

```
1 #!/bin/sh
2 # -----
3 # Licensed to the Apache Software Foundation (ASF)
4 # under one or more contributor license agreements. See the
5 # NOTICE file
6 # distributed with this work for additional
7 # information
8 # regarding copyright ownership. The ASF licenses
9 # this file
10 # to you under the Apache License, Version 2.0 (the
11 # "License"); you may not use this file except in
12 # compliance
13 # with the License. You may obtain a copy of the
14 # License at
15 #
16 #     https://www.apache.org/licenses/LICENSE-2.0
17 #
18 # Unless required by applicable law or agreed to in
19 # writing,
20 # software distributed under the License is
21 # distributed on an
22 # "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF
23 # ANY
24 # KIND, either express or implied. See the License
25 # for the
26 # specific language governing permissions and
27 # limitations
28 #
29 # under the License.
30 #
31 # -----
32 #
33 # Maven Start Up Batch script
34 #
35 # Required ENV vars:
36 #
37 # JAVA_HOME - location of a JDK home dir
38 #
```

```
28 # Optional ENV vars
29 #
30 #   M2_HOME - location of maven2's installed home dir
31 #   MAVEN_OPTS - parameters passed to the Java VM
32 #     when running Maven
33 #       e.g. to debug Maven itself, use
34 #             set MAVEN_OPTS=-Xdebug -Xrunjdwp:transport=
35 #                   dt_socket,server=y,suspend=y,address=8000
36 #   MAVEN_SKIP_RC - flag to disable loading of
37 #     mavenrc files
38 #
39 # -----
40 # -----
41 #
42 #
43 if [ -z "$MAVEN_SKIP_RC" ] ; then
44     if [ -f /usr/local/etc/mavenrc ] ; then
45         . /usr/local/etc/mavenrc
46     fi
47     if [ -f /etc/mavenrc ] ; then
48         . /etc/mavenrc
49     fi
50     if [ -f "$HOME/.mavenrc" ] ; then
51         . "$HOME/.mavenrc"
52     fi
53 # OS specific support. $var _must_ be set to either
54 # true or false.
55 cygwin=false;
56 darwin=false;
57 mingw=false
58 case "`uname`" in
59     CYGWIN*) cygwin=true ;;
60     MINGW*) mingw=true;;
61     Darwin*) darwin=true
62         # Use /usr/libexec/java_home if available,
63         # otherwise fall back to /Library/Java/Home
64         # See https://developer.apple.com/library/mac/qa/
```

```
62 qa1170/_index.html
63     if [ -z "$JAVA_HOME" ]; then
64         if [ -x "/usr/libexec/java_home" ]; then
65             export JAVA_HOME=`/usr/libexec/java_home``
66         else
67             export JAVA_HOME="/Library/Java/Home"
68         fi
69     fi
70 ;;
71 esac
72
73 if [ -z "$JAVA_HOME" ] ; then
74     if [ -r /etc/gentoo-release ] ; then
75         JAVA_HOME=`java-config --jre-home``
76     fi
77 fi
78
79 if [ -z "$M2_HOME" ] ; then
80     ## resolve links - $0 may be a link to maven's
81     ## home
82     PRG="$0"
83
84     # need this for relative symlinks
85     while [ -h "$PRG" ] ; do
86         ls=`ls -ld "$PRG"`
87         link=`expr "$ls" : '.*-> \(.*\)$'`
88         if expr "$link" : '/.*' > /dev/null; then
89             PRG="$link"
90         else
91             PRG=`dirname "$PRG"`/$link
92         fi
93     done
94
95     saveddir=`pwd`
96
97     M2_HOME=`dirname "$PRG`/..
98
99     # make it fully qualified
100    M2_HOME=`cd "$M2_HOME" && pwd`/
101    cd "$saveddir"
```

```

102  # echo Using m2 at $M2_HOME
103 fi
104
105 # For Cygwin, ensure paths are in UNIX format before
   anything is touched
106 if $cygwin ; then
107   [ -n "$M2_HOME" ] &&
108   M2_HOME=`cygpath --unix "$M2_HOME"`
109   [ -n "$JAVA_HOME" ] &&
110   JAVA_HOME=`cygpath --unix "$JAVA_HOME"`
111   [ -n "$CLASSPATH" ] &&
112   CLASSPATH=`cygpath --path --unix "$CLASSPATH"`
113 fi
114
115 # For Mingw, ensure paths are in UNIX format before
   anything is touched
116 if $mingw ; then
117   [ -n "$M2_HOME" ] &&
118   M2_HOME="`cd \"$M2_HOME\"; pwd`"
119   [ -n "$JAVA_HOME" ] &&
120   JAVA_HOME="`cd \"$JAVA_HOME\"; pwd`"
121 fi
122
123 if [ -z "$JAVA_HOME" ]; then
124   javaExecutable=`which javac`
125   if [ -n "$javaExecutable" ] && ! [ "`expr \"\$javaExecutable\" : '\$([^\$]*\$)'`" = "no" ]; then
126     # readlink(1) is not available as standard on
       Solaris 10.
127     readLink=`which readlink`
128     if [ ! `expr "$readLink" : '\$([^\$]*\$)'` = "no" ];
129     ]; then
130       if $darwin ; then
131         javaHome="`dirname \"$javaExecutable\"`"
132         javaExecutable="`cd \"$javaHome\" && pwd -P`"
133         /javac"
134       else
135         javaExecutable="`readlink -f \"\$javaExecutable\"`"
136       fi
137       javaHome="`dirname \"$javaExecutable\"`"

```

```
136     javaHome=`expr "$javaHome" : '\(.*\)/bin'`  
137     JAVA_HOME="$javaHome"  
138     export JAVA_HOME  
139 fi  
140 fi  
141 fi  
142  
143 if [ -z "$JAVACMD" ] ; then  
144   if [ -n "$JAVA_HOME" ] ; then  
145     if [ -x "$JAVA_HOME/jre/sh/java" ] ; then  
146       # IBM's JDK on AIX uses strange locations for  
       the executables  
147       JAVACMD="$JAVA_HOME/jre/sh/java"  
148     else  
149       JAVACMD="$JAVA_HOME/bin/java"  
150     fi  
151   else  
152     JAVACMD=""`\\unset -f command; \\command -v java`  
     "  
153   fi  
154 fi  
155  
156 if [ ! -x "$JAVACMD" ] ; then  
157   echo "Error: JAVA_HOME is not defined correctly."  
     >&2  
158   echo " We cannot execute $JAVACMD" >&2  
159   exit 1  
160 fi  
161  
162 if [ -z "$JAVA_HOME" ] ; then  
163   echo "Warning: JAVA_HOME environment variable is  
     not set."  
164 fi  
165  
166 CLASSWORLDS_LAUNCHER=org.codehaus.plexus.classworlds  
     .launcher.Launcher  
167  
168 # traverses directory structure from process work  
     directory to filesystem root  
169 # first directory with .mvn subdirectory is  
     considered project base directory
```

```

170 find_maven_basedir() {
171
172     if [ -z "$1" ]
173     then
174         echo "Path not specified to find_maven_basedir"
175         return 1
176     fi
177
178     basedir="$1"
179     wdir="$1"
180     while [ "$wdir" != '/' ] ; do
181         if [ -d "$wdir"/.mvn ] ; then
182             basedir=$wdir
183             break
184         fi
185         # workaround for JBEAP-8937 (on Solaris 10/Sparc
186     )
187         if [ -d "${wdir}" ] ; then
188             wdir=`cd "${wdir}/.."; pwd`
189         fi
190         # end of workaround
191     done
192     echo "${basedir}"
193
194 # concatenates all lines of a file
195 concat_lines() {
196     if [ -f "$1" ] ; then
197         echo "$(tr -s '\n' ' ' < "$1")"
198     fi
199 }
200
201 BASE_DIR=`find_maven_basedir "$(pwd)"`#
202 if [ -z "$BASE_DIR" ] ; then
203     exit 1;
204 fi
205
206 ######
207 # Extension to allow automatically downloading the
# maven-wrapper.jar from Maven-central

```

```

208 # This allows using the maven wrapper in projects
209 # that prohibit checking in binary data.
210 ######
211 ######
212 if [ -r "$BASE_DIR/.mvn/wrapper/maven-wrapper.jar"
213 ]; then
214     if [ "$MVNW_VERBOSE" = true ]; then
215         echo "Found .mvn/wrapper/maven-wrapper.jar"
216     fi
217 else
218     if [ "$MVNW_VERBOSE" = true ]; then
219         echo "Couldn't find .mvn/wrapper/maven-wrapper
220         .jar, downloading it ..."
221     fi
222     if [ -n "$MVNW_REPOURL" ]; then
223         jarUrl="$MVNW_REPOURL/org/apache/maven/wrapper
224         /maven-wrapper/3.1.0/maven-wrapper-3.1.0.jar"
225     else
226         jarUrl="https://repo.maven.apache.org/maven2/
227         org/apache/maven/wrapper/maven-wrapper/3.1.0/maven-
228         wrapper-3.1.0.jar"
229     fi
230     while IFS="=" read key value; do
231         case "$key" in
232             wrapperUrl) jarUrl="$value";
233             break ;;
234         esac
235         done < "$BASE_DIR/.mvn/wrapper/maven-wrapper.
236         properties"
237         if [ "$MVNW_VERBOSE" = true ]; then
238             echo "Downloading from: $jarUrl"
239         fi
240         wrapperJarPath="$BASE_DIR/.mvn/wrapper/maven-
241         wrapper.jar"
242         if $cygwin; then
243             wrapperJarPath=`cygpath --path --windows "
244             $wrapperJarPath"`
245         fi
246
247         if command -v wget > /dev/null; then
248             if [ "$MVNW_VERBOSE" = true ]; then
249                 echo "Found wget ... using wget"

```

```

238         fi
239         if [ -z "$MVNW_USERNAME" ] || [ -z "
$MVNW_PASSWORD" ]; then
240             wget "$jarUrl" -O "$wrapperJarPath" ||
rm -f "$wrapperJarPath"
241         else
242             wget --http-user=$MVNW_USERNAME --http-
password=$MVNW_PASSWORD "$jarUrl" -O "
$wrapperJarPath" || rm -f "$wrapperJarPath"
243         fi
244     elif command -v curl > /dev/null; then
245         if [ "$MVNW_VERBOSE" = true ]; then
246             echo "Found curl ... using curl"
247         fi
248         if [ -z "$MVNW_USERNAME" ] || [ -z "
$MVNW_PASSWORD" ]; then
249             curl -O "$wrapperJarPath" "$jarUrl" -f
250         else
251             curl --user $MVNW_USERNAME:
$MVNW_PASSWORD -O "$wrapperJarPath" "$jarUrl" -f
252         fi
253
254     else
255         if [ "$MVNW_VERBOSE" = true ]; then
256             echo "Falling back to using Java to
download"
257         fi
258         javaClass="$BASE_DIR/.mvn/wrapper/
MavenWrapperDownloader.java"
259         # For Cygwin, switch paths to Windows format
before running javac
260         if $cygwin; then
261             javaClass=`cygpath --path --windows "
$javaClass``
262         fi
263         if [ -e "$javaClass" ]; then
264             if [ ! -e "$BASE_DIR/.mvn/wrapper/
MavenWrapperDownloader.class" ]; then
265                 if [ "$MVNW_VERBOSE" = true ]; then
266                     echo " - Compiling
MavenWrapperDownloader.java ..."

```

```
267          fi
268          # Compiling the Java class
269          ("$JAVA_HOME/bin/javac" "$javaClass"
270      )
271      fi
272      if [ -e "$BASE_DIR/.mvn/wrapper/
MavenWrapperDownloader.class" ]; then
273          # Running the downloader
274          if [ "$MVNW_VERBOSE" = true ]; then
275              echo " - Running
MavenWrapperDownloader.java ..."
276          ("$JAVA_HOME/bin/java" -cp .mvn/
wrapper MavenWrapperDownloader "
$MAVEN_PROJECTBASEDIR")
277      fi
278      fi
279  fi
280 fi
281 ######
282 # End of extension
283 #####
284
285 export MAVEN_PROJECTBASEDIR=${MAVEN_BASEDIR:-"
$BASE_DIR"}
286 if [ "$MVNW_VERBOSE" = true ]; then
287     echo $MAVEN_PROJECTBASEDIR
288 fi
289 MAVEN_OPTS=$(concat_lines "$MAVEN_PROJECTBASEDIR/.
mvn/jvm.config") $MAVEN_OPTS"
290
291 # For Cygwin, switch paths to Windows format before
running java
292 if $cygwin; then
293     [ -n "$M2_HOME" ] &&
294         M2_HOME=`cygpath --path --windows "$M2_HOME"`
295     [ -n "$JAVA_HOME" ] &&
296         JAVA_HOME=`cygpath --path --windows "$JAVA_HOME"
`
```

```
297 [ -n "$CLASSPATH" ] &&
298   CLASSPATH=`cygpath --path --windows "$CLASSPATH"
`  

299 [ -n "$MAVEN_PROJECTBASEDIR" ] &&
300   MAVEN_PROJECTBASEDIR=`cygpath --path --windows "
$MAVEN_PROJECTBASEDIR"`
301 fi
302
303 # Provide a "standardized" way to retrieve the CLI
args that will
304 # work with both Windows and non-Windows executions.
305 MAVEN_CMD_LINE_ARGS="$MAVEN_CONFIG $@"
306 export MAVEN_CMD_LINE_ARGS
307
308 WRAPPER_LAUNCHER=org.apache.maven.wrapper.
MavenWrapperMain
309
310 exec "$JAVACMD" \
311   $MAVEN_OPTS \
312   $MAVEN_DEBUG_OPTS \
313   -classpath "$MAVEN_PROJECTBASEDIR/.mvn/wrapper/
maven-wrapper.jar" \
314   "-Dmaven.home=${M2_HOME}" \
315   "-Dmaven.multiModuleProjectDirectory=${
MAVEN_PROJECTBASEDIR}" \
316   ${WRAPPER_LAUNCHER} $MAVEN_CONFIG "$@"
317
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3           xmlns:xsi="http://www.w3.org/2001/XMLSchema-
4           instance"
5               xsi:schemaLocation="http://maven.apache.org/
6               POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.
7               xsd">
8       <modelVersion>4.0.0</modelVersion>
9
10      <groupId>com.example</groupId>
11      <artifactId>BattleGame</artifactId>
12      <version>1.0-SNAPSHOT</version>
13      <name>BattleGame</name>
14
15      <properties>
16          <project.build.sourceEncoding>UTF-8</project.
17          build.sourceEncoding>
18          <junit.version>5.8.2</junit.version>
19      </properties>
20
21      <dependencies>
22          <dependency>
23              <groupId>org.openjfx</groupId>
24              <artifactId>javafx-controls</artifactId>
25              <version>18.0.1</version>
26          </dependency>
27          <dependency>
28              <groupId>org.openjfx</groupId>
29              <artifactId>javafx-fxml</artifactId>
30              <version>18.0.1</version>
31          </dependency>
32
33          <dependency>
34              <groupId>org.junit.jupiter</groupId>
35              <artifactId>junit-jupiter-api</artifactId
36          >
37              <version>${junit.version}</version>
38              <scope>test</scope>
39          </dependency>
40          <dependency>
41              <groupId>org.junit.jupiter</groupId>
```

```
37          <artifactId>junit-jupiter-engine</
38          artifactId>
39              <version>${junit.version}</version>
40              <scope>test</scope>
41          </dependency>
42      </dependencies>
43
44      <build>
45          <plugins>
46              <plugin>
47                  <groupId>org.apache.maven.plugins</
48                  artifactId>
49                      <artifactId>maven-compiler-plugin</
50                      artifactId>
51                          <version>3.10.1</version>
52                          <configuration>
53                              <source>18</source>
54                              <target>18</target>
55                          </configuration>
56                      </plugin>
57                      <plugin>
58                          <groupId>org.openjfx</groupId>
59                          <artifactId>javafx-maven-plugin</
60                          artifactId>
61                          <version>0.0.8</version>
62                          <executions>
63                              <execution>
64                                  <!-- Default configuration
65 for running with: mvn clean javafx:run -->
66                                  <id>default-cli</id>
67                                  <configuration>
68                                      <mainClass>com.example.
69                                          battlegame/com.example.battlegame.HelloApplication</
70                                      mainClass>
71                                      <launcher>app</launcher>
72                                      <jlinkZipName>app</
73                                      jlinkZipName>
74                                      <jlinkImageName>app</
75                                      jlinkImageName>
76                                      <noManPages>true</
77                                      noManPages>
```

```
68                               <stripDebug>true</
69                               <noHeaderFiles>true</
70                               noHeaderFiles>
71                               </configuration>
72                               </execution>
73                               </executions>
74                               </plugin>
75                               </plugins>
76 </build>
76 </project>
```

```
1 @REM
-----
-----
2 @REM Licensed to the Apache Software Foundation (ASF
) under one
3 @REM or more contributor license agreements. See the
NOTICE file
4 @REM distributed with this work for additional
information
5 @REM regarding copyright ownership. The ASF licenses
this file
6 @REM to you under the Apache License, Version 2.0 (
the
7 @REM "License"); you may not use this file except in
compliance
8 @REM with the License. You may obtain a copy of the
License at
9 @REM
10 @REM      https://www.apache.org/licenses/LICENSE-2.0
11 @REM
12 @REM Unless required by applicable law or agreed to
in writing,
13 @REM software distributed under the License is
distributed on an
14 @REM "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
OF ANY
15 @REM KIND, either express or implied. See the
License for the
16 @REM specific language governing permissions and
limitations
17 @REM under the License.
18 @REM
-----
-----
19
20 @REM
-----
-----
21 @REM Maven Start Up Batch script
22 @REM
23 @REM Required ENV vars:
```

```

24 @REM JAVA_HOME - location of a JDK home dir
25 @REM
26 @REM Optional ENV vars
27 @REM M2_HOME - location of maven2's installed home
dir
28 @REM MAVEN_BATCH_ECHO - set to 'on' to enable the
echoing of the batch commands
29 @REM MAVEN_BATCH_PAUSE - set to 'on' to wait for a
keystroke before ending
30 @REM MAVEN_OPTS - parameters passed to the Java VM
when running Maven
31 @REM e.g. to debug Maven itself, use
32 @REM set MAVEN_OPTS=-Xdebug -Xrunjdwp:transport=
dt_socket,server=y,suspend=y,address=8000
33 @REM MAVEN_SKIP_RC - flag to disable loading of
mavenrc files
34 @REM
-----  

-----  

35
36 @REM Begin all REM lines with '@' in case
MAVEN_BATCH_ECHO is 'on'
37 @echo off
38 @REM set title of command window
39 title %0
40 @REM enable echoing by setting MAVEN_BATCH_ECHO to '
on'
41 @if "%MAVEN_BATCH_ECHO%" == "on" echo %
MAVEN_BATCH_ECHO%
42
43 @REM set %HOME% to equivalent of $HOME
44 if "%HOME%" == "" (set "HOME=%HOMEDRIVE%%HOMEPATH%")
45
46 @REM Execute a user defined script before this one
47 if not "%MAVEN_SKIP_RC%" == "" goto skipRcPre
48 @REM check for pre script, once with legacy .bat
ending and once with .cmd ending
49 if exist "%USERPROFILE%\mavenrc_pre.bat" call "%
USERPROFILE%\mavenrc_pre.bat" %*
50 if exist "%USERPROFILE%\mavenrc_pre.cmd" call "%
USERPROFILE%\mavenrc_pre.cmd" %*

```

```
51 :skipRcPre
52
53 @setlocal
54
55 set ERROR_CODE=0
56
57 REM To isolate internal variables from possible post
      scripts, we use another setlocal
58 @setlocal
59
60 REM === START VALIDATION ===
61 if not "%JAVA_HOME%" == "" goto OkJHome
62
63 echo.
64 echo Error: JAVA_HOME not found in your environment.
      >&2
65 echo Please set the JAVA_HOME variable in your
      environment to match the >&2
66 echo location of your Java installation. >&2
67 echo.
68 goto error
69
70 :OkJHome
71 if exist "%JAVA_HOME%\bin\java.exe" goto init
72
73 echo.
74 echo Error: JAVA_HOME is set to an invalid directory
      . >&2
75 echo JAVA_HOME = "%JAVA_HOME%" >&2
76 echo Please set the JAVA_HOME variable in your
      environment to match the >&2
77 echo location of your Java installation. >&2
78 echo.
79 goto error
80
81 REM === END VALIDATION ===
82
83 :init
84
85 REM Find the project base dir, i.e. the directory
      that contains the folder ".mvn".
```

```
86 @REM Fallback to current working directory if not
87   found.
88 set MAVEN_PROJECTBASEDIR=%MAVEN_BASEDIR%
89 IF NOT "%MAVEN_PROJECTBASEDIR%"==""
90   goto endDetectBaseDir
91 set EXEC_DIR=%CD%
92 set WDIR=%EXEC_DIR%
93 :findBaseDir
94 IF EXIST "%WDIR%\.\mvn" goto baseDirFound
95 cd ..
96 IF "%WDIR%"=="%CD%" goto baseDirNotFound
97 set WDIR=%CD%
98 goto findBaseDir
99
100 :baseDirFound
101 set MAVEN_PROJECTBASEDIR=%WDIR%
102 cd "%EXEC_DIR%"
103 goto endDetectBaseDir
104
105 :baseDirNotFound
106 set MAVEN_PROJECTBASEDIR=%EXEC_DIR%
107 cd "%EXEC_DIR%"
108
109 :endDetectBaseDir
110
111 IF NOT EXIST "%MAVEN_PROJECTBASEDIR%\.\mvn\jvm.config"
112   " goto endReadAdditionalConfig
113 @setlocal EnableExtensions EnableDelayedExpansion
114 for /F "usebackq delims=" %%a in ("%MAVEN_PROJECTBASEDIR%\.\mvn\jvm.config") do set
115   JVM_CONFIG_MAVEN_PROPS=!JVM_CONFIG_MAVEN_PROPS! %%a
116 @endlocal & set JVM_CONFIG_MAVEN_PROPS=%
117   JVM_CONFIG_MAVEN_PROPS%
118
119 SET MAVEN_JAVA_EXE="%JAVA_HOME%\bin\java.exe"
120 set WRAPPER_JAR="%MAVEN_PROJECTBASEDIR%\.\mvn\wrapper
```

```
120 \maven-wrapper.jar"
121 set WRAPPER_LAUNCHER=org.apache.maven.wrapper.
    MavenWrapperMain
122
123 set DOWNLOAD_URL="https://repo.maven.apache.org/
    maven2/org/apache/maven/wrapper/maven-wrapper/3.1.0/
    maven-wrapper-3.1.0.jar"
124
125 FOR /F "usebackq tokens=1,2 delims==" %%A IN ("%_
    MAVEN_PROJECTBASEDIR%\.mvn\wrapper\maven-wrapper.
    properties") DO (
126     IF "%%%A%"=="wrapperUrl" SET DOWNLOAD_URL=%%%B
127 )
128
129 @REM Extension to allow automatically downloading
    the maven-wrapper.jar from Maven-central
130 @REM This allows using the maven wrapper in projects
    that prohibit checking in binary data.
131 if exist %WRAPPER_JAR% (
132     if "%MVNW_VERBOSE%" == "true" (
133         echo Found %WRAPPER_JAR%
134     )
135 ) else (
136     if not "%MVNW_REPOURL%" == "" (
137         SET DOWNLOAD_URL="%MVNW_REPOURL%/org/apache/
            maven/wrapper/maven-wrapper/3.1.0/maven-wrapper-3.1.
            0.jar"
138     )
139     if "%MVNW_VERBOSE%" == "true" (
140         echo Couldn't find %WRAPPER_JAR%,
            downloading it ...
141         echo Downloading from: %DOWNLOAD_URL%
142     )
143
144     powershell -Command "&{`n
145         $webclient = new-object System.Net.
            WebClient;"`n
146         "if (-not ([string]::IsNullOrEmpty('%
            MVNW_USERNAME%') -and [string]::IsNullOrEmpty('%
            MVNW_PASSWORD%'))) {"`n
147             "$webclient.Credentials = new-object System.
```

```
147 Net.NetworkCredential('%MVNW_USERNAME%', '%  
MVNW_PASSWORD%');"^  
148     }"^  
149     "[Net.ServicePointManager]::SecurityProtocol  
= [Net.SecurityProtocolType]::Tls12; $webclient.  
DownloadFile('%DOWNLOAD_URL%', '%WRAPPER_JAR%')"^  
150     }"  
151     if "%MVNW_VERBOSE%" == "true" (  
152         echo Finished downloading %WRAPPER_JAR%  
153     )  
154 )  
155 @REM End of extension  
156  
157 @REM Provide a "standardized" way to retrieve the  
CLI args that will  
158 @REM work with both Windows and non-Windows  
executions.  
159 set MAVEN_CMD_LINE_ARGS=%*  
160  
161 %MAVEN_JAVA_EXE% ^  
162 %JVM_CONFIG_MAVEN_PROPS% ^  
163 %MAVEN_OPTS% ^  
164 %MAVEN_DEBUG_OPTS% ^  
165 -classpath %WRAPPER_JAR% ^  
166 "-Dmaven.multiModuleProjectDirectory=%  
MAVEN_PROJECTBASEDIR%" ^  
167 %WRAPPER_LAUNCHER% %MAVEN_CONFIG% %*  
168 if ERRORLEVEL 1 goto error  
169 goto end  
170  
171 :error  
172 set ERROR_CODE=1  
173  
174 :end  
175 @endlocal & set ERROR_CODE=%ERROR_CODE%  
176  
177 if not "%MAVEN_SKIP_RC%"=="" goto skipRcPost  
178 @REM check for post script, once with legacy .bat  
ending and once with .cmd ending  
179 if exist "%USERPROFILE%\mavenrc_post.bat" call "%  
USERPROFILE%\mavenrc_post.bat"
```

```
180 if exist "%USERPROFILE%\mavenrc_post.cmd" call "%  
USERPROFILE%\mavenrc_post.cmd"  
181 :skipRcPost  
182  
183 @REM pause the script if MAVEN_BATCH_PAUSE is set to  
'on'  
184 if "%MAVEN_BATCH_PAUSE%"=="on" pause  
185  
186 if "%MAVEN_TERMINATE_CMD%"=="on" exit %ERROR_CODE%  
187  
188 cmd /C exit /B %ERROR_CODE%  
189
```

```
1 target/
2 !.mvn/wrapper/maven-wrapper.jar
3 !**/src/main/**/target/
4 !**/src/test/**/target/
5
6 ### IntelliJ IDEA ####
7 .idea/modules.xml
8 .idea/jarRepositories.xml
9 .idea/compiler.xml
10 .idea/libraries/
11 *.iws
12 *.iml
13 *.ipr
14
15 ### Eclipse ####
16 .apt_generated
17 .classpath
18 .factorypath
19 .project
20 .settings
21 .springBeans
22 .sts4-cache
23
24 ### NetBeans ####
25 /nbproject/private/
26 /nbbuild/
27 /dist/
28 /nbdist/
29 /.nb-gradle/
30 build/
31 !**/src/main/**/build/
32 !**/src/test/**/build/
33
34 ### VS Code ####
35 .vscode/
36
37 ### Mac OS ####
38 .DS_Store
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <module org.jetbrains.idea.maven.project.
  MavenProjectsManager.isMavenModule="true" type=
  JAVA_MODULE" version="4">
3   <component name="NewModuleRootManager"
  LANGUAGE_LEVEL="JDK_18">
4     <output url="file://$MODULE_DIR$/target/classes"
  />
5     <output-test url="file://$MODULE_DIR$/target/test-
  classes" />
6     <content url="file://$MODULE_DIR$">
7       <sourceFolder url="file://$MODULE_DIR$/src/main
  /java" isTestSource="false" />
8       <sourceFolder url="file://$MODULE_DIR$/src/main
  /resources" type="java-resource" />
9       <excludeFolder url="file://$MODULE_DIR$/target"
  />
10    </content>
11    <orderEntry type="inheritedJdk" />
12    <orderEntry type="sourceFolder" forTests="false"
  />
13    <orderEntry type="library" name="Maven: org.
  openjfx:javafx-controls:18.0.1" level="project" />
14    <orderEntry type="library" name="Maven: org.
  openjfx:javafx-controls:win:18.0.1" level="project"
  />
15    <orderEntry type="library" name="Maven: org.
  openjfx:javafx-graphics:18.0.1" level="project" />
16    <orderEntry type="library" name="Maven: org.
  openjfx:javafx-graphics:win:18.0.1" level="project"
  />
17    <orderEntry type="library" name="Maven: org.
  openjfx:javafx-base:18.0.1" level="project" />
18    <orderEntry type="library" name="Maven: org.
  openjfx:javafx-base:win:18.0.1" level="project" />
19    <orderEntry type="library" name="Maven: org.
  openjfx:javafx-fxml:18.0.1" level="project" />
20    <orderEntry type="library" name="Maven: org.
  openjfx:javafx-fxml:win:18.0.1" level="project" />
21    <orderEntry type="library" scope="TEST" name=
  Maven: org.junit.jupiter:junit-jupiter-api:5.8.2"
```

```
21 level="project" />
22     <orderEntry type="library" scope="TEST" name=
    Maven: org.opentest4j:opentest4j:1.2.0" level=
    project" />
23     <orderEntry type="library" scope="TEST" name=
    Maven: org.junit.platform:junit-platform-commons:1.8.
    2" level="project" />
24     <orderEntry type="library" scope="TEST" name=
    Maven: org.apiguardian:apiguardian-api:1.1.2" level=
    project" />
25     <orderEntry type="library" scope="TEST" name=
    Maven: org.junit.jupiter:junit-jupiter-engine:5.8.2"
    level="project" />
26     <orderEntry type="library" scope="TEST" name=
    Maven: org.junit.platform:junit-platform-engine:1.8.2
    " level="project" />
27 </component>
28 </module>
```