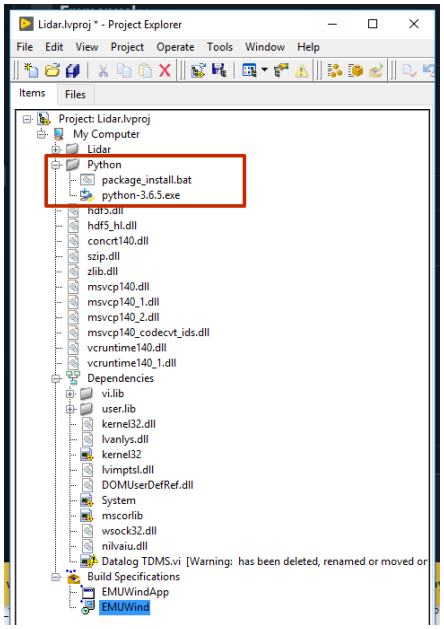
So you think you can compile/build LabVIEW…😏

You will need…

- LabVIEW Professional Development License
- All Sourcecode organised within a LabVIEW Project

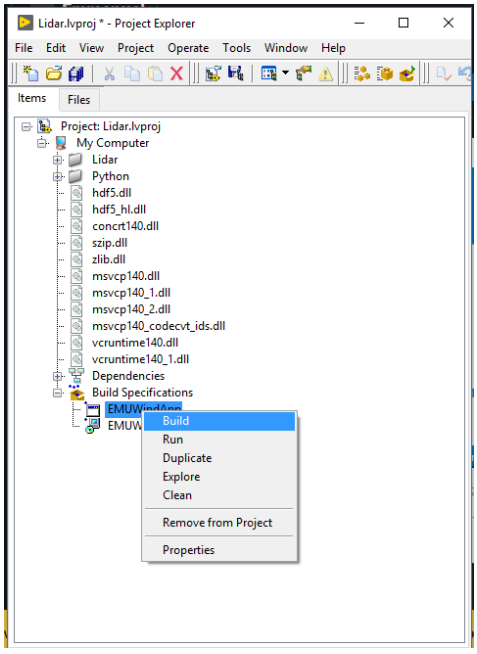
When in doubt refer to NI LabVIEW documentation: https://www.ni.com/docs/en-US/bundle/labview/page/building-and-distributing-applications.html

You can compile/build anything created under Build Specifications.

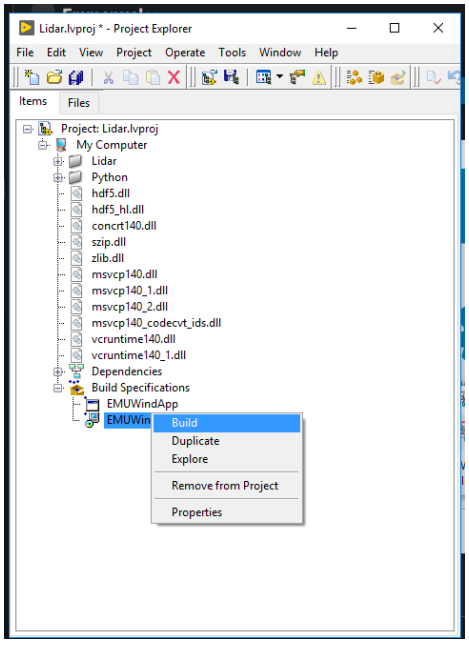


Python Installer and bat script needed for installer included within the project

- Note: If your installer needs other installers include them within the project.



Build executable



Build installer

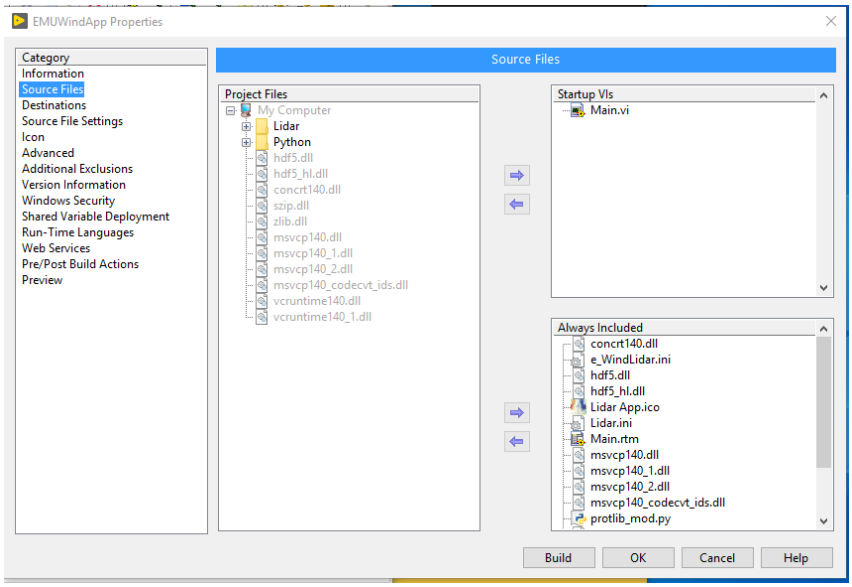
- Creating and building installers and executables are located in Build Specifications in the Project Explorer. **By right-clicking Build specifications, creation of other or new executables and Installer are possible**

• Customizing executable

Executables run on machines with installed LabVIEW Runtime or LabVIEW Software.

- To customize properties of installer or executable, use “Properties” menu in Project Explorer for executable

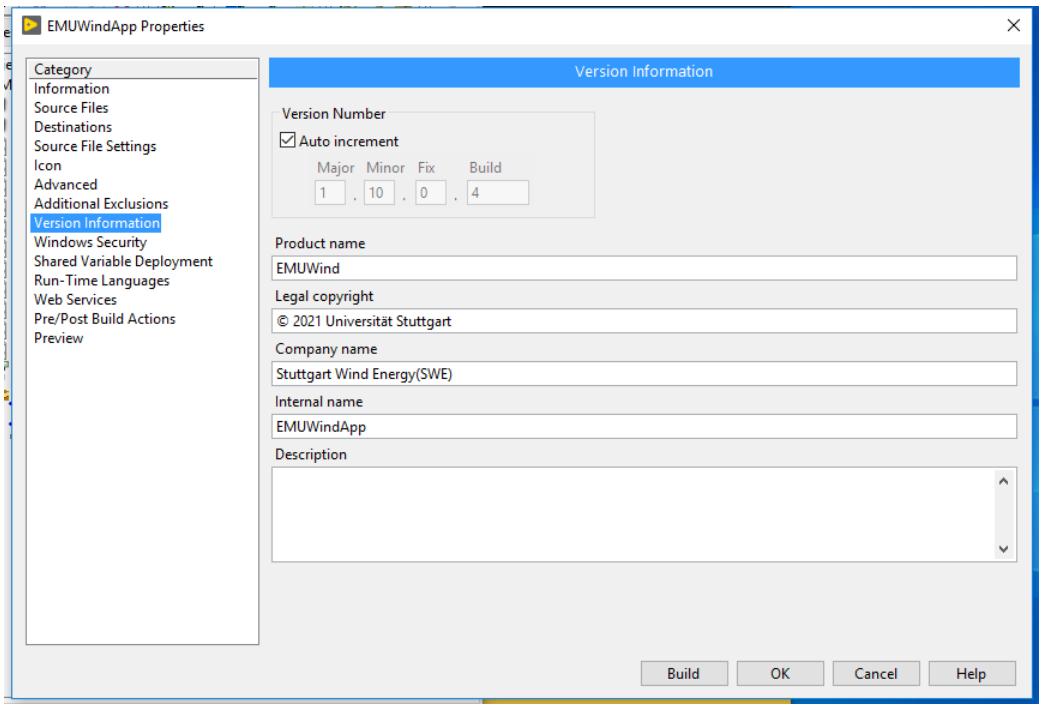
All source files necessary for executables can be customised in “Source Files”



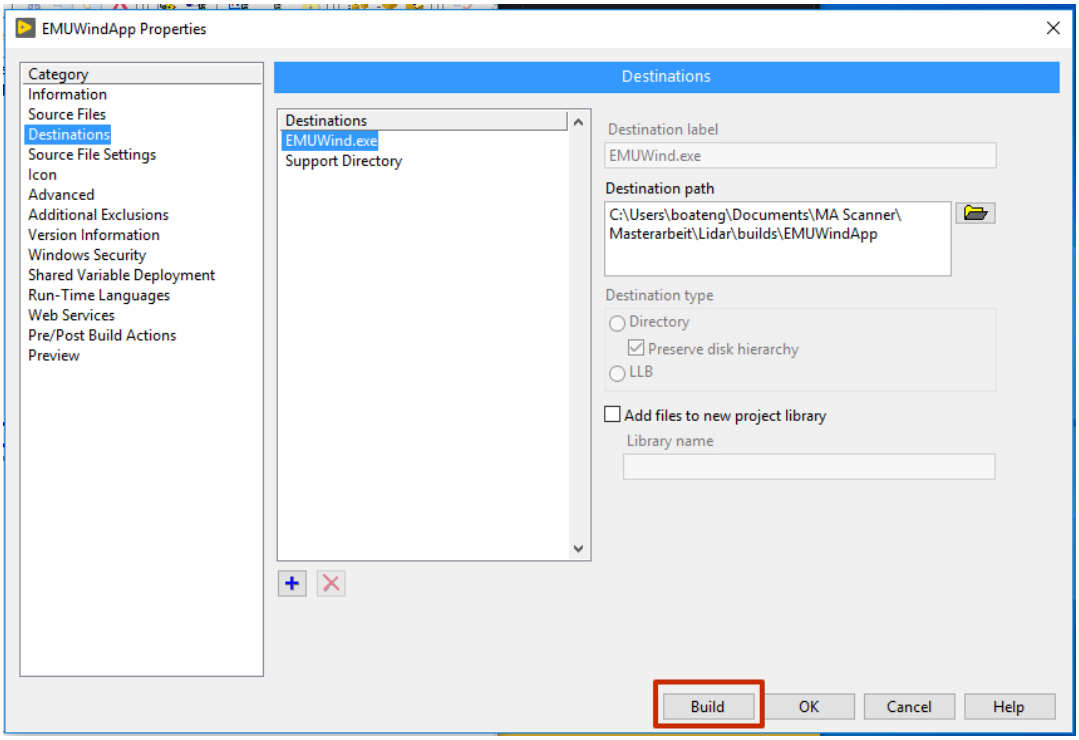
All DLLs used by non-standard libraries installed via VI Package Manager need to be within the project and included as Source Files.

Else, VI functions made using non-standard libraries will not work in built application

Version Information changes can be done automatically or manually by deselecting Auto-Increment



Customize the destination path of your application to be and BUILD
- Note: every build auto increments the version information



Refer to LabVIEW Documentation for other functions listed in the category

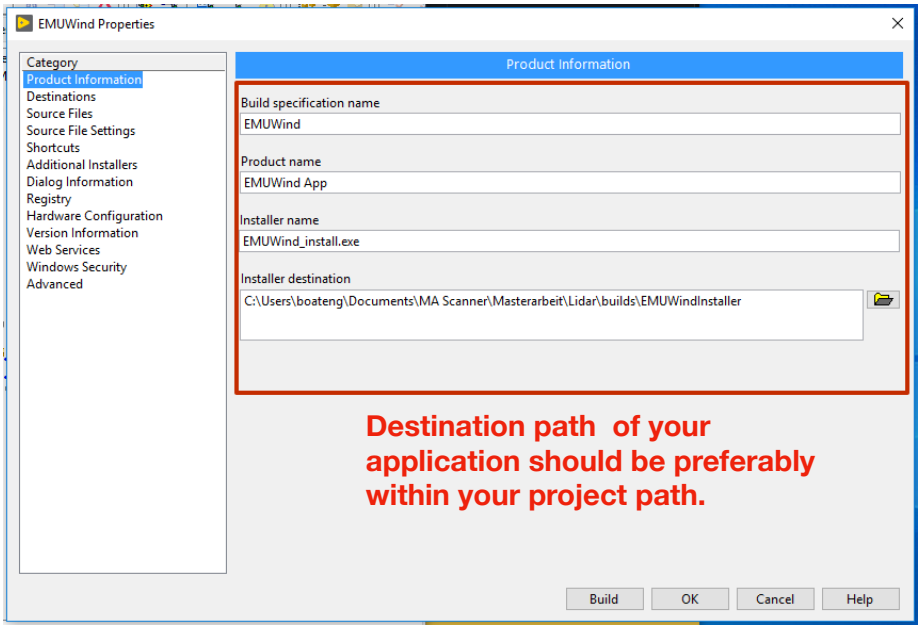
Step-by-Step Instructions

1. Open the Project Explorer:
   - Ensure all your source code is organized within the LabVIEW Project.
2. Access Build Specifications:
   - Right-click on “Build Specifications” in the Project Explorer to create a new executable.
3. Include Necessary Files:
   - Ensure all required source files and dependencies are included in the project. This includes any non-standard DLLs used by your application.
4. Customize Executable Properties:
   - Use the “Properties” menu in the Project Explorer to customize your executable’s settings, such as:
   - **Source Files**: Specify the source files to be included.
   - **Version Information**: Manage version control. You can enable auto-increment or manually update the version.
5. Specify Destination Path:
   - Set the destination path for your application.
6. Build the Executable:
   - Click “Build” to compile your application. Note that each build will auto-increment the version information.

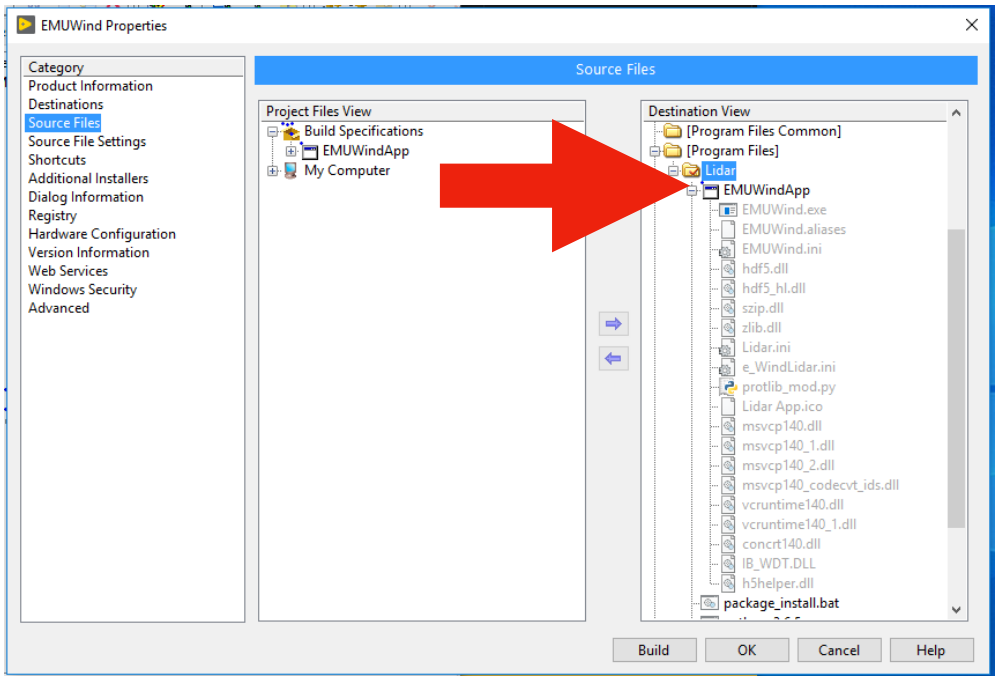
- **Customizing installers (a few pointers)**

Installers install your built application on any machine without the need for installing LabVIEW IDE.

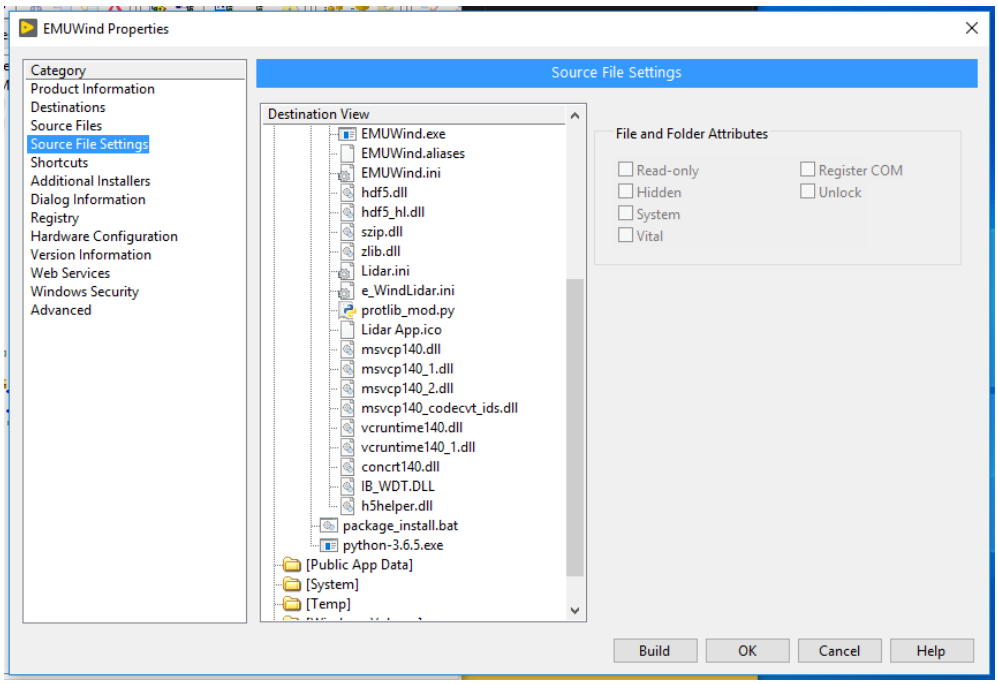
- Select name of installer, application name and destination path of the installer after building in Product Information view



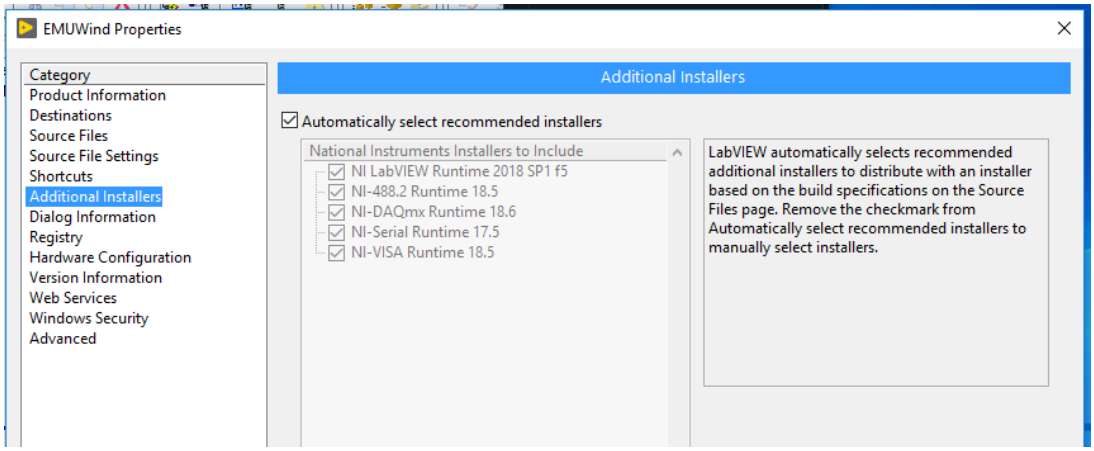
- All files that need to be included within the installer for your application should be selected in Source Files. This includes other installers such as Python installers. If the executable Source Files does not include needed files, they need to be added manually.



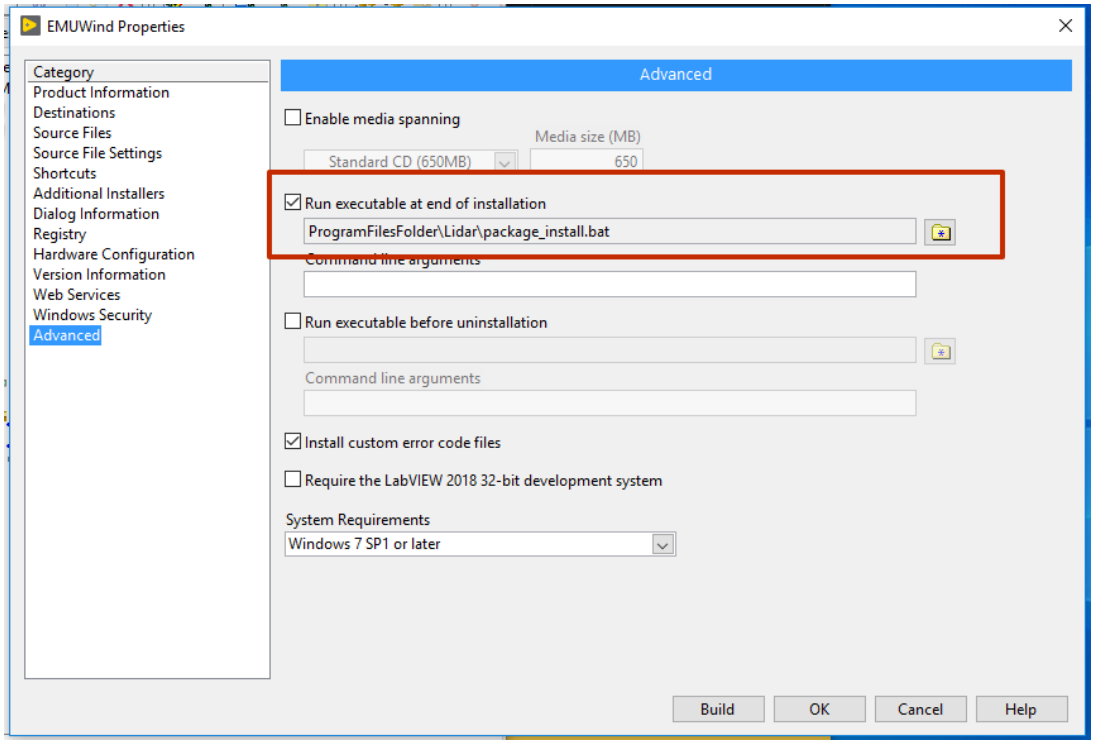
Customize individual source file settings and attributes if desired. Hide all files that user should not see in Program Files



Additional installers category is solely for LabVIEW Runtime. LabVIEW Runtime allows running applications without installing the full LabVIEW IDE or Software. Ideally use recommended automatic setting from National Instruments. Unless you know what you are doing. **For non-standard function libraries, check the needed runtime in NI Documentation if it is not installed**



In the Advanced Menu, you can include scripts or files that need to be run at the start or end of the installation of the application. ***package_install.bat is a shell script developed to install Python libraries "numpy" and "socket" used within executable/application for the program functionality. The script auto-deletes after installation is completed.***



Step-by-Step Instructions

1. Access Build Specifications:
   - Right-click on "Build Specifications" and select "New Installer" to begin creating an installer for your application.
2. Configure Installer Settings:
   - **Product Information**: Define the installer name, application name, and destination path.
   - **Source Files**: Include all files necessary for the application, including additional installers (e.g., Python installer). Ensure non-standard libraries are correctly referenced.
3. Customize Source File Settings:
   - Adjust individual source file settings and attributes as needed. Hide files that the user should not see in the Program Files directory.
4. Include Additional Installers:
   - Use the "Additional Installers" category to include the LabVIEW Runtime. This is crucial for running applications on machines without the LabVIEW IDE.
5. Advanced Menu Options:
   - You can include scripts or files to be executed at the start or end of the installation. For example, package_install.bat can be used to install necessary Python libraries like numpy and socket.
6. Build the Installer:
   - After configuring all settings, click "Build" to create the installer package or OK to save changes

Best Practices

- **Organize Your Project**: Keep all source files and dependencies well-organized within the LabVIEW Project for easier management.
- **Version Control**: Update version information and maintain a changelog to track changes in your builds.
- **Test Builds**: **Always test your executables and installers on a clean system to ensure all dependencies are correctly handled.**

Troubleshooting and Common Issues

- **Missing Dependencies**: If application fails to run, check that all necessary DLLs and additional installers are included in the build.
- **Installer Fails**: Verify that the destination paths and file permissions are correct. Ensure all required scripts are configured properly and execute as intended.