

# Guide to uploading artifacts to the Central Repository

In order for users of Maven to utilize artifacts produced by your project, you must deploy them to a remote repository. Many open source projects want to allow users of their projects who build with Maven to have transparent access to their project's artifacts. In order to allow for this, a project should deploy their artifacts to the Central Repository (/repository/).

## Requirements

1. **releases**: Only *releases* can be uploaded to the Central Repository, that means files that won't change and that only depend on other files already released and available in the repository,
2. **javadoc and sources** for IDE lookup,
3. **PGP signature**,
4. **minimum POM information**: There are some requirements for the minimal information in the POMs that are in the Central Repository, see here (<https://central.sonatype.org/pages/requirements.html#sufficient-metadata>),
5. **coordinates**: Picking the appropriate coordinates for your project is important. See the guidelines here (<https://central.sonatype.org/pages/choosing-your-coordinates.html>), particularly on groupId and domain ownership (<https://central.sonatype.org/pages/producers.html#individual-projects-open-source-software-repository-hosting-ossrh>).

The updated list of requirements can be found **here** (<https://central.sonatype.org/pages/requirements.html>).

## Explanation

Some folks have asked "*why do we require all this information in the POM for deployed artifacts?*", so here's a small explanation.

The POM being deployed with the artifact is part of the process to make transitive dependencies a reality in Maven. The logic for getting transitive dependencies working is really not that hard, the problem is getting the data. The other applications that are made possible by having all the POMs available for artifacts are vast, so by placing them into the Central Repository as part of the process we open up the doors to new ideas that involve unified access to project POMs.

We ask for the license because it is possible that your project's license may change in the course of its lifetime, and we are trying to create tools to help sort out licensing issues. For example, knowing all the licenses for a particular graph of artifacts, we could have some strategies that would identify potential licensing problems.

## A basic sample:

```
1. <project>
2.   <modelVersion>4.0.0</modelVersion>
3.   <groupId>org.apache.maven</groupId>
4.   <artifactId>maven</artifactId>
5.   <version>2.0</version>
6.   <packaging>jar</packaging>
7.
8.   <name>Maven core</name>
9.   <description>The maven main core project description</description>
10.  <url>http://maven.apache.org</url>
11.
12.  <licenses>
13.    <license>
14.      <name>Apache License, Version 2.0</name>
15.      <url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
16.      <distribution>repo</distribution>
17.    </license>
18.  </licenses>
19.
20.  <scm>
21.    <url>https://svn.apache.org/viewvc/maven</url>
22.  </scm>
23.
24.  <dependencies>
25.    <dependency>
26.      <groupId>...</groupId>
27.      <artifactId>...</artifactId>
28.      <version>...</version>
29.    </dependency>
30.    ...
31.  </dependencies>
32.
33.  <!--
34.  NOT RECOMMENDED: (see FAQ)
35.  <repositories></repositories>
36.  <pluginRepositories></pluginRepositories>
37.  -->
38. </project>
```

## PGP Signature

When people download artifacts from the Central Repository, they might want to verify these artifacts' PGP signatures against a public key server. If there are no signatures, then users have no guarantee that they are downloading the original artifact.

To improve the quality of the Central Repository, we require you to provide PGP signatures for all your artifacts (all files except checksums), and distribute your public key to a key server like <http://pgp.mit.edu> (<http://pgp.mit.edu>). Read [Working with PGP Signatures](http://central.sonatype.org/pages/working-with-pgp-signatures.html) (<http://central.sonatype.org/pages/working-with-pgp-signatures.html>) for more information.

# FAQ and common mistakes

- I have other `repositories` or `pluginRepositories` listed in my POM, is that a problem?

At present, this won't preclude your project from being included, but we do strongly encourage making sure all your dependencies are included in the Central Repository. If you rely on sketchy repositories that have junk in them or disappear, it just creates havoc for downstream users. Try to keep your dependencies among reliable repos like Central, Jboss, etc.

- What about artifacts that can't be distributed because of their license?

In that case only the POM for that dependency is required, listing where the dependency can be downloaded from. See an example (<https://repo.maven.apache.org/maven2/javax/activation/activation/1.0.2/activation-1.0.2.pom>).

- I have a patched version of the foo project developed at foo.com, what `groupId` should I use?

When you patch / modify a third party project, that patched version becomes your project and therefore should be distributed under a `groupId` you control as any project you would have developed, never under `com.foo`. See above considerations about `groupId`.

- My project is hosted at a project hosting service like SourceForge or Github, what should I use as groupId?

If your project name is `foo` at SourceForge, you can use `net.sf.foo`. If your username is `bar` on Github, you can use `com.github.bar`. You can also use another reversed domain name you control. The group ID does not have to reflect the project host.

# Publishing your artifacts to the Central Repository

## Approved Repository Hosting

Instead of maintaining repository rsync feeds for each projects, we now encourage projects to use an approved repository hosting location.

Currently approved repository hosting locations:

- Apache Software Foundation (<https://repository.apache.org/>) (for all Apache projects)
- see the full list (<https://central.sonatype.org/publish/large-orgs/>)

Automatic publication will be provided for Forges that provide hosting services for OSS projects and other large project repositories that meet certain minimum criteria such as validation of PGP keys and pom contents as defined above. If you are interested in becoming an approved Forge, contact us (<https://central.sonatype.org/publish/large-orgs/>).

## Other Projects

The easiest way to upload another project is to use the Open Source Software Repository Hosting (OSSRH) (<http://central.sonatype.org/pages/ossrh-guide.html>), which is an approved repository provided by Sonatype for *any* OSS Project that want to get their artifacts into the Central Repository.

## Explanations

Having each project maintain its own repository with rsync to the Central Repository was the preferred process until January 2010. However, we are no longer accepting rsync requests on a per project basis.

Over time, we have learned that this process is not scalable. Many of the projects being synced release very infrequently, yet we have to hit hundreds of servers several times a day looking for artifacts that don't change. Additionally, there is no good mechanism currently for validating the incoming data via the rsync, and this leads to bad metadata that affects everyone.

The aggregation of projects into larger feeds allows us to sync faster and more often, and ensuring these locations perform sufficient checks increases the quality of metadata for everyone.