

JakartaPMCPropsedChanges

Changes to the Jakarta PMC

!THIS DOCUMENT IS WORK IN PROGRESS! It is not yet considered factually accurate or objective. Written initially by Stephen Colebourne based on reading far too many emails 😊

!IT IS PROBABLY NOW DEAD. TO BE REMOVED!

This document outlines the issues currently (end 2003/start 2004) facing Jakarta.

Jakarta and the ASF board

All code in Jakarta is owned by the Apache Software Foundation (ASF), a not-for-profit organization based in the USA. The ASF is governed by a board who have **legal** responsibility for all code produced using the Apache name, including Jakarta.

The ASF board delegates responsibility over the code to a number of [Top Level Projects](#) (TLP). These can be easily identified by the website address: Ant is a TLP (ant.apache.org), so is Jakarta (jakarta.apache.org), whereas Tomcat is not (jakarta.apache.org/tomcat).

What is significant about a TLP is not that it owns a website, but that it has **legal** responsibility for the code it looks after. This responsibility is delegated by the ASF board to a Project Management Committee (PMC) placed in charge of the project. Jakarta is controlled by the Jakarta PMC.

PMC responsibilities - Oversight and Releases

The [Jakarta PMC](#) is responsible for the *oversight* of all code in Jakarta. It must ensure that no harmful code is added (such as a virus) and that development progresses in a suitable manner. It does this by having at least one person (preferably many) from the PMC read every CVS commit message.

The PMC is also responsible for all releases of code from within Jakarta. Only the PMC may sanction a release, and only PMC members may vote on a release. The PMC is also responsible for voting on new committers.

The PMC does **not** have the authority to create any **ASF recognized** subprojects of its own.

Subprojects

Jakarta has created many subprojects such as Tomcat, Struts and Velocity becoming what is known as an *umbrella* project. However, these subprojects have **no formal status**. The Jakarta PMC is still responsible for each of these subprojects because it cannot delegate power. The Jakarta PMC must still provide oversight of the subprojects, authorise releases and vote on new committers.

PMC members and committers

There is a formal process within the ASF depending on the amount of dedication you show and the amount of responsibility you take. In summary:

```
Contributor (patches) -> Committer (authorized access) -> PMC member
```

A PMC member, has additional responsibility to a Committer. A PMC member must *oversee* the code, ensuring progress is made, and no damage caused. They also have a binding vote on the direction of the project, including releases. By allowing someone to become a Committer, you allow direct contribution to the codebase, but no voting rights. The PMC is overseeing them. The Committer contributes, but does not have a say.

Issues and Changes

If you have read the above and know anything about Jakarta, you will know that many of the above rules and legal restrictions are being broken. The ASF board has expressed concern on various occasions. The main area of concern is having one central PMC that is separate from the individual codebases.

Two changes have already occurred within Jakarta to combat this, and discussions are ongoing about other changes.

Change 1 - Bigger PMC

The original charter for the Jakarta project envisaged a PMC of seven (7) people. Clearly it is unrealistic for these seven people to oversee the 250 or so committers and 20 or so projects of Jakarta. Thus one solution is to increase the size of the PMC.

Recent elections have increased the number of PMC members to about 60 at present.

Change 2 - Subprojects to TLP

A number of Jakarta subprojects have been 'promoted' out of Jakarta to become TLP, notably Ant, Maven, James, Avalon and Log4J. These now have their own PMC and are responsible for their own destiny, reporting directly to the ASF board. They are still linked from the Jakarta home page.

Other Jakarta sub projects have indicated they wish to follow this course and become TLPs.

An example

Consider Struts, a successful Jakarta subproject. It currently has discussions and votes on the Struts mailing list which determine the future direction of that subproject. It also has committers and PMC members who contribute and oversee the code.

But there is a problem. Officially, the ASF does not recognize Struts as a project (its a subproject) - the ASF Board only recognizes Jakarta. Thus, the **only voters** allowed for decisions on Struts are **Jakarta PMC members**, and in fact a PMC member from Tapestry or BCEL has just as much right to vote as any Struts-based PMC member. Note also that Struts Committers have no rights to vote (unless they are also members of the Jakarta PMC). Of course, social norms prevent Tapestry PMC members from voting on Struts, but according to the ASF they do have the right.

One email from Greg Stein, Board chairman, tries to capture this: <http://mail-archives.apache.org/eyebrowse/ReadMsg?listName=general@incubator.apache.org&msgNo=2711>

Some Discussion

Under "PMC members and committers", we write "the Committer contributes, but does not have a say". This is true in a sense, but also false in a sense. Historically, the Jakarta PMC has followed the lead of the committers on a particular subproject. That is to say, suppose the Struts committers vote (perhaps from the ASF perspective this is more of a POLL than an VOTE) to take some action. The decision isn't binding until the PMC has approved it, often through lazy consensus (i.e., no one votes against it), because only the PMC has decision making authority. To say that the committers on a subproject don't have a voice in the decision making would be false. AFAIK, there has yet to be an example where a VOTE taken at the subproject level was overruled at the PMC level. On the other hand, to say that only the PMC can make binding decisions would be true. In practice it's more like the committers propose an action, and the PMC ratifies it via lazy consensus.