Output – Creating Installers or Zip Archives

Application Output · https://www.exeoutput.com/help/distributingpub

ExeOutput for PHP generates single executable files that are ready for distribution. You only need to distribute the application's `.exe` file (and, in some cases, additional external files located in the `Data` and/or `MySQL` subfolders) to your users or customers.

After building your application, you may want to distribute it over the internet. You can choose to package your compiled application into a **Zip archive** or an **installer/setup package**. Installers are advantageous because they can install multiple applications, install redistributables, include additional files (like README documents), create shortcuts in the Windows Start menu, and offer an uninstaller to allow users to remove all traces of your application from their computer.

Once an application is created, you can use the **Export to Zip** or **Create Installer** commands in **Application Output**.

Exporting to Zip

ExeOutput for PHP will prompt you to specify the location for the Zip archive. It will then scan for all files used by your application (including the `.exe` file, `Data` subfolder, and the MySQL subfolder) and compress them into the Zip archive.

Note

ExeOutput for PHP stores the correct path information within the Zip archive. You only need to deploy the Zip archive; users can extract it on their computer and run your application immediately.

Generating an Installer or Setup Package

ExeOutput for PHP utilizes **Paquet Builder** to generate custom, compact setup programs for distribution. You must install Paquet Builder before using this feature.

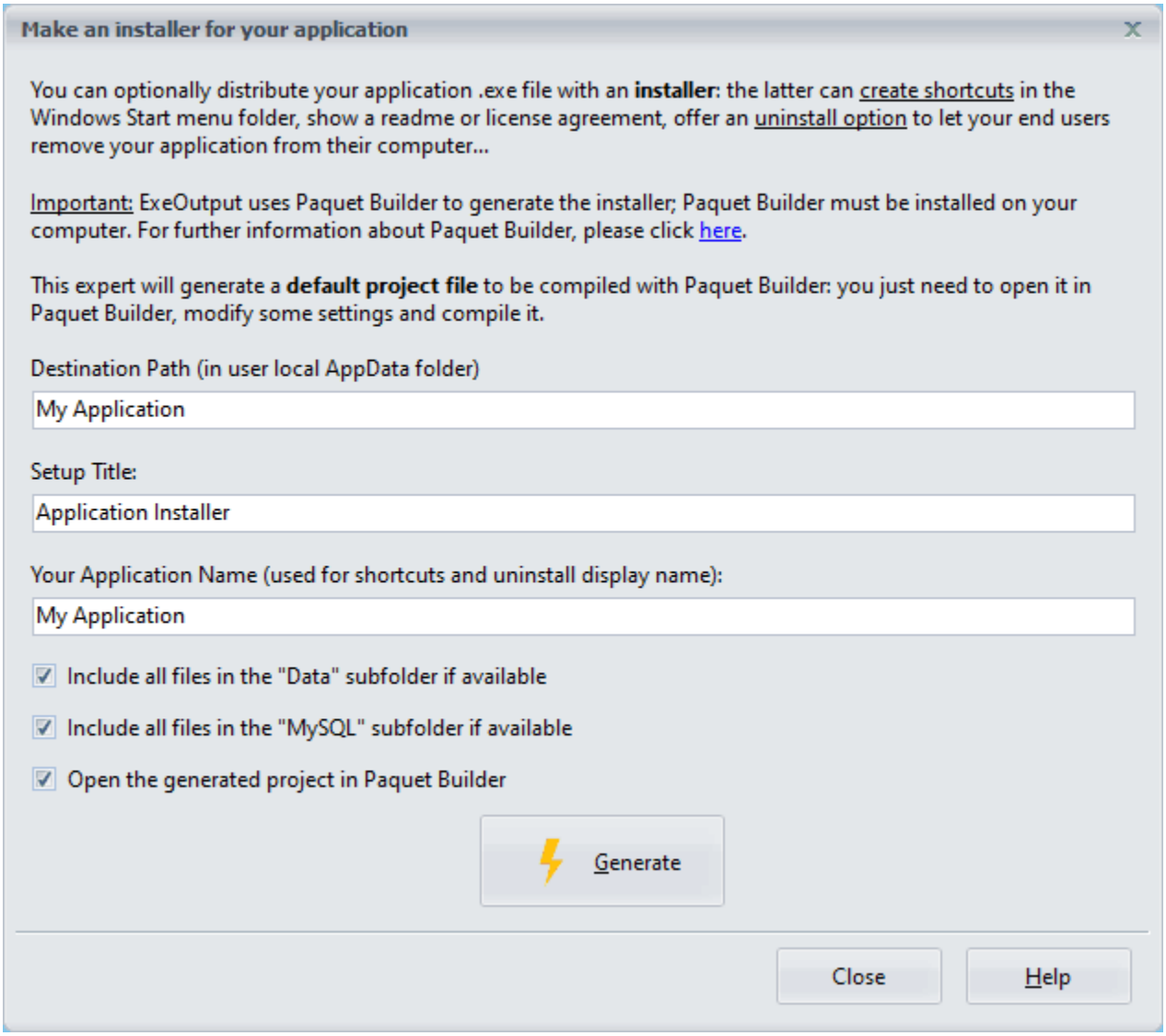
Paquet Builder combines the functionalities of a 7z self-extracting archive creator and a setup routine generator. With its comprehensive feature set, you can create flexible and compact self-extractors for

professional file and software distribution. Package any document or program files, visually construct simple or sophisticated multilingual distribution and installation packages, generate updates and patches, and wrap multimedia presentations or multiple Windows Installer MSI setups into single `.exe` files for internet distribution.

More information about Paquet Builder can be found at https://www.installpackbuilder.com.

How It Works

You will encounter a window with the following fields:



ExeOutput for PHP will create a default project for Paquet Builder. You can then edit this project within Paquet Builder and compile it to create the setup package.

First, fill in the three fields: **Destination Path**, **Setup Title**, and **Your Application Name**.

- **Destination Path**: The default installation folder for your application.
- **Setup Title**: The title that will appear on all setup windows.
- **Your Application Name**: The name of your application.

Next, click **Generate** to create the project. ExeOutput for PHP will then launch Paquet Builder, allowing you to modify the project as needed.

Tip

ExeOutput for PHP will automatically include external files from your application's `Data` and `MySQL` subfolders, unless you disable this option.

Exploring the Storage Folder

ExeOutput for PHP applications store data such as settings, cache, and user preferences in a dedicated **storage folder** on the user's computer.

To easily access this folder for testing and debugging purposes, you can click the **Explore Storage Folder** button on the Deployment page in ExeOutput for PHP. This will open the storage folder for the last compiled application in Windows Explorer.

The location of the storage folder can be customized. See the deployment options for more details.