How to build STAMP Workbench using commands of PowerShell

The procedure how to build and run STAMP Workbench using commands of Windows PowerShell is written on the following. We did these operation checks by the following environment.

- Windows 10 version 20H2
- AdoptOpenJDK (or Eclipse Adoptium OpenJDK) version 1.8.0_292
- Apache Maven version 3.6.3
- Windows PowerShell version 5.1

Install STAMP Workbench on your PC

- Download STAMP Workbench version 2.0.0 from here.
- Install STAMP Workbench on your PC. STAMP Workbench's default folder to install is C:\Program Files\stampworkbench [*1]

Prepare STAMP Workbench project file which includes source codes and build information files

- Download STAMP Workbench project file "stamp-project-2_0_0.zip" which includes source code files and build information files from here.
- Extract zipped file "stamp-project-2_0_0.zip" to any folder [*2] on your PC.

Install AdoptOpenJDK (or Eclipse Adoptium OpenJDK)

- Install JDK (Java Development Kit) on your PC to execute Apache Maven.
- Add the path to JDK to your "JAVA_HOME" environment variable on Windows.

Install Apache Maven

Install a project management tool "Apache Maven" to build and run STAMP Workbench. The procedure how to install Maven is below.

- Download Maven from this download site. http://maven.apache.org/download.html

- Extract downloaded file to any folder [*3] on your PC.

- Add the path to Maven [*3] to your "M2_HOME" environment variable on Wondows.

- Set ${M2_HOME} to your PATH environment variable on windows.

- Execute PowerShell and run command "mvn -v" to confirm that Maven works.

Compile source codes and build STAMP Workbench by command of PowerShell

- Run PowerShell on your PC, and change current folder to the STAMP Workbench project files' folder [*2]. And then change current folder to stamp-project\projects again.
- Set variable for PowerShell to execute build command.
    - $stampworkbench = "STAMP Workbench's folder [*1]"
    - $any_path = "any path to Maven's repository folder. e.g. ${env:M2_HOME}/repo" Maven will reserve downloaded files and built jar in this repository.
- Run following command.
    - mvn "-Dstampworkbench=${stampworkbench}" "-Dmaven.repo.local=${any_path}" -s "${env:M2_HOME}/conf/settings.xml" clean install

- If following logs are shown to your display, building STAMP Workbench is successful.

[INFO] --
[INFO] Reactor Summary:
[INFO]
[INFO] STAMP Workbench - Root POM .... SUCCESS [0.629 s]
[INFO] STAMP Workbench - STAMP/STPA Metamodel .... SUCCESS [10.291 s]
[INFO] STAMP Workbench - STAMP/STPA Notation Metamodel .... SUCCESS [5.644 s]
[INFO] STAMP Workbench - STAMP/STPA Metamodel Edit .... SUCCESS [5.051 s]
[INFO] STAMP Workbench - STAMP/STPA Notation Metamodel Edit SUCCESS [4.688 s]
[INFO] STAMP Workbench - STAMP/STPA Notation Editor .... SUCCESS [3.728 s]
[INFO] STAMP Workbench - UI .... SUCCESS [9.307 s]
[INFO] STAMP Workbench - Application Bootstrap .... SUCCESS [2.204 s]
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --

Launch STAMP Workbench from PowerShell

- Run STAMP Workbench with following command of PowerShell.
  - mvn "-Dstampworkbench=${stampworkbench}" "-Dmaven.repo.local=${any_path}" exec:java -f net.astah.stpa.stamp.app/pom.xml

How to start directly without depending on Maven

- Replacing the JAR files in the STAMP Workbench folder with JAR files in the build result will allow you to launch STAMP Workbench directly without relying on Maven.
  - ➢ Replace each project's build artifact (`* / target / *. Jar`) with JAR files of the same name under the STAMP Workbench installation folder. However, note that `stampworkbench.jar` has been replaced with another name in the build artifact,` net.astah.stpa.stamp.app/target/*.jar`.
- The startup `stampworkbench.jar` needs to be replaced with` META-INF / MANIFEST.MF` by the following method.
  - ➢ Extract the above file from the JAR included in STAMP Workbench and update the JAR in the build result.
  - ➢ JAR can be updated with `jar umf $ {extracted MANIFEST.MF} stampworkbench.jar -C $ {empty directory}` etc.