

Create a release using CVSNT

This tutorial describes how to create a release using CVSNT. This tutorial assumes that the CVS Client and Maven are installed on your machine. If CVS Client is not installed you can get it from here:

- [CVS Client](#)

Setting Up Your Project

Defining Your CVS Project in Maven POM

You will need to add the *SCM* and *Release* plug-in to your POM.

pom.xml

```
<project>
...
<plugins>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-scm-plugin</artifactId>
    <version>1.2</version>
    <configuration>
      <connectionType>developerconnection</connectionType>
    </configuration>
  </plugin>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-release-plugin</artifactId>
    <version>2.0-beta-9</version>
    <configuration>

<tagBase>scm:cvs:sserver:${cvs.username}@host:/cvsrepo:myproject/core</tagBase>
      </configuration>
    </plugin>
  </plugins>
...
</project>
```

You will need to change the *tagBase* for your cvs repository.

Now you need to define the *SCM* connector in your Maven POM.

pom.xml

```
<project>
...
  <scm>

  <developerConnection>scm:cvs:sserver:${cvs.username}@host:/cvsrepo:myproject/core</dev
  eloperConnection>
    <url>http://host/ViewVC/viewvc.cgi/</url>
  </scm>
...
</project>
```

Enable Maven to Deploy Releases and Snapshot

When maven does the final step of performing the release, it will push the release to your organizations maven repository.

In order for maven to deploy to your organization's maven repository you have to have a *settings.xml* file in the *.m2* folder under your home directory. Here is an example of how one can be set up.

settings.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <interactiveMode>true</interactiveMode>
  <servers>
    <server>
      <id>organization_releases</id>
      <username>username</username>
      <password>password</password>
    </server>
    <server>
      <id>organization_snapshots</id>
      <username>username</username>
      <password>password</password>
    </server>
  </servers>
  <profiles>
    <profile>
      <id>username</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <properties>
        <cvsexecutable>username</cvsexecutable>
      </properties>
    </profile>
  </profiles>
</settings>
```

Creating the Release

We are going to do this from the command line. First we will need to check out the project.

Command Prompt

```
cvs -d :sserver:username@host:/cvsrepo login
```

Once successfully logged in we will check out the project.

Command Prompt

```
cvs -d :sserver:username@host:/cvsrepo co myproject/core
```

Now that the project is checked out I would recommend the first time you try it on a project to do a dry run. What Maven will do is simulate the

release but will not commit anything into CVS. Navigate to the pom and run the following command.

Command Prompt

```
mvn release:prepare -DdryRun=true -Dmaven.scm.provider.cvs.implementation=cvs_native
```

If it all seems to go as planned then clean the release by calling the following command.

Command Prompt

```
mvn release:clean
```

Now you can really prepare the release by removing the *-DdryRun=true*.

Command Prompt

```
mvn release:prepare -Dmaven.scm.provider.cvs.implementation=cvs_native
```

Once the release is successfully prepared you can finalize the release by using the following Maven command

Command Prompt

```
mvn release:perform -Dmaven.scm.provider.cvs.implementation=cvs_native
```

That is it, you have successfully created a release.