Becoming an Apache Forrest committer

This is a discussion of how users can become committers within the Apache Forrest project.

Table of contents

1. What is a committer?

Warning:
This document is under development and does not yet represent the view of our community.

Note:
Content is being gleaned from various current and past discussions on the Forrest dev mailing list, in particular this. Further editing of this page is needed and would be greatly appreciated.

Committer is an term used at the ASF to signify someone who is committed to a particular project and who is invited to be part of the core group within the project that ensures the project's vitality (represented by the PMC, Project Management Committee).

One thing that is sometimes hard to understand when you are new to the open development1 process used at the ASF, is that we value the community more than the code. A strong and healthy community will be respectful and be a fun and rewarding place. Strong code will evolve.

2. Contributing to the Project - CoPDoC

The foundation of a project and how the community contributes to it is known by the acronym CoPDoC:

- (Co)mmunity - one must interact with others, and share vision and knowledge
- (P)roject - a clear vision and consensus are needed
- (Do)cumentation - without it, the stuff remains only in the minds of the authors
- (C)ode - discussion goes nowhere without code

3. Becoming a Committer

There is nothing at The Apache Software Foundation that says you must write code in order to be a committer. Anyone who is supportive of the community and works in any of the CoPDoC areas is a likely candidate for committership.

Apache is a meritocracy. That is, once someone has contributed sufficiently to any area of CoPDoC they can be voted in as a committer. Being a committer does not mean you commit code, it means you are committed to the project.

One of the key contributions people can make to the community is through the support of a wide user base by assisting users on the user list, writing user oriented docs and ensuring the user viewpoint is understood by all developers. A main idea behind being a committer is the ability to be a mentor and to work cooperatively with your peers.

The following diagram shows the progression of a user to a committer/mentor.

Meritocracy progresses this way `------------> ------------------------->`

Note that this is not a hierarchy, it is a progression from a broad user base from which those that wish to to contribute to the ongoing development of the project (again, through any aspect of CoPDoC, not just coding) can become involved as developers. From these developers are those who take on

additional roles of mentoring and more fully commit themselves to the project.

4. Responsibilities

The additional responsibilities of the PMC as a whole are outlined in the Apache Forrest project guidelines2. It should be noted though that Apache projects should be fun, not pressure. As a PMC member, just as a developer, you do as much work as you feel like doing. You do not need to participate in every discussion, just because it concerns the PMC. Neither do you need to participate in every vote or in every development issue. We like people to be involved and we reward contributions (meritocracy), but we do not punish a lack of contributions. People come and go as their needs change and we adapt to those changes.

5. Adding to the discussions and contributing code

Discussion leads to a clearer community understanding of the project's goals and objectives and also of how the community works.

Of course, there needs to be a balance between too much chat and not enough code. If something is easy to do in code and does not impact the overall product (such as a bug fix) then just go ahead and do it. However, if something is to introduce a new feature, then it is best to introduce your idea to the community via an email to the dev mail list first. In this introduction you should outline why you want to do something, how you propose to do it (pseudo code is a good way of expressing this) and ask for comments. Any comments received will help to fine tune the design and, in many cases, produce a quicker, more elegant solution (this is the benefit of many eyes on a solution).

The absence of comments from others does not mean it is not a good idea, in fact the reverse is true, it means nobody has any objection or anything to add. It is only if people respond that you need to discuss further. Once the discussion reaches consensus then coding can accelerate. Once you have implicit or explicit approval for your contribution, just go ahead and do it. Be sure to document what you have done whilst you are at it. Without documentation (comments in code, mailing list discussion and user docs) your code is next to useless - nobody knows it is there and nobody knows how it works.

Online discussion is important for community building. Offline discussion and one-to-one conversations deny the community the chance to become involved and lead to solutions that are not ideal. So please do as much discussion as possible on the dev or user mailing list.

6. References

1 Open development at Apache Forrest.

2 Apache Forrest project guidelines.