Usage

Below are the different goals and the minimalist configurations of the Help Plugin.

The help:active-profiles Goal

The active-profiles goal is used to discover which profiles have been applied to the projects currently being built. For each project in the build session, it will output a list of profiles which have been applied to that project, along with the source of the profile (POM, settings.xml or profiles.xml).

You can execute this goal using the following command:

```
# mvn help:active-profiles
```

Note: you could also use the output parameter to redirect output to a file.

The help:all-profiles Goal

The all-profiles goal is used to discover all available profiles under the current project.

You can execute this goal using the following command:

```
# mvn help:all-profiles
```

Note: you could also use the output parameter to redirect output to a file.

The help:describe Goal

The describe goal is used to discover information about Maven plugins. Given either a plugin or a groupId, an artifactId and optionally a version, the mojo will lookup that plugin and output details about it. If the user also specifies which mojo to describe, the describe mojo will limit output to the details of that mojo, including parameters.

You can execute this goal using the following command:

```
# mvn help:describe -DgroupId=org.somewhere -DartifactId=some-plugin -Dversion=0.0.0
```

Note: you could also use the output parameter to redirect output to a file.

Refer to Configuring Describe Goal for more information about its configuration.

The help:effective-pom Goal

The effective-pom goal is used to make visible the POM that results from the application of interpolation, inheritance and active profiles. It provides a useful way of removing the guesswork about just what ends up in the POM that Maven uses to build your project. It will iterate over all projects in the current build session, printing the effective POM for each.

You can execute this goal using the following command:

```
# mvn help:effective-pom
```

Note: you could also use the output parameter to redirect output to a file.

The help:effective-settings Goal

The effective-settings goal is used to view the settings that Maven actually uses to run the build. These settings are a result of merging the global file with the user's file, with the user's file taking precedence.

You can execute this goal using the following command:

```
# mvn help:effective-settings
```

Note: you could also use the output parameter to redirect output to a file.

The help:system Goal

The system goal is used to view the system information like system properties and environment variables.

You can execute this goal using the following command:

```
# mvn help:system
```

Note: you could also use the output parameter to redirect output to a file.

The expressions Goal

When you develop a new Maven Plugin, you could need to use some expressions given by Maven. To list them, just call the expressions goal:

```
# mvn help:expressions
...
[INFO] [help:expressions]
[INFO] Maven supports the following Plugin expressions:

${project}: No description available.

${settings}: No description available.

${session}: No description available.

${plugin}: No description available.

${rootless}: No description available.

${settings.interactiveMode}: If false, flags the system to skip prompting the
```

```
user for any information, or holding up the build waiting for any input. NOTE:
It's also possible to switch to batch (ie. non-interactive) mode using the '-B'
command-line option.
...
```

Note: you could also use the `output` parameter to redirect output to a file.

The `help:evaluate` Goal

You could use this interactive goal to evaluate some Maven expressions. To do it, just call the help:evaluate goal:

```
# mvn help:evaluate -Dartifact=org.apache.maven.plugins:maven-help-plugin
...
[INFO] [help:evaluate]
[INFO] Enter the Maven expression i.e. ${project.groupId} or 0 to exit?:
${project.artifactId}
[INFO]
maven-help-plugin
[INFO] Enter the Maven expression i.e. ${project.groupId} or 0 to exit?:
${project.none}
[INFO]
null object or invalid expression
...
```

The `artifact` parameter refers to ask expressions on the artifact POM. If omitted, the evaluate goal uses the current pom.

You could ask for all Maven expressions listed by `help:expressions` goal, i.e. ${project}, ${project.licenses}, ${settings}, ${settings.profiles}.