

# Versioning

由 Myrle Krantz创建, 最终由 Markus Geiss修改于 一月 07, 2016

We utilize semantic versioning as defined here:

<http://semver.org/>

A short summary:

"Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards-compatible manner, and
3. PATCH version when you make backwards-compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format."

When in doubt, we will prefer creating a new minor release over creating a patch to an existing minor release. Working on multiple parallel branches of releases splits our capacity, and reduces our effectiveness as a community.

Fineract releases may be qualified by text following the version number, which all represent various steps in the pre-release process. In order from least- to most-stable:

Pre-Release Qualifier	Meaning	Example
snapshot[+<x>]	A release currently in development. The contents of x are optional, freely defineable, and mean something only to the person creating the snapshot.	1.1.0-snapshot+myrlesbuild23
m.<whole number>	Milestone. We know or expect there to be problems, but solicit testing from interested individuals. In case there are multiple milestones, they are numbered. The larger the number, the more advanced and (theoretically) more stable the release.	1.1.0-m.0
rc.<whole number>	Release Candidate. This release is a candidate to become the final proposed version. If no serious bugs are found, the -rc tag will be dropped, and the contents of this release will be declared stable.	1.1.0-rc.0
<none>	If there's no pre-release qualifier, it's a release, not a pre-release.	1.1.0

## First release

Our first release of Apache Fineract will have a major version of 0. It is not yet determined when we will iterate that release version to 1. We do not commit to avoid breaking compatibility inside of a major version until we iterate the release version to 1. (Once we've done that, we can delete this section too.)

## Milestone releases

When we want new features to get wide testing before we enter the formal stabilization period, we will sometimes release milestone versions. There is no requirement to do multiple milestone releases. We could just jump straight to "rc.0".

Milestones are for people who want to help test, and who understand that there may be incompatible changes before the final release.

## Breaking compatibility

Our compatibility rules only start to apply once a *final* x.y.0 version has been blessed and released. Any API's, wire protocols, and on-disk formats that appeared on trunk or in an alpha/beta/rc pre-release are not subject to *any* compatibility promises; we may change them arbitrarily until the final release, if we find a good reason to do so.

In particular we might not provide an upgrade path for persistent data with the final release. Consequently, pre-releases should **never** be trusted with any data meant for long-term safe-keeping.

At the same time, we wish to remind the reader that the best time to point out API design bugs is before they are released and set in stone during the initial design and pre-release stages.

(Thanks to the Subversion team for the content we 'borrowed' from the [Subversion release guideline](#))



Sander van der Heijden 发表：

Nice work, do we need to establish when a beta is warranted? Or maybe make that part of the voting on the change when it is sent in as a PR?

Big +1 on breaking compatibility! Definitely something we need sooner rather than later!



Myrle Krantz 发表：

After thinking about your question, I couldn't come up with any principle that would differentiate an alpha from a beta. So I've dropped the beta version. We can always do multiple alphas.



Myrle Krantz 发表：

### Reasoning behind changes

Put a dot between alpha and number: according to semver.org, if it's just a number, it can be ordered numerically. If we ever get ourselves into a situation where we need more than 10 alpha releases or release candidates we can still declare their order.

Started alpha and rc candidate numbering at zero: because starting at 1 when everything else starts at zero is inconsistent.

Removed beta: because I don't know how we would distinguish a beta from an alpha.

Added snapshot: so that we can, if necessary distribute builds while still in development.

Started major release numbering at zero: because this gives us the most flexibility at this early stage.



Markus Geiss 发表：

I always feel alpha as something bad ... what do you think about naming it milestone if we want to release a preview build.



Myrle Krantz 发表：

The problem with that (and with the current proposal), is that, according to the semantic versioning description, the pre-release part of the version is ordered lexically. In other words, "snapshot" is a later version than "alpha", and "milestone", and "rc". This is annoying, but if we're going to put stuff in maven, we should probably just ignore it.

I don't have a strong opinion on "alpha" vs. "milestone" or "m". I don't think we need both. Since no one else is expressing a different opinion, I'll swap in "m" for "alpha", and see if that gets someone to say something.

本站点授予Apache Software Foundation**Atlassian Confluence**开源项目**免费授权**。今天开始评估 Confluence。

This Confluence installation runs a Free Gliffy License - Evaluate the Gliffy Confluence Plugin for your Wiki!