AvalonReleaseManagerHowto

How to be an Avalon Release Manager

This howto takes you through the steps of successfully handling the avalon release process. Its currently a kind of bulletted no-nonsense thing, but anyone should be able to follow.

Who can be a Release Manager?

Anyone. You don't even need to be a committer. But you do need someone with an account to the relevant machine (currently daedalus.apache.org / www.apache.org, will become minotaur.apache.org / www.apache.org / cvs.apache.org) to do the actual upload of the releases.

The process, step by step

1. Get up to speed on the basics

you need to know about our technical infrastructure and the tools available to you:

http://www.apache.org/dev/

Our releases go to http://www.apache.org/dist/avalon/, which is organised in accordance to the mandatory Mirroring Guidelines and in addition is usuable as a maven repository. You get them there by copying them securely using an SSH client (the scp tool in OpenSSH, or SecureFX, or pscp from the putty suite, or any number of other clients).

2. Get a general sense of subproject direction

Is anyone planning a big refactoring of the stuff you want to release? What is everyone's opinion on the quality of that material? What do people want to do with it? Find the answers to these questions one way or another. Usually a mailing list thread will do.

3. Get people to accept you as the release manager

This is usually not a lot of hassle and subject to lazy consensus, but make your intentions known. Can be tied in with item 4 below.

4. Draft a release plan

Decide on how you want to handle the release process. And write that up. You might want to create a stable branch in cvs, use a tracker such as jira to assign bugs to a particular release, etc etc. Or you might want to not do any of that and simply keep a long list of mailing list thread references on a wiki page. But figure that out.

Also, get a general idea of how many release candidates will be neccessary, how you will call them and what 'milestones' you will associate with those release candidates. Write all that up, but don't worry about being overly precise.

Release plan example

```
 subject: Avalon-Dagger Release Plan
to: dev@avalon.apache.org
from: johndoe@hotmail.com

Hi gang,

I would like to get a 1.2 release together for
Avalon-Dagger in the next month or so. I'm volunteering
to be the release manager and coordinate that release.

I will make a release candidate available roughly every
week at

http://www.johndoe.com/avalon/release-candidates/dagger/

until no big blocker issues remain. I will also
periodically post an overview of the remaining issues to
the mailing list. I expect to have a
final release in three weeks from now. Issues will be
tracked using the wiki; everyone is requested to send
details of any issues to the development mailing list
*and* add a link to the relevant thread on the wiki page
at

http://wiki.apache.org/avalon/AvalonDaggerReleaseManagement

Overview of open issues
=======================
ADAG-123 (Refactor Dagger3 class to be ThreadSafe)
ADAG-157 (TimedDagger has y2k vulnerability)
ADAG-149 (Remove JDK-1.4 dependencies)
ADAG-142 (Proper javadocs on TimedDagger)
```

5. Build Release Candidates

We like to name our release candidates ${blah}~~${version}-RC#, where ${blah}~~${version} will be the final name of the distribution and -RC# is the release candidate identifier (so, for example avalon-dagger-1.2-RC1, avalon-dagger-1.2-RC2, avalon-dagger-1.2-RC3, etc). However, don't name the actual distributions like that; give those the final name (so, avalon-dagger-1.2). Just put them in a place where it is clear they are release candidates and not final (ie, http://cvs.apache.org/~johndoe/release-candidates and *not* http://www.apache.org/dist/avalon/).

To build a release candidate, simply follow the normal build procedure (a cvs update, a build cleanup, followed by a distribution build). That may look like:

```
 cd ~/cvs/avalon/dagger/
cvs -z3 up -P -d
maven clean
maven dist
```

6. Sign the files

You need to use GnuPG or a similar tool to sign the releases. Since many people are not used to PGP signing tools, a lot more extensive info is at /SigningReleases.

7. Tag cvs

To keep track of how you built what release and to be able to reconstruct a particular release candidate later, you should tag the cvs. This is really simple. Here's a sample command:

```
 cd ~/cvs/avalon/dagger/
cvs tag -R Dagger_1_2_a1
```

you can learn more about the 'tag' command by issueing the

```
  cvs -H tag
```

command. In order to make sense out of CVS tags, we follow a consistent tag format that replaces all '.' and '-' with a ' ' *(cvs only allows 0-9A-Za-z for tag names, no '.')*, starts with a capital letter, is otherwise lowercase, and replaces any 'RC#' with either 'a#' or 'b#'. So

```
  Cornerstone_1_0_a1
Avalon_4_1_4
Phoenix_4_0_1_b2
```

are valid examples.

8. make the candidate files available

Upload the files to your chosen public location. For example:

```
cd ~/cvs/avalon/dagger/target/distributions
ssh cvs.apache.org -c
  'mkdir -p ~/public_html/avalon/release-candidates/dagger/binaries'
ssh cvs.apache.org -c
  'mkdir -p ~/public_html/avalon/release-candidates/dagger/source'
scp *bin*.zip *bin*.gz *bin*.asc
   cvs.apache.org:~/public_html/avalon/release-candidates/binaries/
scp *src*.zip *src*.gz *src*.asc
   cvs.apache.org:~/public_html/avalon/release-candidates/source/
```

}}}
which are bash (unix/linux/cygwin) commands using OpenSSH. But you can use graphical tools like SecureFX as well if you like.

9. Announce the release candidate

Ideally, you can get a team of beta testers to try out every release candidate to make sure no bugs creep in. Get their attention by posting an announcement to the user and the development mailing list.

Example Announcement

```
 subject:[[ann]][Dagger] Release Candidate 1 available
to: dev@avalon.apache.org, users@avalon.apache.org
from: johndoe@hotmail.com

Hi gang,

A new avalon-dagger-1.2 release candidate (RC1) is up
for review and comment at

http://cvs.apache.org/~hsimpson/release-candidates/avalon/dagger/

I'd like all users of these components to test it. I
would like you all to add all open issues, requests, bug
reports, etc, relating to these components to this wiki page:

http://nagoya.apache.org/wiki/apachewiki.cgi?AvalonDaggerReleaseManagement

in addition to sending them to the dev@avalon.apache.org
mailing list.

I would like to see everyone interested in helping out
looking at that page, adding to it, and either committing
fixes, or submitting patches, depending on cvs karma, of
course. Thanks for your help in ensuring a quality release!

A new release candidate will be made available roughly
weekly until all major issues have been resolved.

best regards,

- John Doe
```

10. Rinse and repeat

Keep building release candidates until you think your stuff is good to go.

11. Call for a PMC vote

(Mostly) for legal reasons, the Avalon PMC must vote on all releases. Anyone can call for a vote though. If the release process went well, this should be mostly a formality. Note that you don't have to be on the PMC (or even a committer) to initiate this vote. Note you will always need to get a PMC member to administer the vote

Example call to vote

```
subject: [PMC:VOTE] Avalon-Dagger Release
to: dev@avalon.apache.org, pmc@avalon.apache.org
from: johndoe@hotmail.com

Hi gang,

I'm calling for a vote to make the avalon-dagger
release candidate that has been up for review and
commment at

 http://cvs.apache.org/~johndoe/release-candidates/avalon/dagger/

into a final release. PMC members, please place
your votes:

 [ ] +1 (for)
 [ ] ±0 (abstain)
 [ ] -1 (against)

 Project Background
 ------------------
 (explain the status of the package, how it has
 changed or not changed recently, a brief overview
 of what it does, etc etc).
```

12. Administer the vote

Votes are open for a week, normally (or until majority is attained). Count and publish the results.

Example vote result

```
subject: [PMC:VOTE-RESULT] Avalon-Dagger Release
to: dev@avalon.apache.org, pmc@avalon.apache.org
from: johndoe@hotmail.com

Hi gang,

I called for a vote to make the avalon-dagger
release candidate that has been up for review and
commment at

  http://cvs.apache.org/~johndoe/release-candidates/avalon/dagger/

into a final release. The results of the vote are
as follows:

[" 10 "] +1 (for)
[" 3 "] ±0 (abstain)
[" 2 "] -1 (against)

Conclusion
----------
Accordingly, the releases will be placed in the
appropriate location and an announcement will be
sent.
```

13. Distribute the files

copy the release candidate to the appropriate location on the distribution server, normally in a subdirectory of

www.apache.org:/www/www.apache.org/dist/avalon/

for example:

```
scp -r cvs.apache.org:~/public_html/avalon/release-candidates/dagger .
scp -r dagger www.apache.org:/www/www.apache.org/dist/avalon/
```

14. Send the release announcment

Wait for 24 hours to allow all the distribution mirrors to pick up the changes, then send a release announcement. For major releases, send them to users@avalon.apache.org, dev@avalon.apache.org and announce@apache.org. For smaller releases, omit the announce@apache.org address. You may also want to send the release announcements to news sites such as theserverside.com, slashdot.org, etc etc.

Also make sure to post the announcement on the website.

Example Release Announcement

```
subject: Avalon-Dagger 1.2 released
to: dev@avalon.apache.org, users@avalon.apache.org,
       announce@apache.org
from: dev@avalon.apache.org

The Avalon team is proud to announce the 1.2 final release of
Avalon-Dagger.

Avalon-Dagger is a state-of-the-art IoC container supporting
advanced features such as Aspect-Oriented Programming,
Interceptors and an asynchronous event architecture.
The new 1.2 release provides bugfixes, minor enhancements and
major new features like hot swapping of components,
full support for ["AltRMI"] remoting, and more.

You may get the official release from the following URL:

  http://avalon.apache.org/download.cgi

For more information on Avalon-Dagger, please see

 http://avalon.apache.org/dagger/

For more information on the Apache Avalon project, please see

 http://avalon.apache.org/

best regards,

 - The Avalon Development Team
```

In summary

Basically:

- know your tools (use ant, maven, ssh, scp, gpg and some shell scripting to automate everything, as described above)
- don't make a hassle out of the process, just do things and volunteer
- write a plan
- build frequent release candidates
- drink coffee, be patient

checklist:

1. refresh tech skills
2. discuss goals
3. write release plan, volunteer
4. build, and test release candidate
5. tag cvs
6. sign and deploy release candidate
7. announce release candidate
8. repeat 4-7 as required, automating as much as possible
9. call PMC vote to give a candidate the 'final' status

10. deploy release
11. send announcements (after 24 hours)
12. take a deserved break!