Code Howtos / JPackage: Creating a binary and debug it

JPackage: Creating a binary and debug it

JabRef uses jpackage to build binary application bundles and installers for Windows, Linux, and macOS. For Gradle, we use the Badass JLink Plugin.

Build Windows binaries locally

Preparation: Install WiX Toolset

1. Open administrative shell
2. Use Chocolatey to install it: `choco install wixtoolset`

Create the application image:

```
./gradlew -PprojVersion="5.0.50013" -PprojVersionInfo="5.0-ci.13--2020-03-05--c8e5924" jpackageImage
```

Create the installer:

```
./gradlew -PprojVersion="5.0.50013" -PprojVersionInfo="5.0-ci.13--2020-03-05--c8e5924" jpackage
```

Debugging jpackage installations

Sometimes issues with modularity only arise in the installed version and do not occur if you run from source. Using remote debugging, it’s still possible to hook your IDE into the running JabRef application to enable debugging.

Debugging on Windows

1. Open `build.gradle`, under jlink options remove `--strip-debug`
2. Build using `jpackageImage` (or let the CI build a new version)
3. Modify the `build\image\JabRef\runtime\bin\Jabref.bat` file, replace the last line with

   ```
   pushd %DIR% & %JAVA_EXEC% -Xdebug -Xrunjdwp:server=y,transport=dt_socket,address=8000,suspend=n -p "
   ```

4. Open your IDE and add a “Remote Debugging Configuration” for `localhost:8000`
5. Start JabRef by running the above bat file
6. Connect with your IDE using remote debugging