

[Collapse All](#) | [Expand All](#)

- [Home](#)
- › [Archived](#)
- › [Acknowledgements](#)
- [Blog and News](#)
- › [Documentation v2-0-alpha](#)
- › [Documentation v1-6](#)
- › [Documentation v1-5](#)
- › [Documentation v1-1](#)
- › [Download](#)
- › [Developer](#)
- [Existing installations](#)
- [Feature requests](#)
- [Online Demo](#)
- › [Project organisation](#)
- [Roadmap](#)
- ▼ [Release Schedule](#)
 - [Release Schedule until v1-0](#)
- Release status and life cycle**
- › [Reviews and prototypes](#)
- [Support and Mailinglists](#)
- [Twitter](#)
- [FileSender's workflows](#)
- [Installation - Linux Source 2-0-Alpha-from-svn](#)

Release status and life cycle



Version 20, last updated by [xjansen](#) at 2016-01-06

This document describes the FileSender software life cycle. What labels do we use to describe the various states a release can be in, from birth (beta) to end-of-life (EOL). FileSender follows the well-known Debian lingo for describing the status of its releases.

Distribution channels and current release status

For all distribution channels (Tarball, Debian and RPM packages) the following channels (repositories) are used:

Status label	Description
--------------	-------------

oldstable	previous supported release (security fixes only)	1.5
stable	recommended and supported release	1.6.1
testing	next stable version	1.6.1
unstable	beta version	1.7-beta1
experimental	snapshot builds/experimental	1.7-beta1

Release history

Version	Status	Phase change planned around
1.5	oldstable	1.5 as of 6-01-2016
1.6.1	stable	1.6.1 as of 6-01-2016
1.6.1	testing	1.6.1 as of 30-12-2015
1.7 (beta + rc)	unstable	1.7-beta1 as of 6-1-2016

Release life cycle

All FileSender releases go through the following phases during their life:

Unstable: under development: nightly builds and beta builds

Testing: candidate to become **stable**. Production quality and well tested, but needs to prove itself in the field. Support, bug and security fixes are provided.

Stable: field tested boring-no-problems production release. Support, bug and security fixes are provided.

Oldstable: the previous stable release. Security fixes only.

Each channel can be occupied by one release. When we release a FileSender version for use in production environments, it moves from **Unstable** to **Testing**. This bumps the release living in Testing to Stable and the one living in Stable to Oldstable. The current Oldstable release dies.

When a release in Testing has run problem-free for at least an 8 week period on at least 2 production sites that see daily real traffic, it **can** (but does not need to) be moved to **stable**. A release in "stable" must be what it says on the tin: **stable**. It should run unattended and feel fine about it. We make an effort to have a solid release in both Testing and Stable at any given point in time.

When does support end?

When a release moves out of "Oldstable" support is no longer provided and the release is labeled **EOL, End of Life**. New *revisions* of a 'major.minor' will obsolete the previous 'major.minor', so for example '1.0.1' will obsolete '1.0.0'. The unstable/testing/stable/oldstable cycle is used for the major.minor releases. Minor revisions will be put in the distribution channel where the major.minor is.

Version numbering

FileSender version numbering follows the **major.minor.revision** method.

major: incremented when major back-end changes occur that require a lot of attention when upgrading. Examples are significant changes in the API, database schema etc.;

minor: when introducing changes too big to be labeled **revision** and too small to use

revision: small changes. This will typically be bug fixes, security fixes and UI polish.

Pre-releases

Beta releases: for major releases we will typically release one or more beta versions. These are tested by the testing team and are assumed to work reasonably well. Beta releases need to be field tested for robustness, bugs and assumption verification.

Release candidates: for both **major releases** and **minor releases** we will at least **1 Release Candidate** version.

Release: when a Release Candidate has been running on at least two FileSender sites without error for a period of at least 1 week under meaningful use, this release candidate is re-branded as a release. There are no code changes, no database changes and no config file changes between a release candidate and a release.

Pre-releases will be labeled with a `-{beta|rc}[n]` suffix, for example `1.1-rc2` or `1.5-beta1`

Snapshot releases

Between releases there can and will be intermediate snapshot builds. Snapshot builds are regarded as pre-release builds of a to be released target version. The snapshot version will be added to the target version number with a hyphen. *If there is no target version yet we could add the snapshot version with a + sign to indicate a post-release snapshot?*

Snapshot suffix: `-{builddate|svnrevision}` where `builddate = YYYYMMDDhhmm`

Examples: `1.5-201202041550` `1.5.1-r1488`

Package naming and versioning

Packaging principles

Main version of a package should be the "target" version to be officially released. Both alpha/beta/rc's and snapshots are to be considered a pre-release of that target version. When publishing packages the pre-release/snapshot version should be regarded as "lower" than the final target release but "higher" than the previous target release.

For RPM this is done using the Fedora recommendations by defining/using a Release tag.

For Debian this is (for many reasons) a bit more complicated when using both pre-releases and snapshots for the same target version. The use of a `-label` in the `upstream_version` part appears to be the recommended way.

Tarballs and zipfiles

tarballs and zipfiles will be named following the above mentioned versioning scheme with the name `filesender`

RPM packages

http://fedoraproject.org/wiki/Packaging:NamingGuidelines#Pre-Release_packages

http://directory.fedoraproject.org/wiki/Release_Procedure

In specfile Version=FS-version, Release=package-revision

[package-revision] = [n][.m][.fs-sub-version]

n=0 for pre-releases and dev builds

n=1 for releases

m=for pre-releases only: increasing number which will be incremented when switching from dev-build to pre-release or vice versa

1.5-0.1.r1488	nightly in 1.5 dev cycle
1.5-0.1.r1492	nightly
1.5-0.2.beta1	beta1
1.5-0.3.r1495	nightly
1.5-0.3.r1497	nightly
1.5-0.4.rc1	rc1
1.5-1	final release

1.5.1-0.1.[r1500](#) new nightly in 1.5.1 dev cycle

Debian packages

<http://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Version>
[epoch]upstream_version[-debian_revision]

<https://help.launchpad.net/Packaging/PPA/BuildingASourcePackage>

pre-release tags can (must according to policy) be in upstream_version like 1.5-rc1 *but* that would conflict with snapshot/dev builds.

1.5-beta1

1.5-rc1

1.5-201202042315

Comments are disabled for this space. In order to enable comments, Messages tool must be added to project.
[You can add Messages tool from Tools section on the Admin tab.](#)