

How to Make Your Own ASF Project



Table of Contents

Online at: <URL:<http://Ken.Coar.Org/slides/incubator/>>.

The Apache Software Foundation is one of many organisations that supports the development of open software. Unlike a lot of them, though, its main focus is on community development rather than code. This presentation will give a brief outline of what it means to be part of an Apache project, and then describe the steps involved in bringing an existing project, or a new one, to petition to become part of the Apache organisation.

- [What's the Point?](#)
 - [The Process](#)
 - [The Sponsor](#)
 - [The Mentors](#)
 - [The Proposal](#)
 - [Licences](#)
 - [While You Were Incubating](#)
 - [Graduating](#)
-

[Ken A L Coar](#)

Copyright © 2005 by Ken A L Coar. All rights reserved.

Slide 1 of 9.

This slide last modified at Thursday, 31 March 2005 11:09:10 EST

What's the Point?



From its origins a few years ago, The Apache Software Foundation has grown to include dozens of projects, and more are being added all the time. What does it mean to be an Apache project, and how does one get created?

A new package may come from within an existing ASF project, or it might be coming from a previous life at some other location, such as [SourceForge](#) or [CodeHaus](#), or it might be a completely new concept that someone wants to *start* at the ASF. This presentation is about the approach from the outside.

As to why a project might want to be part of the Apache collection.. Each one has its own reasons, which may or may not include:

- Desire to be associated with the Apache name
- Developer(s) want to move on, and pass legacy code to someone else for retention
- Want to tap into and engage existing developer communities
- Package fills a niche in existing Apache spectrum
- Want to build a community around the package

From the point of view of the Foundation, not all of those reasons are valid, of course. The Foundation has a policy of *not* becoming a repository for abandoned code, nor a

home for limited-usefulness packages trying to trade on the Apache brand.

Packages that either have a thriving developer community, or seem likely to accrete one (based on the usefulness of the package), are much more likely to be acceptable.

[Ken A L Coar](#)

Copyright © 2005 by Ken A L Coar. All rights reserved.

Slide 2 of 9.

This slide last modified at Thursday, 31 March 2005 11:09:10 EST

The Process



Here's a view of the overall process of a package joining Apache, from 10'000m:

1. Find a sponsor
2. Find mentors
3. Make proposal to the Incubator
4. Deal with IP issues
5. Build community and demonstrate use and acceptance of the Apache Way
6. Graduate

All of these are covered in the following slides.

[Ken A L Coar](#)

Copyright © 2005 by Ken A L Coar. All rights reserved.

Slide 3 of 9.

This slide last modified at Thursday, 31 March 2005 11:09:10 EST

The Sponsor



What does the package want to be when it grows up?

Life in the Apache Incubator is meant to be a temporary thing — which means that a package (or *podling*, as an incubating project is called) needs to have a destination to which it will migrate after graduation.

There are two possibilities: Upon successfully leaving the Incubator, the package can either join an existing Apache TLP, or it can become one itself. The destination has an impact on the choice of sponsors.

If the package is going to be 'adopted' by an existing project, the Incubator just needs an assurance that the latter guarantees a place for the podling upon graduation. The proposal gets reviewed by both the TLP and the Incubator, and both the project and the Incubator will be sharing responsibility for guiding the package to eventual graduation.

If the package is destined to become a top-level project in its own right, then it needs to have either the Incubator itself or the Apache board of directors recognise its worth and agree to sponsor it.

Copyright © 2005 by Ken A L Coar. All rights reserved.

This slide last modified at Thursday, 31 March 2005 11:09:10 EST

The Mentors



A *mentor* is someone who is already part of the Apache structure and familiar with how things get done. A mentor acts as liaison between the podling and the Incubator, and sometimes between the podling and its sponsor. Mentors also provide advice, answer questions, and make suggestions to the podling in order to help guide it through the incubation process. A mentor should be one of the first sources consulted if there are organisational or procedural questions.

A podling may have multiple mentors, whose experiences may complement or overlap each other's. For instance, a podling destined to join an existing project might have a mentor from that project and from the Incubator itself — the one to guide the podling so that it will fit in upon graduation, and the other to ensure that all the necessary steps are taken to make the graduation happen.

[Ken A L Coar](#)

Copyright © 2005 by Ken A L Coar. All rights reserved.

Slide 5 of 9.

This slide last modified at Thursday, 31 March 2005 11:09:10 EST

The Proposal



The proposal to the Incubator needs to cover a few specific points, but its whole purpose is to describe what the package is and why it should be at Apache. Sometimes proposals are submitted in plain text *via* email, and sometimes they're built as pages in the Incubator's wiki. Here are links to some proposals that have been made in the past:

- [Nutch](#)
- [Apollo](#)
- [Hermes](#)
- [Muse](#)
- [MyFaces](#)

A sort of template has emerged, and proposals now typically include the following specific sections:

(0) rationale

What's the package about? What does it do? Why does it even exist? Who uses it?

(0.1) criteria

How well does the existing project already fit in with Apache methodology?

- Meritocracy
- Community
- Core Developers
- Alignment with existing Apache packages

(0.2) warning signs

Does the package exhibit any of the following symptoms? If so, acceptance might be contra-indicated.

- Is it an orphaned/abandoned package?
- Are the current developers inexperienced with open source?
- Are all the developers from the same company or organisation?
- Is development dependent upon contributions from salaried developers?
- Does the package have no ties to other Apache products?
- Does it appear that getting the Apache name is a prominent goal?

(1) scope of the subprojects

If the package is going to be adopted by an existing TLP, how does it fit in with the project and other packages already there?

(2) identify the initial source from which the subproject is to be populated

(3) identify the ASF resources to be created

What mailing lists, code repositories, *etc.*, will be needed?

(3.1) mailing list(s)

Typically one for users, one for developers, one for the PPMC, and one for change reports

(3.2) Subversion or CVS repositories

Which repository modules will need to be created, and are they to be CVS or SVN?

(3.3) Jira

What module(s) need to be created in the Apache bug-racking systems?

(4) identify the initial set of committers

(5) identify apache sponsor

Individual champion and/or sponsoring group, initial list of mentors

[Ken A L Coar](#)

Copyright © 2005 by Ken A L Coar. All rights reserved.

Slide 6 of 9.

This slide last modified at Thursday, 31 March 2005 11:09:10 EST

Licences



Before a podling can leave the Incubator and take its place as part of the overall Apache structure, the intellectual property issues need to be sorted out and clarified. Among other things, that means that any copyright holders need to have licensed their contributions to the ASF, that all the contributors have executed their own licence agreements with the Foundation, and that all of the code has been relicensed under the Apache licence.

In addition, searches need to be performed to attempt to ensure that any names involved in the package aren't infringing on existing marks.

[Ken A L Coar](#)

Copyright © 2005 by Ken A L Coar. All rights reserved.

Slide 7 of 9.

This slide last modified at Thursday, 31 March 2005 11:09:10 EST

While You Were Incubating



During the incubation process, infrastructure will be set up to support the podling's operation. This includes code repositories, mailing lists, and potentially entries in a bug-tracking system.

That's the 'hardware' aspect. The 'soft' side of incubation includes providing regular status reports to the Incubator, growing the community, and adopting as much transparency as possible — that is, doing everything in the open. One particular aspect of transparency that is sometimes difficult to adopt is that all technical decisions involving the podling **must** be made on public mailing lists. There is a natural tendency for people to just discuss things — especially potentially embarrassing things — in private or with friends, and this tendency in particular is one that both participants and mentors need to guard against with particular vigilance.

Software releases can be made while a podling is in the incubator, but only if they clearly identify the origin as being an incubating project and not an official part of the Foundation. Podling releases are also distributed only from the central Apache site, and aren't part of the extensive worldwide mirror system.

Graduating



Ah, yes — graduation day!

A podling graduates when the Incubator participants on the `general@incubator.apache.org` mailing list vote to let it happen. And that occurs only after someone proposes it for a vote, and only if all of the checklist items on the podling's status page are completed.

At graduation, the podling becomes officially part of the Foundation. Its releases are distributed through the mirror system, its repositories and mailing lists move out of the 'Incubator.Apache.Org' realm and to their final homes, and responsibility for the package's ongoing health and viability is passed to the appropriate project management committee.

Good luck! Don't forget to write!

[Ken A L Coar](#)

Copyright © 2005 by Ken A L Coar. All rights reserved.

Slide 9 of 9.

This slide last modified at Thursday, 31 March 2005 11:09:10 EST