Presentation with/of the `latex-maven-plugin`

E. Reissner

September 22, 2024 ernst.reissner@simuline.eu

This is some additional text

Purpose of these documents

The purpose of this presentation is twofold:

- Give an overview over the plugin. But:
  - The single official description of the plugin is the manual [Rei].
  - The project site gives already an overview.
  - The content of this presentation is updated only by need.
- Demonstrate that the plugin can compile a presentation using the `beamer` document class and a handout written as `beamerarticle`.
  - In fact, when citing this presentation as [Rei23], this refers to both.
  - The presentation is about the plugin.
  - The handout adds information on how to write a beamer presentation.
  - Both documents turn `beamer` class and package `beamerarticle` into preferred usage ensuring tests.

Ant and Maven

Automatically creates documents from LaTeX sources during the Maven `site` phase and in an ant run.
This comprises running basic tools like `lualatex` and `bibtex` and rerunning them by need.

Supported IO

Supports

- many output formats like PDF, DVI, HTML, DOCX, RTF, TXT and others
- many graphical input formats like PNG, MP, FIG, gnuplot; also provides a separate goal creating them, `grp`
- document classes including `book`, `article`, ..., `beamer`, `beamerarticle` for presentations, `leaflet` and the letter class `scrlttr2`.
- bibliography, index/indices, glossary/glossaries and embedded code

Checks

Checks

- sources with `chktex` and logs the results in target and goal `chk`
- versions of used tools via goal `vrs`
- log files detecting errors and warnings
- Check of return codes
- further excessive logging of warnings and errors
- whether a document could have been reproduced, by demand

Orchestration and document development

Orchestration of various tools detecting need for execution e.g. of `bibtex` including `rerunfilecheck` for `lualatex` and friends.
Supports document development, mainly by cooperating with editor, viewer and with other tools in the build chain. Offers

- various scripts, most notably an installation script for extensions of VS Code.
- configuration file `.chktexrc` for `chktex`.
- configuration file `.latexmkrc` for `latexmk` synchronized with the configuration.
- header files like `header.tex` to unify packages loaded by LaTeX main files.

Planned Features

- Support `biber` replacing `bibtex` as preferred tool
- Support `xindy` replacing `makeindex` as preferred tool
- Support `bib2gls` replacing `makeglossary` as preferred tool
- Execute `glosstex` if needed
- Usage of the `multibib` macros
- ...

Goals

An overview is on the goals page.
Besides goals referring to creating output of a given type like `pdf`, `odt` and `html`, there is a goal `cfg`, which allows configuring various output formats as `targets`.
One further goal, `chk` to perform a check with `chktex` is allowed among the `targets`.
The other goals cannot serve as targets:

- `vrs` to check versions of tools,
- `grp` to create graphic files speeding up use of `latexmk` and
- `inj` to inject useful files like headers or config files like `.latexmkrc`.

Usage: Set up repository

As this is the main usage of this software, usage refers to usage as a maven plugin. Since it did not yet make it into the repository maven central, it can be accessed only giving the source repository explicitly:

```
<project>
  ...
  <repositories>
  <repository>
    <id>publicRepoAtSimuline</id>
    <name>repo at simuline</name>
    <url>https://www.simuline.eu/RepositoryMaven</url>
  </repository>
  ...
</project>
```

Usage: Set up life cycles

Also, it must be ensured that the plugin is executed when building the application and when building the according site. This is a side effect of configuring the `maven-jxr-plugin`.

```
<project>
...
<reporting>
  <plugins>
    ...
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jxr-plugin</artifactId>
      <version>3.3.0</version>
    </plugin>
    ...
  </plugins>
</reporting>
</project>
```

Usage: Simple Example

Given this, usage can be as simple as setting

```
<project>
  ...
  <build>
    <plugins>
      <plugin>
        <groupId>eu.simuline.m2latex</groupId>
        <artifactId>latex-maven-plugin</artifactId>
        <version>2.0-SNAPSHOT</version>
        <executions>
          <execution>
            <phase>site</phase>
            <goals><goal>pdf</goal></goals>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
  ...
</project>
```

Usage: adding executions for build and clean

Configuring executions for build and clean looks like this.

```
<plugin>
  <groupId>eu.simuline.m2latex</groupId>
  <artifactId>latex-maven-plugin</artifactId>
  <version>2.0-SNAPSHOT</version>
    ...
  <executions>
    <execution>
      <id>process-latex-sources</id>
      <!-- chk, dvi, pdf, html, odt, docx, rtf, txt -->
      <goals><goal>cfg</goal></goals> <!-- phase>site</phase-->
    </execution>
    <execution>
      <id>clear-latex-sources</id>
      <goals><goal>clr</goal></goals> <!-- phase>clean</phase-->
    </execution>
    ...
  </executions>
</plugin>
```

Usage: adding further executions

It is recommended to add executions for injection and version check also

```
<plugin>
  <groupId>eu.simuline.m2latex</groupId>
  <artifactId>latex-maven-plugin</artifactId>
  <version>2.0-SNAPSHOT</version>
    ...
  <executions>
    ...
    <execution>
      <id>inject-files</id>
      <goals><goal>inj</goal></goals> <!-- phase>validate</phase-->
      <configuration>
      <injections>latexmkrc,chktexrc,header</injections>
    </configuration>
    </execution>
    <execution>
      <id>validate-converters</id>
      <goals><goal>vrs</goal></goals> <!-- phase>validate</phase-->
      <configuration>
        <versionsWarnOnly>true</versionsWarnOnly>
      </configuration>
    </execution>
  </executions>
</plugin>
```

Usage: more complex examples

Note that even in this complex example, further configuration is only sketched, which is justified because there is in general little deviation from the default to be mentioned explicitly. So, more complex examples are rare. For a full description of configuration, see the manual [Rei], Chapter 6.

Index I

References I


E. Reißner.
Manual for the latex-maven-plugin and for an according ant-task, Version X.Y.
The current version is vailable at
http://www.simuline.eu/LatexMavenPlugin/manualLMP.pdf.


E. Reißner.
Presentation with/of the latex-maven-plugin.
presentation available at http://www.simuline.eu/LatexMavenPlugin/docClasses/useBeamerPres.pdf, handout at http://www.simuline.eu/LatexMavenPlugin/docClasses/useBeamerHandout.pdf, 10 2023.
Comprises both, presentation and handout.