

# DocumentationPlan

## Documentation Plan

It takes time to produce adequate documentation. Often it's hard to merge changes so a little organisation is a good idea. If you are working on substantial changes to a particular document then sign up on here. Then people can either work on something else or post something to the general list.

Not sure whether management via wiki will be better than via JIRA (create loads of tasks?)

## Page By Page

Feel free to update with new ideas etc

- [Guide For Proposals](#) is reasonably close. there's editorial work needed proof-reading, adding links and add examples of good practice from existing applications. there are a few difficult sections needing more content.
- [Guide For Participation](#) is a new-ish netiquette/getting involve guide. intended to be a short document
  - more content and links into existing documentation ✓
  - add links to mailing list netiquette (when it's done)
- [Release Management Guide](#) is really just an outline. Shape is fine (including TODOs) but content required.
  - Create project templates
  - Review <http://wiki.apache.org/general/ReleaseGuides> and add useful content.
  - Review release management content in policy: move any content which is not policy into guide (VOTE required), link to policy from guide.
  - Review <http://jakarta.apache.org/commons/releases/> and extract anything useful.
  - Best practices - source and binary distributions should unpack to directories with different names. Use whatever-src for the source distribution (more convenient for those who have the source checked out). ✓
  - Add link from foundation release FAQ to best practices in incubator documentation ✓
  - Best practices - Build instructions should give supported version numbers for build tools (for example, maven and ant) ✓
  - Copyright - link to legal documentation: all source capable of copyright should contain license header. Easiest way to comply is to ensure that every human readable file has the header. Note that source includes not just the source code compiled into the final product but also all other resources such as stylesheets, test code and resources, build files and documentation source. When in doubt, add a header. ✓
  - Best practices - LICENSE and NOTICE files in jars so that they can be distributed by maven ✓
  - Add how to create a specification compliant MANIFEST <http://jakarta.apache.org/commons/releases/prepare.html#checkjarmanifest> ✓
  - Extract advice on release process from <http://jakarta.apache.org/commons/releases/>
  - Checks (source distribution):
    - Ensure that source distribution builds according to the instructions contained ✓
    - LICENSES!!! ✓
    - Check for copyright notices in files. Notes to above ✓
    - Check that document is readable ✓
    - Check libraries:
      - ASF policy on dependencies ✓
      - Appropriate license and notice files (where applicable) ✓
  - License for javadocs. This is a little controversial. Adding LICENSE and copyright information to javadocs is a little bit of a PITA. Some legal systems require explicit licenses so it may be worth the effort. ✓
  - [VOTE] and [RESULT][VOTE] convention for tally email. ✓
  - Voting on release candidates: pros and cons. Different processes: struts, tomcat, httpd vote to bless the same artifact. This is high ceremony and slow but reliable. Problem with voting on a release candidate is that it is not the artifact that will be shipped. Disadvantage of this approach is that if the candidate is not acceptable then it means either increasing the version number or risking the release into the wild (through repository) of a jar of unacceptable quality with the same version number.
- <http://incubator.apache.org/guides/projects.html> is just a landing page. probably not a good idea to take this one on ATM. Need a guide for mentors and projects which is what this was probably intended to be.
  - Need section on application of licenses
    - Link to appropriate sections of the
    - Note that source files include configuration etc
    - Note that license need to be applied to documentation as well
- [Guide To Entry](#) is a landing page with basic outline. lot of work required.
  - Starting Point Matrix - Open or Close Community; Open, Closed Or Even No Source but Must Bring An Active Community! Experience of the incubator and requirement are likely to differ for different people on the matrix.
    - Community, no code: the process is the longest and the most difficult. These projects are rare and probably require the active backing of one or more well known figure with proven ability in community building.
    - Open Source, Open Development in an existing open community: the process should potentially be quick. the major effort should be in completing the legal side (the provenance of the code needs to be completely tracked - this may mean that some areas of the code base may need to be replaced with fresh implementations; learning how to create releases that comply with apache policy; replacing or modularizing any dependencies that do not comply with apache policy). it's also useful for both sides to get to know each other better during this period.
    - Pre-proposal - the proposal is a formal document. In some cases, it can be useful to discuss whether there is any interest before the proposal stage. for example [http://mail-archives.apache.org/mod\\_mbox/incubator-general/200607.mbox/%3c154E4E82-C9FE-4EEE-8977-C01EA74C8EBE@apache.org%3e](http://mail-archives.apache.org/mod_mbox/incubator-general/200607.mbox/%3c154E4E82-C9FE-4EEE-8977-C01EA74C8EBE@apache.org%3e).
      - It's usually best to talk about any controversial aspects or anything a little bit new or unusual before the formal proposal stage.
    - Apache is about community. Quite a lot of interest is required to create a viable community without code. Unless you have a trackrecord in community building then people may well be sceptical. You may well . Please don't take this as being mean spirited - it's actually good advice. It's often easier to start a successful new project as a dictatorship. You don't have to worry

about recruiting new committers, about satisfying the Apache criteria for graduation or about the voting rules. There are a number of places where open source hosting is available at no cost. Later, you may find as the project matures and you have build your community you may decide that you want to propose the project for incubation. The Apache community model makes many things easier for developers of mature projects with large and active communities: it's legal support and organisational framework. Or you might decide that you want to create your own foundation similar to Apache probably with like minded projects. That's cool too. You don't have to be part of Apache to benefit from the legal documentation and organisational model. In fact, Apache has good relations with other open source foundations and this networking helps information flow all parties. Some other projects have benefited from advice from Apache members during the period when they set up their foundations. So it might be worth dropping back in.

- The vexed issue of names. Expect kickback.
- [http://mail-archives.apache.org/mod\\_mbox/incubator-general/200608.mbox/%3cc5e632550608061407j3124ededy43cd0654f9146647@mail.gmail.com%3e](http://mail-archives.apache.org/mod_mbox/incubator-general/200608.mbox/%3cc5e632550608061407j3124ededy43cd0654f9146647@mail.gmail.com%3e)
- Section on sub-projects - what this term means at Apache and why we don't like them 😊
  - Single governance model - the PPMC is responsible for everything that happens. All PPMCs should be on every list.
  - Single community
  - Single Karma
  - Multiple products is fine
- [Guide To Mailing Lists](#) needs tidying up. the index page has a missing link to this page. the website content should be replaced with a link to the website guide.
  - more content on mailing list best practices would be good but perhaps should be in the foundation docs
  - Mailing list netiquette. Add content from jakarta mailing list advice to foundation docs. Review content from other projects and add any that seems good. Add netiquette section and link from participation guide. link from netiquette section to foundation docs. Move content from participation guide.
    - Appropriate subject headers. This is a necessity on high volume lists. If the discussion has moved on, please edit the subject. ([WAS Re 😊
    - [OT] and other common prefixes [VOTE][PROPOSAL][POLL]
    - Do not send HTML
    - Attachments are often stripped (prevents propagation of viruses). Encourage people to create an issue in bugzilla or JIRA and then post a follow up to the lists including the URL
    - Remember that Apache lists are a public forum. They are a public record and are extensively and widely mirrored. Once an email is posted to a list the statements it contains will remain publically available for a long time. They are also highly indexed by the major search engines. Expect these mails to be highly ranked and visible. So, your reputation is at stake with every post.
    - Do not discuss on VOTE threads (it makes tallying much more difficult) instead change the subject
    - Do not top post
    - Cut extraneous material but leave enough to provide context
    - Don't cc the individual
    - It is the convention to speak for oneself and not for others. Drop the corporate line. Speaking with hats on. Many committers post from private addresses to reinforce this. <http://www.apache.org/foundation/how-it-works.html#hats>
  - Add board resolution on naming policy to official documents and link with commentary
- [Process Document](#) basically sound but will need updating to link into the new content. References the Jakarta guide for sub projects. This is not a good idea.
- [Roles And Responsibilities](#) too much duplication. Content needs to be consolidated and with more links added.
- [Learning Material](#) the rest of the material in here needs to find new homes. rules for revolutionaries is very badly formatted. the apache way has some good points but reflects older thinking from the membership: moving away from talking about the apache way and towards using open development as a term. perhaps history would be a better place some of this content. need a new document somewhere in www.apache.org describing open development.
- [PPMC Guide](#) Proof-read and decide upon a status and abstract. Merge in material from the wiki.
- [Branding Guide](#) Consensus about trademark and naming need to be added. Once public relations have web presence, add links.
  - Naming policy logically belongs in this guide
- [Guide Index](#)
  - fix broken links ✓
  - Update list so that drafts are included.
  - Need a short guide to choosing good names. Be best to approach PRC as well as poll
    - trademarks
    - good names for open source projects should be unique. marketing often chooses names for association
    - proper nouns are not always a good idea
    - development of names on list is encouraged
    - may be best to ask PRC to approve all names. this would centralise trademark checking and take some of the sting out of debate
- [WebSite](#)
  - The content served from the main Apache servers should be static. Dynamic content can be served from a zone. This is really infrastructure policy so may need to add to main site and link.

## Universal

- Add license headers - most of the documents are missing license headers. These need adding. (./)