

Setting Up Maven Builds for Pipelines

Overview

If you are already using the [Maven Artifactory Plugin](#), your Maven projects will need some additional configuration for use within a Pipeline. Note that Pipelines does not require this plugin and configuration may be simpler without it, but if your project is already using the plugin, then you can configure your pipeline to do the same.

The plugin is used to automatically resolve dependencies, deploy artifacts, and publish builds when running a mvn build command. The procedures shown here enables the plugin to access Artifactory through the credentials stored in the [Artifactory Integration](#) that is used by your pipeline.

Page Contents

- [Overview](#)
- [Usage](#)

JFrog Pipelines Benefits

[DevOps Pipeline](#)

Usage

First, [configure the Project Object Model \(pom.xml\) configuration file of your Maven project](#) to use the [Maven Artifactory Plugin](#).

The Maven Artifactory Plugin will need to know your Artifactory credentials to be able to publish your Maven builds to repositories. You can provide this information in the Maven project's pom.xml configuration file, in the *publisher* block along with the names of the repositories where artifacts are to be published.

```
<publisher>
  <contextUrl> .. </contextUrl>
  <username> .. </username>
  <password> .. </password>
  <repoKey> .. </repoKey>
  <snapshotRepoKey> .. </snapshotRepoKey>
</publisher>
```

To keep this credential information secure, your pom.xml should reference these values as system properties that you have defined in your Maven project's settings.xml configuration file.

For even greater security, you should declare those *properties* in settings.xml using the environment variables provided by the [Artifactory Integration](#) to the step's execution environment. This also makes your Maven project adaptable to Pipelines – you can change which Artifactory instance is used in your pipeline without having to change your Maven project.

For example, this settings.xml file references the credentials and URL stored in the MyArtifactory integration:

settings.xml

```
<settings>
  <profiles>
    <profile>
      <id>artifactory-plugin-properties</id>
      <properties>
        <username>${env.int_MyArtifactory_user}</username>
        <password>${env.int_MyArtifactory_apikey}</password>
        <artifactoryUrl>${env.int_MyArtifactory_url}</artifactoryUrl>
      </properties>
    </profile>
  </profiles>

  <activeProfiles>
    <activeProfile>artifactory-plugin-properties</activeProfile>
  </activeProfiles>
</settings>
```

Note that the MyArtifactory integration must have been specified in the step's integrations tag (required for the MvnBuild native step) for these environment variables to be available to the step's execution environment.

Once defined in `settings.xml`, these properties can then be referenced in the `pom.xml` configuration file:

pom.xml

```
<publisher>
  <contextUrl>${artifactoryUrl}</contextUrl>
  <username>${username}</username>
  <password>${password}</password>
  <repoKey>libs-release-local</repoKey>
  <snapshotRepoKey>libs-snapshot-local</snapshotRepoKey>
</publisher>
```