CommonsEtiquette

Apache Commons Etiquette

This page describes and attempts to explain some of the peculiarities of Apache Commons. This is etiquette defined by the community rather than the formal rules (for those see http//commons.apache.org/commons/charter.html). This page was created (after several requests for information) to satisfy developers and committers rather than users. (But if anyone wants to add user etiquette, that'd be cool 🙂

The wiki seemed like a good place where this could be informally developed by the community. It might become a web page later (and then again, it might not).

The information contained in this page is unoffical advice rather than rules that have been officially adopted by the Commons team. It was founded by observations of the goings on in the commons from the subproject's creation till the present.

Mailing Lists

The commons components share just two lists. Questions about components are best asked on user@commons. Development is organized on dev@commons. Before posting questions to the list, users are requested to search the archives to see if the question has been answered before.

When sending mail, please add a prefix containing the name of the component. For example, if the mail concerns **commons-foo**, the subject should be prefixed with **[foo]**. The volumes on the development list are high. This can be difficult to manage without using filters to sort the mail. Most email clients allow filtering on subject and this convention allow developers to pick out the components they are interested in more quicky.

The Sandbox

The sandbox is a space provided for the development of experimental code by existing committers. It is divided into components.

Any ASF committer (not just Apache Commons committers) has the right ask for karma (commit access) and have it granted. The right place to ask is on the dev mailing list.

Components cannot be released from the sandbox. This means no binaries posted anywere on the apache site as "0.x" releases, as this implies that Apache supports the released code. Users of sandbox code are expected to extract the latest source from the source code repository and build the code themselves.

Commit access to the sandbox is unfortunately **not** available to people who are not existing ASF committers, no matter how good their idea. Such projects are generally encouraged to start somewhere like sourceforge.net. Once a solid community has been established and existing projects are using the component, it may be possible to integrate the project direct into commons if the project developers feel that is appropriate, and the commons community feels the component is a good fit with commons goals.

Sandbox Etiquette

Committers have karma for the whole sandbox. This means that a committer could alter any component. But we're all grown ups (right?) and so we'll play nicely together (right?). The right thing to do is to ask on the list or talk to the owners of the component (see the STATUS file) before diving in. The owners may not be still subscribed to the dev mailing list and so you might need to contact them directly.

A little patience and discussion in this will avoid flames later 🙂

Sandbox Oversight

Oversight for the sandbox is provided by the Commons team. Oversight means responsibility for ensuring that everything in the sandbox complies with Apache Software Foundation policies. Though (it is to be hoped that) committers should be able to be trusted, in the last resort the Commons team reserves the right to remove offending material from the source code repository.

If questionable material is found, the right way to proceed is to first raise the matter on the dev mailing list. The community will form a judgement about the material and the committers will be able to respond. It's a good idea to inform the committers directly (in case they are not on the list). It's also a good idea to copy the Commons PMC.

The Commons Proper

The Commons Proper consists of small, reusable components each developed independently.

Commons Committers

The commons proper is governed by the same membership rules as any other jakarta subproject. That means committers are elected. Unlike the sandbox, there are no automatic rights for existing Apache committers.

On the other hand, the barrier to entry for existing Apache committers has traditionally been pretty low. If you express an interest, show willing by creating a few patches and you're known to existing committers, there's usually no problem. VOTEs proposed by commons committers for existing Apache committers listing their achievements have (in the past) been passed very quickly.

Commons Etiquette

Every commons committer has karma to every component. But a point of etiquette (enforced by the charter) is that each committer who commits to a component must add their name to the STATUS file for that component. It's also good form to post a message saying 'hi, i'd like to get involved with component foo' before you dive in. Traditionally, adding your name to the STATUS file is a committers first commit on a component.

For components that use **Maven** as their build tool (and that is most of them now), you should add your name to the developers list in the project.xml file rather than the STATUS file if you intend to work on a component.

VOTEs and POLLS

A VOTE is an official decision thread. Anyone can express a preference (by indicating +1, +0, -0 or -1 in the traditional manner) but only some votes are binding. All are encouraged to VOTE but traditionally (to ease the work required to tally the vote) *(non-binding)* is added by those who know their votes are non-binding. Only votes from Commons PMC members are binding.

The dev mailing list is a busy place. Very much a bazaar rather than a Cathedral. This means that VOTE threads have a habit of petering out. It a good idea to post a `[RESULT][VOTE]` which counts the binding VOTEs and tells people the result. The tally should separate binding from non-binding VOTES. It is also good to place a time limit on [VOTE] threads. Finally, VOTE threads often digress into interesting discussions not directly related to the VOTE iteslf. In these cases, it is better to start new threads with appropriate subject headers for the side discussions. This makes it easier to navigate the list archives and also keeps the VOTE thread on track (or leads to it being stopped, where appropriate).

A POLL is an unofficial decision thread. These are useful for gauging the general feeling of the broader user community. Again, preferences should be expressed in the usual fashion. In this case, there is no need to indicate non-binding (since none are!).

Promotion

Promotion is the process by which components move from the sandbox into the commons proper.

Promotion to the commons proper is not the only route out of the sandbox. The commons proper isn't always the best place for all components from the sandbox. There isn't any reason why a component couldn't move directly from the sandbox to become a project or a subproject. At least one component has moved from the sandbox to sourceforge and then finally back to the incubator.

Promotion is basically a beauty contest. If the component can win enough votes and few enough people vote against it, then the component is promoted. But there is one thing that is most definitely required:

- Compliance with Apache Software Foundation policies. This means a full license at the top of every file. It means auditing the dependencies. It means ensuring the copyright date is correct on the licenses.

There some points of etiqutte and a few criteria that (though they are not rules) often seem to influence the voting.

- Evidence of compliance with the charter. This means having the documents required in the charter including a PROPOSAL. (Please look through the charter.) Please list all committers. Something like 'all commons-foo committers' isn't acceptable - a list is needed.
- A good PROPOSAL. A good PROPOSAL is clearly written and tightly scoped (ie. specific rather than general). Commons components are small, resuable components. The commons does not do frameworks and anything frameworkesque is likely to be viewed with scepticism. A PROPOSAL that duplicates an existing component will probably be viewed with suspicion. This is not because duplication is disallowed (overlapping components are specifically allowed by the charter) but because it indicates that the PROPOSAL fails to indicate the essential difference between the proposed component and the existing one. For example, a PROSPOSAL for a small, fast, compact xml-object mapper with minimal dependencies would be more likely to succeed than a PROPOSAL for 'a better version of commons-digester'.
- The health of the development community. Fellow committers need to be persuaded that users will be supported and the code pushed forward by the listed committers. This is a major issue since there's only a limited amount of energy amongst the commons committers and no one wants to have to support a component whose committers have gone AWOL.
- The people proposing the component. It's a sad fact of life but a PROPOSAL that comes from well known and respected Apache committers is more likely to be viewed positively than a PROPOSAL by people not well known to the Commons Team. Please don't get offended - you'll just need to work that little bit harder.

Ettiquette - Proposing A Promotion VOTE

- Discuss and try to formulate a consensus first. Promotion votes can (and do) get messy unless this happens. Create a discussion thread on the list and try to find out any reasons people might have for voting against. You might need to alter your charter, add missing files or resolve dependency issues. It's easy for everyone if all main issues are sorted before you propose the proper VOTE. If revisions to the proposal are required, the VOTEing can get very messy.

- Post a promotion email whose subject and body make it clear that this is a formal promotion VOTE. The subject should be prefixed by `[VOTE]` and should be something like `[VOTE] Promote commons-foo`. The body should be clear and fairly formal.

- Proposal - always include a copy of the PROPOSAL that's being VOTE'd on in the VOTE email. This is important since it is clear to everyone what they are VOTEing on. It also prevents being put in the embarasing position of the PROPOSAL being VOTE'd on being modifed in CVS half way through a VOTE thread.

- Please give the proposal enough time to give everything the chance to VOTE. I leave promotion VOTEs several days - maybe up to a week. When VOTEs have stopped coming in then please the proposer should post a `[RESULT][VOTE]` giving counts. Only the VOTEs of commons committers are binding so please make sure that these are tallied separately. The reason why a result email is good is that VOTE thread tend to peter out and so without a final email, it's hard to look back through the archives and find out what's happened. Another reason is that it's a good way to let everyone know what the result was. If there are any disagreements about the result, they can be resolved then.