PamguardCompiler uses 2 different programs to create the Pamguard installer:

1. Launch4j takes an executable jar file and turns it into an exe file that Windows can run natively
2. Inno Script takes all the files you give it and bundles them all up into an installer package.

PamguardCompiler presents the user with a set of options (installer name, file paths, java params, etc). When you hit the Compile button, it creates 3 files:

1. tempProperties.txt, which contains the options the user has selected and is used to repopulate the GUI with previous values next time the app is run
2. BuilderConfigs/antSettings.txt, which also contains the options the user has selected but in a different format, to be used by the ant script when compiling
3. BuilderConfigs/tempclasspath.txt, which copies the information from the .classpath file to also be used in the ant script when compiling.

PamguardCompiler then executes the ant script located in BuilderConfigs/Create\_Pamguard\_Executable.xml OR Create\_Pamguard\_Executable\_withJRE.xml, depending on whether or not the user wants to bundle the jre into the installer. The \_withJRE version has extra code to copy JRE files into the appropriate subfolders, and some added parameters so that the executables know where to look for the JRE. Unfortunately I couldn’t combine these into a single xml doc because (as far as I know) ANT v1.8 doesn’t have IF statements, and I don’t know ANT script well enough to come up with a workaround. **So if we need to make any changes to the script, we need to make the changes to BOTH versions.**

Create\_Pamguard\_Executable.xml loads the information in antSettings.txt and tempclasspath.txt. It pulls the version info out of PamguardVersionInfo.java (which is why that class needs to have a main method). It clears out the files in WorkingDir and then compiles a new runnable jar file (and associated helper files) into the empty WorkingDir subfolder. It creates Windows-compatible executables, copies any remaining support files (libraries, jre, readme, etc) and then starts the Inno Script script, passing it a bunch of params.

Inno Script takes the params passed from Create\_Pamguard\_Executable and bundles everything together into a single executable installer file. There are a lot of sections to the script, controlling how the install wizard looks and acts, as well as setting registry keys and deleting unused files. Inno Script help files are pretty confusing, but if you need to do something just google it and there are quite a few sample scripts out on the web. Plus, stackoverflow forums have questions/answers about how to do things.