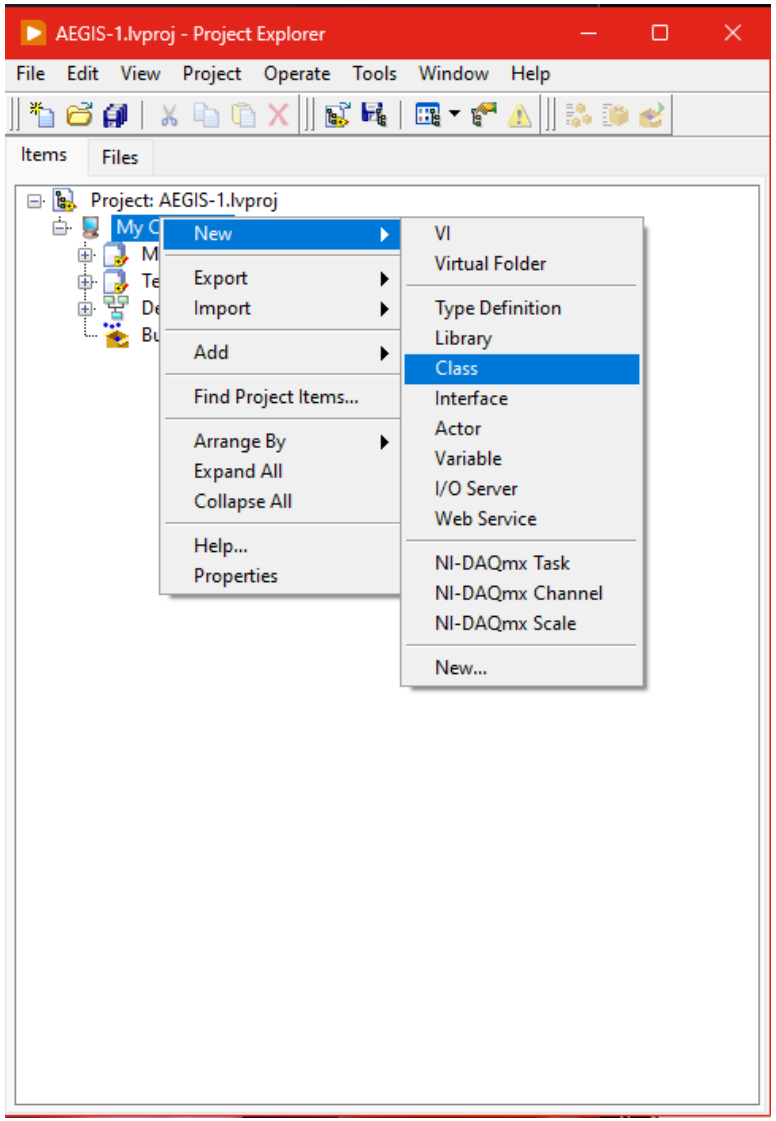


Your project should look like this.

Hardware and Detector class

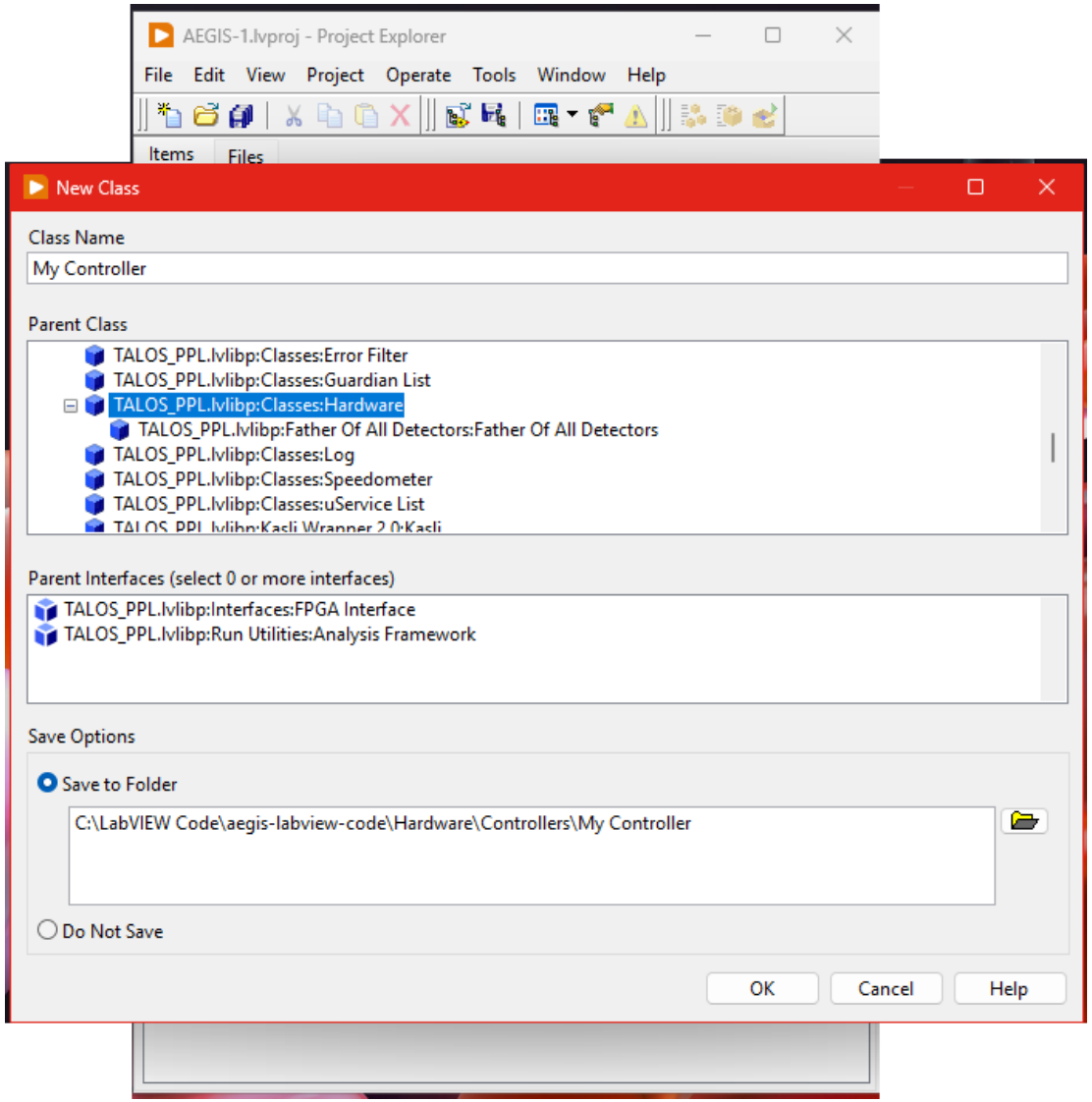
If you want to create a uService for a controller or detector with some new hardware, you will need to create a hardware class (Detector class is a child of the Hardware class)

Hardware/Detector class



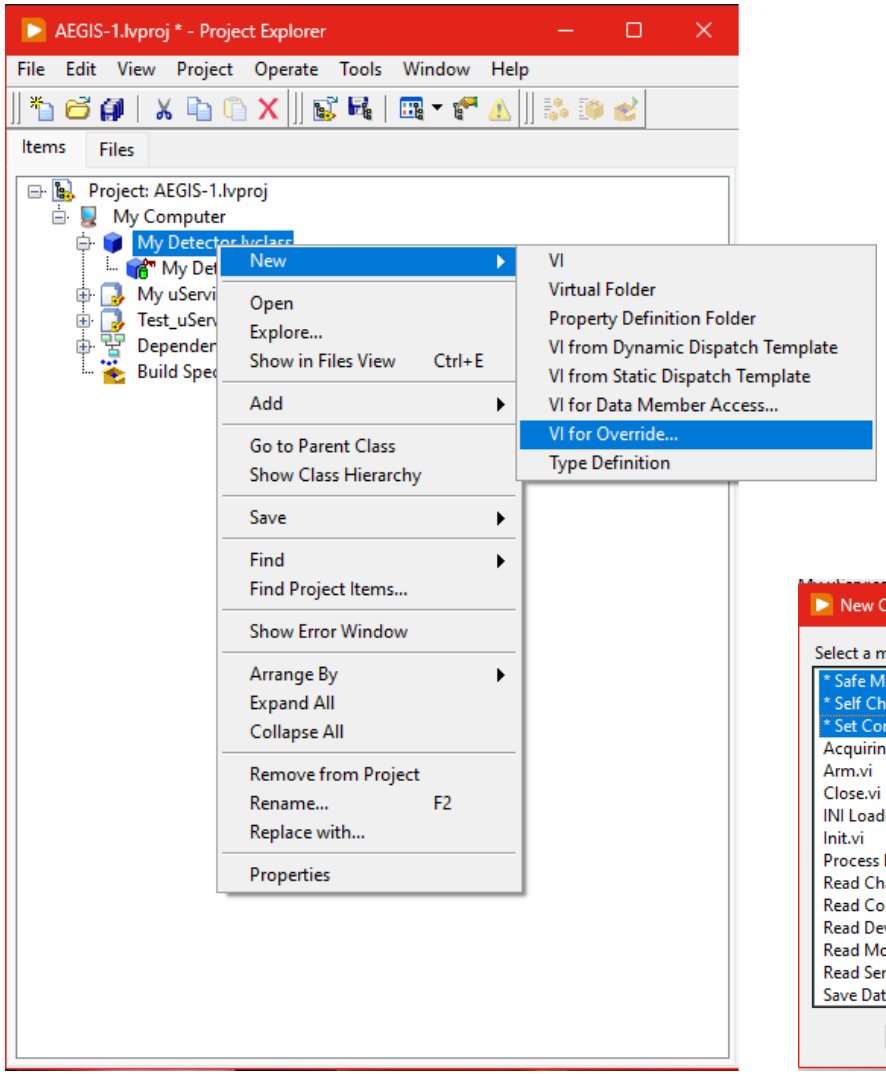
1. Right-click on the *My Computer* in the LV's Project Explorer and select *Add->Class*

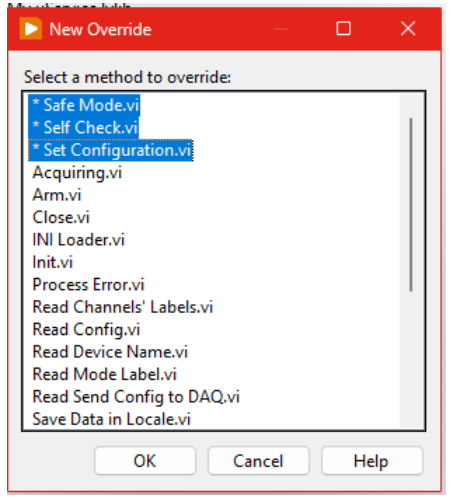
Hardware class



2. Give a new name to the class, usually it will be the name of the controller
    - This class is a general class for operating a specific controller. The same controller can be used by different uServices
3. From the *Parent Class* scroll down and selectr *TALOS_PPL.lvlibp:Classes:Hardware* (if you are implementing a hardware class for a detector, select the entry below *TALOS_PPL.lvlibp:Father of All Detectors:Father of All Detectors*)
4. Tick the *Save to Folder* and then press the browse button to the right. Navigate to the *Harware>Controllers* (or *Detectors*) folder in the project, type the name of your class and press save
5. Press Ok and your new class is created
6. Fill in the methods for the class by creating new VI's (preferably from the *Dynamic Dispatch Templace*)
    - This methods should be general ones for the device: init, close, set parameter, move (depending on the device of course)

Detector class





7. Detector class is used inside Detector Manager and it needs some methods to be overwritten
    1. Rick-click on the detector class created
    2. Select *New->VI for Override…*
    3. Select all Vis with the * and press OK
    4. Implement the functionality according to your desires

Detector Manager uService

Detector Manager is a special uService class that can be used to define new uService for a detector quickly.

You can create it with the same scheme as a new class, but selecting *TALOS_PPL.lvlibp:Detector Manager:Detector Manager* as the parent instead.

Then there are 2 Vis that should be overwritten:

- **Init.vi** – here a new instance of the detector class should be created and saved inside the private data cluster of the class
- **SubPanel.vi** – this will be the GUI inserted into the main the uService