

clean local repository maven plugin

The «clean local repository plugin» provides the capability to delete and parametrized deletion of «artifacts» from the local maven repository.

It's well known that in a continuous integration environment, the local maven repository is the theatre of a growing file system consumption generated by the continuous and incremental build of «snapshot» and «released» libraries.

This behavior is not really known during personal development and compilation of individual projects but it's especially prejudicial in industrial context with build of many large and parallelized projects. In such cases, this plugin allows selection and auto deletion of unnecessary artifacts to moderate your file system consumption.

Default behaviour is to keep only the latest « snapshot » version of a project libraries and metadata but to fit your needs, you can refer to the following goal and usage sections to find more on this plugin options.

Goals Overview

The clean local repository plugin has two main goals:

- `cleanlocalrepository:list` analyzes the pom.xml of the project, determines and lists which artifact can be deleted regarding the given options.
- `cleanlocalrepository:clean` tells Maven to clear project artifacts out of the local repository regarding the given options.

Basic usage

The default behavior of the plugin is to first resolve the current project artifacts tree, then delete all versions excepting the current one from the local repository :

```
mvn cleanlocalrepository:clean
```

Options

Here is the optional parameters list allowing to control advanced behavior of the plugin :

Name	Type	Since	Description
<code>deleteProjectSnapshot</code>	boolean	1.1	Controls whether the plugin tries to delete the current project snapshot versions from the local repository regarding to the <code>snapshotRetentionDelay</code> and <code>snapshotVersionsRetention</code> options. Default value is: true.
<code>deleteProjectRelease</code>	boolean	1.1	Controls whether the plugin tries to delete the current project released versions from the local repository regarding to the <code>releaseRetentionDelay</code> and <code>releaseVersionsRetention</code> options. Default value is: true.
<code>deleteAllSnapshots</code>	boolean	1.1	Controls whether the plugin tries to delete all the snapshot versions from the local repository regarding to the <code>snapshotRetentionDelay</code> and <code>snapshotVersionsRetention</code> options. Default value is: false.
<code>snapshotRetentionDelay</code>	int	1.0	

			Controls the expiration delay (in days) before deleting a snapshot version. Default value is: 7
snapshotVersionsRetention	int	1.0	Controls the number of different snapshot versions to keep without deletion. Default value is: 1
releaseRetentionDelay	int	1.0	Controls the expiration delay (in days) before deleting a release version. Default value is: 7
releaseVersionsRetention	int	1.0	Controls the number of different release versions to keep without deletion. Default value is: 1
deleteFromRegularExpression	RegExp	1.0	Delete all files from the local repository which match with the given regular expression (@see Pattern definition). The selection pattern is applied on the file system path from the root of the maven local repository. e.g. DeleteFromRegularExpression = ^com.* will delete all artifacts and metadata published on the \${localMavenRepository}/com path.
deleteEmptyFolders	boolean	1.0	Controls whether the plugin have to delete all the empty folders from the local maven repository Default value is: true
executeDeleteOnExist	boolean	1.0	Controls whether delete command will be attempted only for normal termination of the virtual machine Default value is: true
deleteWholeLocalRepository	boolean	1.0	Controls whether the plugin have to purge the whole local maven repository. Default value is: false

Usage

Brief example on how to use the clean local repository plugin goals. Refer to the options section to feet this given configuration to your need.

Configure the plugin something like this if you intend to bind it to execute along with your build:

```
<project>
[...]
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>clean-local-repository-plugin</artifactId>
```

```

    <version>1.0</version>
    <executions>
      <execution>
        <id>cleanlocalrepository </id>
        <phase>process-sources</phase>
        <goals>
          <goal>clean</goal>
        </goals>
        <configuration>
          <deleteRelease>>false</deleteRelease>
          <snapshotRetentionDelay>3</snapshotRetentionDelay>
          <!-- other configurations here -->
        </configuration>
      </execution>
    </executions>
  </plugin>
</plugins>
</build>
[...]
```

If you intend to configure this mojo for execution on the command line using:

```
mvn clean-local-repository:clean -Dclean-local-repository.deleteCurrentRelease=false -Dclean-local-repository.snapshotRetentionDelay=3
```

Other configuration

If you need a flexible way to resolve artifact for deletion, you can use the [deleteFromRegularExpression](#) options to configure deletion instruction based on regular expression.

Please refer to the corresponding option section to configure this parameter. You can also see the complete Pattern documentation on the [Java Pattern API](#).

Warning

Because local maven repository can be a shared resource, be careful of the execution context and options of this plugin in a multi-projects environment. Especially for the [deleteWholeLocalRepository](#) option which can generate side effect if deletion is executed simultaneously with another project compilation.

To prevent build errors in a continuous integration context like Jenkins, we highly recommend the usage and coupling of CI plugins like JobBlocker for creating administration task based on these options.

Resources

Here are some links that provide more reference regarding local maven repository and dependencies :

- [Configuring your local maven repository](#)
- [Dependency Mechanism](#)