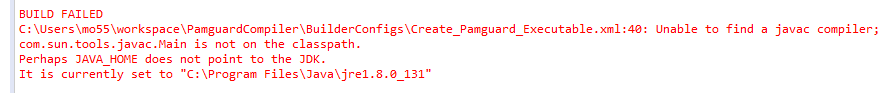
If you get this error when trying to run PamguardCompiler:



Then it means your build path is not set up correctly.

First, you have to have a JDK installed – a JRE will not have javac.exe. Java 8 JDK installers can be downloaded here:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html#close>

Now configure the project. In Eclipse:

* right-click on the project in the Package Explorer and select Build Path > Configure Build Path.
* In the libraries tab, find the JRE System library. Highlight it and hit Edit
* Click on the Installed JREs button
* If there is no JDK listed in the Installed JREs box:
  + Click Add
  + Select Standard VM and hit Next
  + Click the Directory button and select the top-level JDK folder (e.g. C:\Program Files\Java\jdk1.8.0\_131)
  + Click the Add External JARs button
  + Navigate to the JDK install folder, and go into the lib subfolder. Select tools.jar, and hit Open
  + Select Finish
* If there is a JDK listed, it’s not set up properly. Highlight it and hit the Edit button, then:
  + Click the Add External Libraries button
  + Navigate to the JDK install folder, and go into the lib subfolder. Select tools.jar, and hit Open
  + Select Finish
* Hit OK to close the Preferences window
* Make sure the JDK is selected as the library to use for this project. If it’s not the Workspace Default, click on Alternate JRE and make sure to select it in the drop-down box. Hit Finish.
* Hit OK.

If you get this error when trying to run PamguardCompiler:



The problem is the letter used in the version number. Only letters a-f can be used, because they can be converted by Windows to integers (it must assume they are hex).

In the Create\_Pamguard\_Executable.xml script, the version number is read from PamguardVersionInfo and stored in the property nsisVersion. This is a string, so no problems with the letter yet.

When launch4j is then run, nsisVersion is used for the property versionInfo.fileVersion. This is checked in PamguardCompiler\Tools\launch4jForPamguard\src\net\sf\launch4j\config\VersionInfo.java to make sure it only contains numbers or letters – again, no problem. It gets saved in the field fileVersion.

C:\Users\mo55\workspace\PamguardCompiler\Tools\launch4jForPamguard\src\net\sf\launch4j\RcBuilder.java uses VersionInfo.fileVersion to build up a string buffer in method addVersionInfo.

C:\Users\mo55\workspace\PamguardCompiler\Tools\launch4jForPamguard\src\net\sf\launch4j\Builder.java uses the stringBuffer in it’s build() method to create a temp file, and calls bin\windres.exe to generate a coff file which can be used by Windows resources to set the output file (the compiled executable) resources. This is where it fails: according to this page:

https://msdn.microsoft.com/en-us/library/windows/desktop/aa381058(v=vs.85).aspx

The FILEVERSION statement in the VERSIONINFO resource must contain 4 16-bit integers. I guess that it’s able to interpret letters a-f as integers, probably assuming they are hex values. But any other letter will cause it to fail.

Because this is a Windows issue, there is no way around it. At the moment we only use 3 integers to define the version number, appending .00 on the end of the nsisVersion property in Create\_Pamguard\_Executable.xml to add the fourth. We either start using the fourth integer instead of letters, or resign ourselves to only using letters up to ‘f’.