Apache ANT

SSHEXEC

```
<project name="test" default="test">

  <target name="test">

          <sshexec host="192.168.242.134" username="root" password="root123"
command="touch file5" trust="true" />

<!-- user credentials through comnnad prompt -->
  <!--<sshexec host="192.168.242.134"
  username="${username}"
  password="${password}"
  command="touch somefile"
  trust="true"/> -->


  </target>

</project>
```

SCP

```
<project name="test" default="test">

  <target name="test">

          <scp file="build/jar/HelloWorld.jar"
todir="root@192.168.200.134:/dir1" password="password" trust="true" />

  </target>

</project>
```

Data Types

1. fileset

- The fileset data types represents a collection of files. It is used as a filter to include or exclude files that match a particular pattern.
- For example, refer the following code. Here, the src attribute points to the source folder of the project.
- The fileset selects all .java files in the source folder except those contain the word 'Stub'. The case-sensitive filter is applied to the fileset which means a file with the name Samplestub.java will not be excluded from the fileset.

```
<fileset dir="${src}" casesensitive="yes">
   <include name="**/*.java"/>
   <exclude name="**/*Stub*"/>
</fileset>
```

2.Pattern set

A pattern set is a pattern that allows to filter files or folders easily based on certain patterns. Patterns can be created using the following meta characters:

- ? - Matches one character only.
- * - Matches zero or many characters.
- ** - Matches zero or many directories recursively.

The following example depicts the usage of a pattern set.

```
<patternset id="java.files.without.stubs">

   <include name="src/**/*.java"/>

   <exclude name="src/**/*Stub*"/>

</patternset>
```

The patternset can then be reused with a fileset as follows:

```
<fileset dir="${src}" casesensitive="yes">
   <patternset refid="java.files.without.stubs"/>
</fileset>
```

3. File list

The filelist data type is similar to the file set except the following differences:

- filelist contains explicitly named lists of files and it does not support wild cards.
- filelist data type can be applied for existing or non-existing files.

Let us see the following example of the filelist data type. Here, the attribute **webapp.src.folder** points to the web application source folder of the project.

```
<filelist id="config.files" dir="${webapp.src.folder}">
   <file name="applicationConfig.xml"/>
   <file name="faces-config.xml"/>
   <file name="web.xml"/>
   <file name="portlet.xml"/>
</filelist>
```