

Releasing OpenJPA 1.0.x or 1.1.x (Old SCP Steps)

true

Instructions for creating an official OpenJPA release

{scrollbar}

Making an OpenJPA 1.0.x or 1.1.x Release

These instructions guide the release manager through the steps of making an official OpenJPA release.

Prerequisites

1. You should read the [OpenJPA Release Policy](#) to decide on the name of the new release, based on the content.
2. You should read the Apache [Release FAQ](#)
3. You must have shell access to people.apache.org
4. You must have the following utilities installed on your local machine and available in your path:
 - [JDK 1.4 / 1.5](#)
 - [ssh](#)
 - [Maven 2](#)
 - [gnupg](#)
5. For Windows users, install [Cygwin](#) in addition to the above utilities
 - Make sure the Net/openssh and Utils/gnupg packages are installed that come with Cygwin installation.
 - Optional: [Putty](#)

Tasks that need to be performed for each release

In the examples below, it is assumed that the release name will be **1.0.1**, and that the current checked-in version name is **1.0.1-SNAPSHOT** and stored in the branch named **1.0.x**.

Monday, 12: 00	Make sure the #One time setup steps have been performed
Monday, 12: 01	Create a sub-branch off of the parent branch from which to make the release. Releasing from a branch will allow any cosmetic changes that need to be made for the release to be approved to be done without preventing other more disruptive advances in the trunk from potentially causing problems with the release. A branch can be made by running: <pre>none svn copy -m "OpenJPA Release 1.0.1 branch" \ https://svn.apache.org/repos/asf/openjpa/branches/1.0.x \ https://svn.apache.org/repos/asf/openjpa/branches/1.0.1</pre>
Monday, 12: 02	Check out a clean branch from which to build the release: <pre>none svn checkout https://svn.apache.org/repos/asf/openjpa/branches/1.0.1 cd 1.0.1</pre> Make sure there is no space characters in the path to the build root subdirectory, i.e. "c:\OpenJPA 1.0.1 Release\build". See "#Space Character in Build Root Path"
Monday, 12: 03	Update all the POMs to remove "-SNAPSHOT" suffix from the version. If you have perl installed, you can easily do it with a single command: <pre>none perl -pi -e "s:<version>1.0.1-SNAPSHOT</version>;<version>1.0.1</version>;g" \ pom.xml */pom.xml */*/pom.xml</pre> Update the <code><scm.dir></code> property in the top level pom.xml to the 1.0.1 release. \ Update the <code><developer></code> element in the top level pom.xml if new committers are added or removed. This applies to the 1.0.x branch only.
Monday, 12: 10	Verify that LICENSE.txt contains up to date licenses for any dependencies which are included in our distribution. Any jars or source code which is included with the OpenJPA distribution that is not covered by the Apache license must be noted in LICENSE.txt. Two examples of this are the persistence and orm dtos (licensed under the CDDL) and Serp. If any new non Apache dependencies have been introduced they will have to be covered here as well. If any discrepancies are found update LICENSE.txt and commit the changes.
Monday, 12: 15	Update BUILDING.txt, CHANGES.txt and RELEASE-NOTES.html <ul style="list-style-type: none">• BUILDING.txt should be included in the source tarball and contains instructions on how to build OpenJPA. Prior to shipping a release we should ensure that those instructions are accurate.• CHANGES.txt contains a text representation of all the changes which have been made since the preceding release. Most of the contents of this file can be generated through JIRA's release notes mechanism here• RELEASE-NOTES.html contains general information on the OpenJPA project as well as an html version of the changes since the preceding version. The html change log may also be generated via JIRA.
Monday, 12: 20	Commit the POM changes <pre>none svn commit -m "Updated to version 1.0.1 for the release"</pre>
Monday, 12: 21	Perform an initial build and install it in the local repository (this step is only required because of an open bug with Maven's javadoc plugin): <pre>none mvn clean install -Dtest=false</pre>

<p>Monday, 12: 23</p>	<p>Now build the release locally, which will build and test, run the Apache Release Audit Tool to verify license headers, generate the javadoc and docbook PDF and HTML, run through the JPA TCK, build the source and binary assemblies in target/site/downloads/, and sign the release files.</p> <pre>none export MAVEN_OPTS=-Xmx1000m mvn --batch-mode deploy site \ -Ptck-profile,examples-profile,license-verify-profile,javadoc-profile,docbook-profile,sign-release \ -Djava14.jar=\${JAVA_HOME}/../1.4/Classes/classes.jar \ -Dtck.zip=\${HOME}/.m2/privaterepos/jpa-1_0b-tck.zip</pre> <p>This operation will also sign the release files with the gpg utility using the <username>@apache.org key. If your code signing key is under a different address, specify it by appending the following argument to the command above: <code>-Dopenjpa.release.keyAlias=mysigningalias@somedomain.org</code></p> <p>The java14.jar path should be changed to the local install path for the JDK 1.4 rt.jar so that JDK 1.4 verification can take place.</p> <p>The tck.zip path should be changed to the local install path to the JPA TCK so that the TCK can be run against the release. If you are not licensed to access the TCK, remove the <code>tck-profile</code> profile and <code>-Dtck.zip</code> property from the command. Ask someone who has access to run the TCK for you.</p> <p>The <code>examples-profile</code> has problem running automatically in this maven build task. You will need to run the examples manually. Run the maven command without the <code>examples-profile</code>, then perform the following steps to run the example:</p> <pre>mkdir openjpa-integration\examples\target\examples unzip target\site\downloads\apache-openjpa-1.0.1-binary.zip -d openjpa-integration\examples\target\examples cd openjpa-integration\examples\target\examples\apache-openjpa-1.0.1\examples\hellojpa ant cd ../relations ant cd ../reversemapping ant</pre>
<p>Monday, 13: 00</p>	<p>Verify the signatures:</p> <pre>none gpg --multifile --verify target/site/downloads/*.asc</pre>
<p>Monday, 13: 01</p>	<p>Upload the release candidate to http://openjpa.apache.org/builds/1.0.1/downloads/:</p> <pre>none mvn site:deploy</pre> <p>Bear in mind that uploads to people.apache.org/www/openjpa.apache.org/builds/ are not visible at http://openjpa.apache.org/builds until after the hourly synchronization has taken place, as described at http://www.apache.org/dev/project-site.html.</p>
<p>Monday, 15: 00</p>	<p>Start a vote for the release on the dev@openjpa.apache.org mailing list. Votes made by committers and members of the OpenJPA project are considered binding for this vote. For an example of the mail, see this archived 1.0.0 vote</p>
<p>Tuesday, Wednesday</p>	<p>While waiting for the vote to complete, perform whatever manual review and testing on the release you deem appropriate.</p>
<p>Thursday, 15:00</p>	<p>If the vote is successful after 3 days tally the votes in an email to dev@openjpa.apache.org, like this one.</p>
<p>Thursday, 15:10</p>	<p>Make the OpenJPA assemblies available to the Apache mirroring system by copying them over on people.apache.org:</p> <pre>none ssh people.apache.org cp -r /www/openjpa.apache.org/builds/1.0.1/downloads/* /www/www.apache.org/dist/openjpa/1.0.1 chgrp -R openjpa /www/www.apache.org/dist/openjpa/1.0.1 chmod -R g+w /www/www.apache.org/dist/openjpa/1.0.1</pre> <p>The OpenJPA binary release will be available via the link http://www.apache.org/dyn/closer.cgi/openjpa/1.0.1/apache-openjpa-1.0.1-binary.zip after 24 hours, as per the Apache mirroring information.</p>
<p>Thursday, 15:15</p>	<p>Now that the release is locked down, convert the writeable 1.0.1 branch to a (du jure) read-only tag:</p> <pre>none svn mv -m "OpenJPA Release 1.0.1 tag" \ https://svn.apache.org/repos/asf/openjpa/branches/1.0.1 \ https://svn.apache.org/repos/asf/openjpa/tags/1.0.1</pre>
<p>Thursday, 15:16</p>	<p>Check out the parent branch and bump up the release number to be the next snapshot version:</p> <pre>none svn checkout https://svn.apache.org/repos/asf/openjpa/branches/1.0.x cd 1.0.x perl -pi -e "s:<version>1.0.1-SNAPSHOT</version>;<version>1.0.2-SNAPSHOT</version>;g" /pom.xml */pom.xml */*/pom.xml svn commit -m "Updating version in branch to 1.0.2-SNAPSHOT"</pre>
<p>Thursday, 15:20</p>	<p>Update the http://cwiki.apache.org/openjpa/downloads.html page with links to the download mirrors, using the existing entries as templates.</p> <p>All artifacts (apache-openjpa-1.0.1-binary.zip and apache-openjpa-1.0.1-source.zip) must link to the mirrors, but signatures (apache-openjpa-1.0.1-binary.zip.asc and apache-openjpa-1.0.1-source.zip.asc) must not link to mirrors.</p>
<p>Thursday, 15:30</p>	<p>The documentation on the server-side must manually be extracted on people.apache.org, and the links at http://openjpa.apache.org/documentation.html need to be updated with the new versions and the "latest" documentation symbolic links need to be updated:</p> <pre>none cd /www/openjpa.apache.org/builds/1.0.1/ unzip downloads/apache-openjpa-1.0.1-binary.zip rm /www/openjpa.apache.org/docs/latest ln -fvs ../builds/1.0.1/apache-openjpa-1.0.1/docs/ /www/openjpa.apache.org/docs/latest</pre> <p>Then point the links on http://openjpa.apache.org/documentation.html to:</p> <ul style="list-style-type: none"> • HTML Manual: http://openjpa.apache.org/builds/1.0.1/apache-openjpa-1.0.1/docs/manual/index.html • HTML Manual (single-page): http://openjpa.apache.org/builds/1.0.1/apache-openjpa-1.0.1/docs/manual/manual.html • PDF Manual: http://openjpa.apache.org/builds/1.0.1/apache-openjpa-1.0.1/docs/manual/manual.pdf • Javadocs: http://openjpa.apache.org/builds/1.0.1/apache-openjpa-1.0.1/docs/javadoc/index.html
<p>Thursday, 15:45</p>	<p>Merge the staged maven2 repository jars to with the official repository to rsync. This is currently a difficult process that requires the maven-staging-plugin, which much be built from source. The process looks something like this (executed in the root of your local OpenJPA checkout):</p> <pre>none svn co http://svn.apache.org/repos/asf/maven/plugins/trunk/maven-stage-plugin/ \ /tmp/maven-stage-plugin mvn -f /tmp/maven-stage-plugin/pom.xml clean install for i in \$(find openjpa-* -name m2-repository egrep -v "openjpa-project openjpa-integration openjpa-examples"); do cd \$i mvn stage:copy -Dsource=file://. \ -Dtarget=scp://people.apache.org/www/people.apache.org/repo/m2-ibiblio-rsync-repository \ -Dversion=1.0.1 cd - done</pre> <p>This process requires Maven 2.0.5. It is currently quite delicate and error-prone. Once the maven-stage-plugin is released, it should be possible to make it more automated. Window/Cygwin user: See #Merging local repository to remote repository in Cygwin/Windows</p> <p>Linux users: You may need to change the <code>-Dsource=</code> option to read <code>-Dsource=file:[]</code>. The <code>file://[]</code> syntax may result in errors creating the wagon file with errors like: <code>"Repository path /openjpa-jdbc-5/target/site/m2-repository does not exist, and cannot be created."</code></p>
<p>Thursday, 16:00</p>	<p>Update the JIRA versions page to mark the version as "released", and set the date to the date that the release was approved. You may also need to make a new release entry for the subsequent release.</p>

Friday, 16: 00	After the mirrors have had time to update (24 hours to be on the safe side), make a news announcement on the OpenJPA wiki. Once the news item is made, it won't show up on the front page unless you make some minor edit to the containing page (e.g., adding a newline somewhere).
Friday, 16: 05	Make an announcement about the release on the users@openjpa.apache.org list (and, for major releases, on the announce@apache.org list as per the Apache Announcement Mailing Lists page). The announcement might look something like this .
Friday, 16: 10	Make an announcement for the freshmeat.net OpenJPA project (optional)
Friday, 17: 00	Have a beer and enjoy your weekend while the world's grateful programmers revel in yet another high-quality release of Apache OpenJPA!

One time setup

These setup steps only need to be performed on a particular machine once.

Developers using Linux workstations can skip over the references to PuTTY and Cygwin

Create and install a SSH key

1	Install PuTTY
2a	Use ssh-keygen to create a SSH key. See Authenticating By Public Key (OpenSSH) for a good description on why and how to perform this task.
2b	In Windows platform, use PuttyGen to create a SSH key (see Putty help for details). <ul style="list-style-type: none"> Use "SSH-2 DSA" key type and 1024-bit key size. Copy the content of the "Public key for pasting...." and save it to a file named <code>authorized_keys</code> for later use. The private key saved by PuTTYGen can only be used in Putty configuration.
3	<code>pscp</code> your SSH public key <code>authorized_keys</code> saved in last step to <code>~/authorized_keys</code>
4	Use PuTTY to login to <code>people.apache.org</code>
5	Create a <code>~\.ssh</code> folder and change its file mode to 700.
6	Move or append <code>~/authorized_keys</code> to <code>~/.ssh/authorized_keys</code> and change its file mode to 600. <ul style="list-style-type: none"> Each public key in the <code>authorized_keys</code> spans only one line. <ul style="list-style-type: none"> For example: <code>"ssh-dss AAAAB3NzaC1kc3MAAA agBmmfZ9uAbSqA= dsa-key-20071107"</code> '#' in the first column is a comment line.
7	Configure putty to use your private key and save the session Specify your private key in the "Connection -> SSH -> Auth" category in Putty configuration.

Create a PGP key

1	Install cygwin, including Utils/gnupg and Net/openssh packages, or install gpg from http://www.gnupg.org/(en)/download/index.html
2	Generate a key-pair with <code>\$ gpg --gen-key</code> using default key kind ("DSA and Elgamal") and ELG-E keys size (2048). <ul style="list-style-type: none"> The generated keys are stored in <code>\$HOME/.gnupg</code> or <code>%HOME%\Application Data\gnupg</code> subdirectory. Save the content in this subdirectory to a safe media. This contains your private key used to sign all the OpenJPA release materials.
3	Backup your cygwin home directory to another media
4	Add your public key to https://svn.apache.org/repos/asf/openjpa/site/docs/KEYS and http://www.apache.org/dist/openjpa/KEYS . See the commands describe at the beginning of this KEYS file to perform this task. The gpg key-pair is used to sign the published artifacts for the OpenJPA releases.
5	Following the instructions in http://people.apache.org/~henkp/trust/ and ask someone in the OpenJPA project to sign your public key.
6	Submit your public key to a key server. E.g. http://pgp.surfnet.nl:11371/ or http://pgp.mit.edu/

Update Maven settings for our servers

1	Create a settings.xml under .m2 (in your Document and Settings folder in Windows)
	<pre> xmlsettings.xml<solid><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"><servers><server><id>people.apache.org</id><username>\${USERNAME}</username><privateKey>\${PATH_TO_PRIVATE_KEY}</privateKey><directoryPermissions>775</directoryPermissions><filePermissions>644</filePermissions></server></servers></settings> \$PATH_TO_PRIVATE_KEY is the path to the private key generated for ssh. E.g. /home/yourLocalUserId/.ssh/id_dsa. </pre>

Expose a copy of known hosts to Maven

1	From cygwin, ssh to people.apache.org, save the public key if prompted, and exit
	cygwin will save the known hosts to your ~/.ssh folder, but the script cannot access it there (from Windows)
2	From cygwin (not Windows) create another .ssh folder at ?
3	Copy the known_hosts file to the new .ssh folder

Troubleshooting

Space Character in Build Root Path

Description	If there are spaces in the path to the build root subdirectory, the maven task uses to generate the revision number for the org.apache.openjpa.revision.properties yields incorrect data. For example: org.apache.openjpa.revision.properties<solid>revision.number=Type 'svnversion --help' for usage. openjpa.version=1.0.1
Solution	Rename the path and remove all spaces.

Merging local repository to remote repository in Cygwin/Windows

Description	<p>The "maven-stage-plugin" is very sensitive to the parameters being passed to it, i.e. the source and target URL properties. When this plugin is used under Cygwin, make sure the following practices are used:</p> <ul style="list-style-type: none"> Use absolute path in the find command's root directory. Problem symptom: \$ find -name m2-repository -not -path ""openjpa-project"" -exec mvn -f "c:/tmp/maven-stage-plugin/pom.xml" stage:copy -Dsource=file://{ } -Dtarget=scp://allee8285@people.apache.org/www/people.apache.org/repo/m2-ibiblio-rsync-repository \ -Dversion=1.0.1 \; [INFO] Scanning for projects... [INFO] Searching repository for plugin with prefix: 'stage'. [INFO] Downloading file from the source repository; [INFO] [ERROR] BUILD ERROR [INFO] [INFO] Error copying repository from file://cygdrive/c/OpenJPA.1.0.1.Release/1.0.1/openjpa-all/target/site/m2-repository to \ scp://allee8285@people.apache.org/www/people.apache.org/repo/m2-ibiblio-rsync-repository Quote and use the drive name in the path. Problem symptom: \$ find /cygdrive/c/OpenJPA.1.0.1.Release/1.0.1 -name m2-repository -not -path ""openjpa-project"" -exec mvn -f /cygwin/c/tmp/maven-stage-plugin/pom.xml stage:copy -Dsource=file://{ } -Dtarget=scp://allee8285@people.apache.org/www/people.apache.org/repo/m2-ibiblio-rsync-repository \ -Dversion=1.0.1 \; [INFO] Scanning for projects... [INFO] Searching repository for plugin with prefix: 'stage'. [INFO] Downloading file from the source repository; [INFO] [ERROR] BUILD ERROR [INFO] [INFO] Error copying repository from file://cygdrive/c/OpenJPA.1.0.1.Release/1.0.1/openjpa-all/target/site/m2-repository to scp://allee8285@people.apache.org/www/people.apache.org/repo/m2-ibiblio-rsync-repository Embedded error: Could not read from file: c:/cygdrive/c/OpenJPA.1.0.1.Release/1.0.1/openjpa-all/target/site/m2-repository \cygdrive/c/OpenJPA.1.0.1.Release/1.0.1/openjpa-all/target/site/m2-repository (Access is denied.) Specify the people.apache.org user id in the target property. Problem symptom: \$ find "c:/OpenJPA.1.0.1.Release/1.0.1" -name m2-repository -not -path ""openjpa-project"" -exec mvn -f "c:/tmp/maven-stage-plugin/pom.xml" stage:copy -Dsource=file://{ } -Dtarget=scp://people.apache.org/www/people.apache.org/repo/m2-ibiblio-rsync-repository \ -Dversion=1.0.1 \; [INFO] Scanning for projects... [INFO] Searching repository for plugin with prefix: 'stage'. [INFO] [INFO] Downloading file from the source repository: /org/apache/openjpa/openjpa/maven-metadata.xml.sha1 [INFO] Downloading metadata from the target repository. Password: ***** Password: ***** [INFO] [ERROR] BUILD ERROR [INFO] [INFO] Error copying repository from file://c:/OpenJPA.1.0.1.Release/1.0.1/openjpa-all/target/site/m2-repository to \ scp://people.apache.org/www/people.apache.org/repo/m2-ibiblio-rsync-repository
Solution	<p>As recommended in the descriptions.</p> <p>For example:</p> <pre> find "c:/OpenJPA.1.0.1.Release/1.0.1" -name m2-repository -not -path ""openjpa-project"" -exec mvn -f "c:/tmp/maven-stage-plugin/pom.xml" stage:copy \ -Dsource=file://{ } -Dtarget=scp://allee8285@people.apache.org/www/people.apache.org/repo/m2-ibiblio-rsync-repository \ -Dversion=1.0.1 \; </pre>

Cygwin/Windows File Path

Description	For Cygwin/Windows user: file and folder path names using drive identifier (e.g. C:\OpenJPA Release\1.0.1) in commands can be expressed as /cygwin/c/OpenJPA Release/1.0.1/. This form of path name specification may have inconsistent and undesirable behaviors.
Solution	<p>Consistently use the following naming conventions:</p> <ul style="list-style-type: none"> Continue to use the Windows form of path name, e.g. C:\a\b\c Use '/' instead of '\' character as file separator, e.g. C:/a/b/c Quote all path name using '"' character, e.g. "C:/a/b/c" Avoid using space characters in path name, e.g. "C:/OpenJPA.Release/1.0.1"

"Too many unapproved licenses:"

Description	Encounter the "Too many unapproved licenses:" message while running the "license-verify-profile" profile in "mvn deploy site..." step. This is caused by extra artifacts in the build tree that the license verification plugin does not recognized. Examples of these artifacts are: <ol style="list-style-type: none">1. Eclipse control files, .classpath, .project2. User created log files
Solution	Avoid the followings: <ol style="list-style-type: none">1. Don't use Eclipse's svn plugin to "Check out" files to a Eclipse project. Simply use the svn command, as described in the release instructions.2. Don't create, directly or indirectly, any files under the release build tree.

Resources

- Apache Apache [Release FAQ](#)
- [Signing Releases](#)
- [Wendy's notes on Release Signing](#)
- Apache [Mirroring Guidelines](#)
- Struts [Release Instructions](#) (upon which these instructions are based)