ReleasePolicy

Core documents

The guiding principles for releases are found in Apache Release Policy and Release Creation Process. We follow the Apache Voting Procedure with the below modifications. In addition, the PMC has defined as policy that the build/README release procedure must be followed when making a release (it defines the SpamAssassin release process in great detail).

Who can make a release?

Any member of the Apache SpamAssassin PMC can make a release designated with Apache. Any committer of the Apache SpamAssassin project can perform most of the steps of producing a release package, but some of the final steps to making it an official Apache release, will require a member of the PMC with appropriate access agreeing to assume the role of Release Manager.

Who is in charge of a release?

The release is coordinated by the Release Manager (RM).To become an RM for a release, one must be a member of the PMC who volunteers to be RM. Any such person will be given access to the project signing key and passphrase, as well as PAUSE co-maintainer status to the Mail::SpamAssassin module on CPAN. The access to the signing key and PAUSE are only to be given to members of the PMC when they have a reasonable expectation of being a Release Manager for a release. It is an acceptable alternative for the Release Manager to generate a new signing key for the new release and publish the public key in all the places specified in the build/README documentation.

There is no set RM. Any PMC member may perform a release at any time. In order to facilitate communication, it is deemed polite to alert the community with your planned release schedule before executing the release. A release should only be made when there is a plan to publicly release it. A release by definition is not a build for private distribution within the SpamAssassin development community. Such a private distribution is referred to as a snapshot. Since a snapshot does not need to be signed with an official project key and since it can't be uploaded to CPAN, any committer can produce a snapshot build. An RM can delegate most of the process of producing a release to a committer, as long as the RM takes final responsibility for the result and handles the signing, upload to CPAN, and upload to the Apache release directories that also require PMC membership for write access.

When do I know if it is a good time to release?

It is our convention to indicate blocking issues in Bugzilla with the severity set to **critical**. A showstopper entry does not automatically imply that a release cannot be made. As the RM has final authority on what makes it into a release, they can choose to ignore the entries.

An item being denoted as **critical** indicates that the group has come to a consensus that no further releases can be made until the entry is resolved. However, issues that concerns a rule, even if the severity is critical or higher, is NOT a blocker to a code release because rules are released separately.

Pre-releases may be still be issued with one or more showstopper bugs, however.

The proposed release schedule is in ReleaseGoals.

What power does the RM wield?

Regarding when a release is made, the RM is the unquestioned authority. No one can contest what makes it into the release so long as the release procedure defined in build/README has been followed. The community will judge the release's quality after it has been issued, but the community can not force the RM to hold off a release for a feature. Remember that this document is only a guideline to the community and future RMs - each RM may run a release in a slightly different way.

How can an RM be confident in a release?

The RM must follow the release procedure defined in build/README. This is project policy set by the PMC and is not optional. Once the tree has been suitably tested by the RM and any other interested parties and the required procedure (including three +1 votes) has been followed, they should "roll" the release out. Of those 3 votes, 1 generally comes from the RM, and 2 from other committers.

What can I call this release?

There are three names for releases approved by the PMC:

- pre-releases (alpha): "3.1.0-pre1", "3.1.0-pre2", etc.
- release candidates (beta): "3.1.0-rc1", "3.1.0-rc2", etc.
- full release (general availability): "3.1.0"

The type of release needs to be chosen before beginning the release procedure since it is part of the tagging of the tree, etc.

Note the use of a dash between the version number and "pre"/"rc" strings; a dash is used, not a space, underscore, or no character at all.

Pre-release indicates that the release is not meant for mainstream usage or may have serious problems that prohibits its use. Release candidates are expected to work and perform basic tasks with no serious issues or work remaining (like optimizing the scores). However, there may be problems with this release that prohibit its widespread adoption. Full releases are recommended for production usage.

Typically, we do at least one RC before a full release.

Who can vote?

Committers & Non-committers may cast a vote for a release's quality. In fact, this is extremely encouraged as it provides much-needed feedback to the community about the release's quality. However, only votes cast by PMC members count towards the designation. Note that no one may veto a release. The PMC or the release manager may revoke all votes on a release if a new major problem is discovered prior to publication of a release and request a revote. In addition, the RM may decide against making a release even if the required votes have been made.

Full (general availability) releases require 3 +1's from PMC members and the release can be made as soon as a minimum quorum of three +1's from PMC members is met. See Apache Voting Procedure.

Pre-releases and RC's can be created and uploaded with just "lazy consensus" – ie. it's all go unless someone speaks up to object. However, you need to make it clear they are not equivalent to a "full"/"official" release – they cannot be uploaded to the "real" website download directories (ie www.apache.org/dist), or announced as a full release via a mail to the announce list. Make it clear that this is an unofficial "test build" by placing it in your public_html dir – e.g. https://people.apache.org/~jm/devel/ – for download.

Oops. We found a problem

If the "point of no return" described in the build/README has not been reached, then the release may be abandoned and begun again.

If the release has been made public and the problem is so severe that the release absolutely should not have been made, then the tarballs may be renamed to "-dontuse" and a new announcement may be sent at the discretion of the release manager.