@REM ----------------------------------------------------------------------------

@REM Licensed to the Apache Software Foundation (ASF) under one

@REM or more contributor license agreements. See the NOTICE file

@REM distributed with this work for additional information

@REM regarding copyright ownership. The ASF licenses this file

@REM to you under the Apache License, Version 2.0 (the

@REM "License"); you may not use this file except in compliance

@REM with the License. You may obtain a copy of the License at

@REM

@REM https://www.apache.org/licenses/LICENSE-2.0

@REM

@REM Unless required by applicable law or agreed to in writing,

@REM software distributed under the License is distributed on an

@REM "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

@REM KIND, either express or implied. See the License for the

@REM specific language governing permissions and limitations

@REM under the License.

@REM ----------------------------------------------------------------------------

@REM ----------------------------------------------------------------------------

@REM Maven Start Up Batch script

@REM

@REM Required ENV vars:

@REM JAVA\_HOME - location of a JDK home dir

@REM

@REM Optional ENV vars

@REM M2\_HOME - location of maven2's installed home dir

@REM MAVEN\_BATCH\_ECHO - set to 'on' to enable the echoing of the batch commands

@REM MAVEN\_BATCH\_PAUSE - set to 'on' to wait for a keystroke before ending

@REM MAVEN\_OPTS - parameters passed to the Java VM when running Maven

@REM e.g. to debug Maven itself, use

@REM set MAVEN\_OPTS=-Xdebug -Xrunjdwp:transport=dt\_socket,server=y,suspend=y,address=8000

@REM MAVEN\_SKIP\_RC - flag to disable loading of mavenrc files

@REM ----------------------------------------------------------------------------

@REM Begin all REM lines with '@' in case MAVEN\_BATCH\_ECHO is 'on'

@echo off

@REM set title of command window

title %0

@REM enable echoing by setting MAVEN\_BATCH\_ECHO to 'on'

@if "%MAVEN\_BATCH\_ECHO%" == "on" echo %MAVEN\_BATCH\_ECHO%

@REM set %HOME% to equivalent of $HOME

if "%HOME%" == "" (set "HOME=%HOMEDRIVE%%HOMEPATH%")

@REM Execute a user defined script before this one

if not "%MAVEN\_SKIP\_RC%" == "" goto skipRcPre

@REM check for pre script, once with legacy .bat ending and once with .cmd ending

if exist "%USERPROFILE%\mavenrc\_pre.bat" call "%USERPROFILE%\mavenrc\_pre.bat" %\*

if exist "%USERPROFILE%\mavenrc\_pre.cmd" call "%USERPROFILE%\mavenrc\_pre.cmd" %\*

:skipRcPre

@setlocal

set ERROR\_CODE=0

@REM To isolate internal variables from possible post scripts, we use another setlocal

@setlocal

@REM ==== START VALIDATION ====

if not "%JAVA\_HOME%" == "" goto OkJHome

echo.

echo Error: JAVA\_HOME not found in your environment. >&2

echo Please set the JAVA\_HOME variable in your environment to match the >&2

echo location of your Java installation. >&2

echo.

goto error

:OkJHome

if exist "%JAVA\_HOME%\bin\java.exe" goto init

echo.

echo Error: JAVA\_HOME is set to an invalid directory. >&2

echo JAVA\_HOME = "%JAVA\_HOME%" >&2

echo Please set the JAVA\_HOME variable in your environment to match the >&2

echo location of your Java installation. >&2

echo.

goto error

@REM ==== END VALIDATION ====

:init

@REM Find the project base dir, i.e. the directory that contains the folder ".mvn".

@REM Fallback to current working directory if not found.

set MAVEN\_PROJECTBASEDIR=%MAVEN\_BASEDIR%

IF NOT "%MAVEN\_PROJECTBASEDIR%"=="" goto endDetectBaseDir

set EXEC\_DIR=%CD%

set WDIR=%EXEC\_DIR%

:findBaseDir

IF EXIST "%WDIR%"\.mvn goto baseDirFound

cd ..

IF "%WDIR%"=="%CD%" goto baseDirNotFound

set WDIR=%CD%

goto findBaseDir

:baseDirFound

set MAVEN\_PROJECTBASEDIR=%WDIR%

cd "%EXEC\_DIR%"

goto endDetectBaseDir

:baseDirNotFound

set MAVEN\_PROJECTBASEDIR=%EXEC\_DIR%

cd "%EXEC\_DIR%"

:endDetectBaseDir

IF NOT EXIST "%MAVEN\_PROJECTBASEDIR%\.mvn\jvm.config" goto endReadAdditionalConfig

@setlocal EnableExtensions EnableDelayedExpansion

for /F "usebackq delims=" %%a in ("%MAVEN\_PROJECTBASEDIR%\.mvn\jvm.config") do set JVM\_CONFIG\_MAVEN\_PROPS=!JVM\_CONFIG\_MAVEN\_PROPS! %%a

@endlocal & set JVM\_CONFIG\_MAVEN\_PROPS=%JVM\_CONFIG\_MAVEN\_PROPS%

:endReadAdditionalConfig

SET MAVEN\_JAVA\_EXE="%JAVA\_HOME%\bin\java.exe"

set WRAPPER\_JAR="%MAVEN\_PROJECTBASEDIR%\.mvn\wrapper\maven-wrapper.jar"

set WRAPPER\_LAUNCHER=org.apache.maven.wrapper.MavenWrapperMain

set DOWNLOAD\_URL="https://repo.maven.apache.org/maven2/org/apache/maven/wrapper/maven-wrapper/3.1.0/maven-wrapper-3.1.0.jar"

FOR /F "usebackq tokens=1,2 delims==" %%A IN ("%MAVEN\_PROJECTBASEDIR%\.mvn\wrapper\maven-wrapper.properties") DO (

IF "%%A"=="wrapperUrl" SET DOWNLOAD\_URL=%%B

)

@REM Extension to allow automatically downloading the maven-wrapper.jar from Maven-central

@REM This allows using the maven wrapper in projects that prohibit checking in binary data.

if exist %WRAPPER\_JAR% (

if "%MVNW\_VERBOSE%" == "true" (

echo Found %WRAPPER\_JAR%

)

) else (

if not "%MVNW\_REPOURL%" == "" (

SET DOWNLOAD\_URL="%MVNW\_REPOURL%/org/apache/maven/wrapper/maven-wrapper/3.1.0/maven-wrapper-3.1.0.jar"

)

if "%MVNW\_VERBOSE%" == "true" (

echo Couldn't find %WRAPPER\_JAR%, downloading it ...

echo Downloading from: %DOWNLOAD\_URL%

)

powershell -Command "&{"^

"$webclient = new-object System.Net.WebClient;"^

"if (-not ([string]::IsNullOrEmpty('%MVNW\_USERNAME%') -and [string]::IsNullOrEmpty('%MVNW\_PASSWORD%'))) {"^

"$webclient.Credentials = new-object System.Net.NetworkCredential('%MVNW\_USERNAME%', '%MVNW\_PASSWORD%');"^

"}"^

"[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12; $webclient.DownloadFile('%DOWNLOAD\_URL%', '%WRAPPER\_JAR%')"^

"}"

if "%MVNW\_VERBOSE%" == "true" (

echo Finished downloading %WRAPPER\_JAR%

)

)

@REM End of extension

@REM Provide a "standardized" way to retrieve the CLI args that will

@REM work with both Windows and non-Windows executions.

set MAVEN\_CMD\_LINE\_ARGS=%\*

%MAVEN\_JAVA\_EXE% ^

%JVM\_CONFIG\_MAVEN\_PROPS% ^

%MAVEN\_OPTS% ^

%MAVEN\_DEBUG\_OPTS% ^

-classpath %WRAPPER\_JAR% ^

"-Dmaven.multiModuleProjectDirectory=%MAVEN\_PROJECTBASEDIR%" ^

%WRAPPER\_LAUNCHER% %MAVEN\_CONFIG% %\*

if ERRORLEVEL 1 goto error

goto end

:error

set ERROR\_CODE=1

:end

@endlocal & set ERROR\_CODE=%ERROR\_CODE%

if not "%MAVEN\_SKIP\_RC%"=="" goto skipRcPost

@REM check for post script, once with legacy .bat ending and once with .cmd ending

if exist "%USERPROFILE%\mavenrc\_post.bat" call "%USERPROFILE%\mavenrc\_post.bat"

if exist "%USERPROFILE%\mavenrc\_post.cmd" call "%USERPROFILE%\mavenrc\_post.cmd"

:skipRcPost

@REM pause the script if MAVEN\_BATCH\_PAUSE is set to 'on'

if "%MAVEN\_BATCH\_PAUSE%"=="on" pause

if "%MAVEN\_TERMINATE\_CMD%"=="on" exit %ERROR\_CODE%

cmd /C exit /B %ERROR\_CODE%