ReleaseProcedure

Introduction

This document describes how to release a new version of Apache log4php.

This procedure has been written and tested in a Linux environment. While it is possible to perform the release in Windows or OSX, this guide does not cover any platform-specific quirks which may be encountered.

Preparation and release guidelines are based on the following documents:

- ASF Release FAQ
- ASF Signing Releases FAQ
- Appache commons release guide

The examples assume version 2.1.0 is being prepared for a release.

Contents

Preparation

Select a release manager

The release manager can be nominated by posting a message to the developers list. Typicaly the poster will volunteer himself as the release manager and it passes by lazy consensus.

Signature keys

The released artifacts must be using the release manager's PGP key. If the release manager does not have one, he must:

- Generate a key
- Upload the key to a public key server such as the MIT PGP Public Key Server

Make sure to remeber the keyphrase used when generating the key. It will later be required to sign the release artifacts.

When the key has been generated, it should be added to the project's KEYS file.

Check the file out:

```
svn co https://svn.apache.org/repos/asf/logging/log4php/meta
```

Add your keys to the end of it. For example, using GnuPG:

```
gpg --list-sigs <your name> >> KEYS
gpg --armor --export <your name> >> KEYS
```

Check in the change:

```
svn commit -m "Added my signing key to KEYS"
```

Update the KEYS file on people.apache.org:

```
scp KEYS people.apache.org:/www/www.apache.org/dist/logging/log4php
```

It is also desirable to:

- have the key signed on a key signing party
- add the key to your FOAF file

POM file revision

Before staging a release candidate take some time to check if any updates are required to pom.xml.

It is recommended to update the used plugins to the latest stable version.

Plugins are defined in <plugin> nodes similar like:

```
<plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-antrun-plugin</artifactId>
        <version>1.7</version>
</plugin>
```

Visit the plugin site (google it) and update the version to the latest stable version of the plugin. Check for any breaking changes in the change log and update the configuration if required. After updating each plugin, please test that it works as expected.

Creating a Release Candidate

Before starting this process shout a Code Freeze message at the developers list.

In this chapter, <username> and <password> should be replaced with the release manager's SVN credenials.

Make a clean checkout of the trunk sourcecode:

```
svn checkout https://svn.apache.org/repos/asf/logging/log4php/trunk log4php-trunk
```

Test the preparation procedure:

```
mvn release:prepare -Dusername=<username> -DdryRun
```

Maven will ask several questions.

```
What is the release version for "Apache log4php"? (org.apache.log4php:apache-log4php) 2.1.0: :
What is SCM release tag or label for "Apache log4php"? (org.apache.log4php:apache-log4php) apache-log4php-
2.1.0: :
What is the new development version for "Apache log4php"? (org.apache.log4php:apache-log4php) 2.1.1-SNAPSHOT: :
```

Provide the release version "2.1.0" and leave the release tag and new development version default values by pressing ENTER.

If something went wrong, you can clean up with:

```
mvn release:clean
```

Prepare the release, this will create a tag in SVN:

```
mvn release:prepare -Dusername=<username> -Dpassword=<password>
```

If the build fails with the message "revision xyz not found", just re-run the prepare goal. This is a known bug.

Finally, end the code freeze with a short message to the developers list.

Create binary packages

Make a clean checkout of the tag sourcecode

```
svn checkout https://svn.apache.org/repos/asf/logging/log4php/tags/apache-log4php-2.1.0 log4php-tag
```

Create the site and binary packages:

```
mvn site package
```

This creates the following files in the "target" directory:

- Apache_log4php-2.1.0.tgz
- apache-log4php-2.1.0-src.tar.gz
- apache-log4php-2.1.0-src.zip

Create checksums

On Linux and Windows you can use md5sum:

```
md5sum -b Apache_log4php-2.1.0.tgz > Apache_log4php-2.1.0.tgz.md5
md5sum -b apache-log4php-2.1.0-src.tar.gz > apache-log4php-2.1.0-src.tar.gz.md5
md5sum -b apache-log4php-2.1.0-src.zip > apache-log4php-2.1.0-src.zip.md5
```

On a Mac, you can use md5:

```
md5 -r Apache_log4php-2.1.0.tgz > Apache_log4php-2.1.0.tgz.md5
md5 -r apache-log4php-2.1.0-src.tar.gz > apache-log4php-2.1.0-src.tar.gz.md5
md5 -r apache-log4php-2.1.0-src.zip > apache-log4php-2.1.0-src.zip.md5
```

Note: md5sum for windows can be found here.

Sign your binaries

Sign your binaries. You can do this with gpg, which is available for linux, windows and mac.

The key for signing needs to be placed in: https://svn.apache.org/repos/asf/logging/log4php/meta/KEYS

```
gpg --armor --output apache-log4php-2.1.0-src.tar.gz.asc --detach-sig apache-log4php-2.1.0-src.tar.gz
gpg --armor --output apache-log4php-2.1.0-src.zip.asc --detach-sig apache-log4php-2.1.0-src.zip
gpg --armor --output Apache_log4php-2.1.0.tgz.asc --detach-sig Apache_log4php-2.1.0.tgz
```

Note: On windows, use Gpg4win

Upload to staging server

First, tar the release packages and the site:

```
cd target
tar cvzf release.tar.gz *log4php*
tar czvf site.tar.gz site
```

Copy the release to the log4php folder on the people server, and site to your home folder:

```
scp release.tar.gz <username>@people.apache.org:/x1/builds/logging/log4php
scp site.tar.gz <username>@people.apache.org:/home/<username>
```

Login to the people server. Create 2.1.0/RC1 directory and unpack the release there:

```
cd /x1/builds/logging/log4php
mkdir 2.1.0
mkdir 2.1.0/RC1
tar xzvf release.tar.gz -C 2.1.0/RC1/
rm release.tar.gz
```

Unpack the site in your home dir (if public_html doesn't exist, create it):

```
cd ~
mkdir public_html/apache-log4php-2.1.0-RC1
tar xzvf site.tar.gz -C public_html/apache-log4php-2.1.0-RC1
rm site.tar.gz
```

The files are now accessible from:

- http://people.apache.org/builds/logging/log4php/2.1.0/RC1/

And the RC site is located at:

- http://people.apache.org/~<username>/apache-log4php-2.1.0-RC1/site

Voting

Once everything has been uploaded to the staging server, send a message to the [developers list](#) to initiate voting.

Example of an email announcemenet:

```
Subject: [VOTE] Release Apache log4php 2.1.0

Dear all,

It is my pleasure to announce the first release candidate for Apache log4php version 2.1.0.

Significant changes in this release include:
  * this
  * that
  * other

Apache log4php 2.1.0 RC1 is available for review here:
  * http://people.apache.org/builds/logging/log4php/2.1.0/RC1/

The KEYS file for verifying signatures is located at:
  * http://www.apache.org/dist/logging/log4php/KEYS

The tag for this release is available here:
  * http://svn.apache.org/viewvc/logging/log4php/tags/apache-log4php-2.1.0/

The site for this version can be found at:
  * http://people.apache.org/~<username>/apache-log4php-2.1.0-RC1/site/

According to the process, please vote with:
[ ] +1 Yes go ahead and release the artifacts
[ ] -1 No, because...

Best regards,
Your RM
```

Votes from PMC members are binding, however votes from other committers, users and contributors are welcome. Release VOTEs should be left open for a minimum of 72 hours so community members have ample opportunity to download, review and test the release candidate.

Three binding votes are needed for voting to be successful.

On voting failure

If the vote fails, fix the problems which were identified and create a new release candidate (if this run was RC1, the next is RC2).

Before that some clean up must be done.

Rename the SVN tag by adding "-RC1" to make room for the next release candidate:

```
svn move https://svn.apache.org/repos/asf/logging/log4php/tags/apache-log4php-2.1.0 https://svn.apache.org/repos
/asf/logging/log4php/tags/apache-log4php-2.1.0-RC1
```

The prepare process will have inceremented the version number in pom.xml. In this case from 2.1.0 to 2.1.1. Revert this change. E.g.:

```
-   <version>2.1.1-SNAPSHOT</version>
+   <version>2.1.0-SNAPSHOT</version>
```

After this has been done, the preparation process can be repeated using the next RC number.

On voting success

Once a vote is successful, post an email to developers list✉ as a reply to the original [VOTE] email, with the subject "[RESULT] Release Apache log4php 2.1". This message should clearly state the binding votes.

Releasing

After after successful vote, the release may be performed.

Move binaries to distribution directory

Move the source packages, their detached signatures and md5 sums from the staging area to the distribution directory.

On people.apache.org:

```
cd /www/www.apache.org/dist/logging/log4php
mkdir 2.1.0
mv /x1/builds/logging/log4php/2.1.0/RC1/* 2.1.0
```

Double check the permissions of all moved files. The file permissions should be 664 (rw-rw-r-) and the group should be "logging", for example:

```
-rw-rw-r--  1 ihabunek  logging    75995 Jun 30 06:39 Apache_log4php-2.1.0.tgz
-rw-rw-r--  1 ihabunek  logging      836 Jun 30 06:39 Apache_log4php-2.1.0.tgz.asc
-rw-rw-r--  1 ihabunek  logging       64 Jun 30 06:39 Apache_log4php-2.1.0.tgz.md5
-rw-rw-r--  1 ihabunek  logging   328635 Jun 30 06:39 apache-log4php-2.1.0-src.tar.gz
-rw-rw-r--  1 ihabunek  logging      836 Jun 30 06:39 apache-log4php-2.1.0-src.tar.gz.asc
-rw-rw-r--  1 ihabunek  logging       66 Jun 30 06:39 apache-log4php-2.1.0-src.tar.gz.md5
-rw-rw-r--  1 ihabunek  logging   714160 Jun 30 06:39 apache-log4php-2.1.0-src.zip
-rw-rw-r--  1 ihabunek  logging      836 Jun 30 06:39 apache-log4php-2.1.0-src.zip.asc
-rw-rw-r--  1 ihabunek  logging       63 Jun 30 06:39 apache-log4php-2.1.0-src.zip.md5
```

Remove older versions

The distribution directory should contain only the latest release of log4php. If any older versions are present they should be removed.

On people.apache.org:

```
cd /www/www.apache.org/dist/logging/log4php
rm -rf 2.0.0
```

Update the site in SVN

At the time of writing this guide, the site-deploy goal in maven was not configured to work properly.

The **post-site** goal will generate the new site to the *site* directory and check out the current production site to the *site-deploy* directory.

```
mvn post-site
```

Since almost all files will change when generating (most include a timestamp if nothing else), the easiest way to update the site is to delete the old one completely and replace it with the new version.

```
cd site-deploy
svn delete *
svn commit -m "Deleting current log4php site to make room for new version."

cp -R ../site/* .
svn add *
svn commit -m "Uploading new version of log4php site."
```

This process should be improved in the future.

Deploy the new site

To deploy the site, go to people.apache.org and update it:

```
cd /www/logging.apache.org/log4php/
svn update
```

Send an announcement

The announcement email should be sent to the following lists:

- log4php-dev@logging.apache.org
- log4php-user@logging.apache.org
- announce@apache.org

The subject of the announcement should be similar to:

```
[ANN] Apache log4php 2.1.0 released
```

For some examples, see the archives: http://markmail.org/search/?q=log4php+order%3Adate-backward+list%3Aorg.apache.announce

Party on! 🙂