Choosing the FreeBSD Version That Is Right For You

The FreeBSD Documentation Project

$FreeBSD: doc/en_US.ISO8859-1/articles/version-guide/article.sgml,v 1.16 2012/03/13 13:38:43 ryusuke Exp $

So you have chosen to install FreeBSD. Welcome! This document is designed to help you to decide which version to install.

Table of Contents

1 Background

To decide which version of FreeBSD is the best for your needs, it is important to understand a few concepts about our development and Release Engineering (`RE`) processes.

FreeBSD is developed by a large group of individuals who are almost entirely volunteers. The source code to the kernel and the most common libraries and utilities is maintained in a *source control system* and available for the general public to download at any time. Separately, compiled versions (`binaries`) are made available on a recurring basis. Some of these binary versions undergo an extensive testing process and are then termed a *release*.

1.1 Releases

`Releases` are named with a *major version number* and *minor version number*.

- The goal of a major release is to introduce a set of new features. Inevitably, as new features are added to FreeBSD, or as older features are no longer useful or are dropped, it is sometimes necessary to break compatibility with previous major releases.

- The goal of a minor release is primarily to fix bugs and improve performance and stability. Keeping both source-level and binary compatibility from one minor release to another is a priority. On occasion, new features may be added to a minor release when it is believed that these other goals will not be compromised.

However, keep in mind that a "release version" is merely a snapshot of the source tree at a particular point in time which is given a particular name (or *tag*). (For instance, the tag that Release Engineering assigned for the 5.4 release was `RELENG_5_4_0_RELEASE`.) Development always continues on what is known as the `HEAD` tag.

1.2 Branches

At the time of each release, a *branch* is created, such as `RELENG_5_4`. Although the source files named by `RELENG_5_4_0_RELEASE` will never change, those on `RELENG_5_4` can, by the adoption of changes from `HEAD` such as fixes for security and other bugs.

During the life of each major release, another tag is created such as `RELENG_5`. In addition to security and other bug fixes, other new changes can be brought in from `HEAD` so as to constitute the changes for the next minor release in the sequence.

1.3 *STABLE* versus *CURRENT*

During the lifetime of each major release, an individual branch may also be termed `STABLE`. This indicates that the FreeBSD Project believes that the branch is of sufficiently proven quality to be used by a wide range of users. Branches that need further testing before being widely adopted are named `CURRENT`.

> **Note:** The FreeBSD Project cannot in and of itself guarantee that the software that it ships is *stable* enough for any given installation. Only each individual user can make that decision. Please keep in mind that the Project is primarily composed of volunteers and is not able to offer any kind of warranty of fitness.

1.4 *Ports* versus *Packages*

Separately from the files distributed above, FreeBSD supports thousands of applications that are developed by third parties outside of the project itself. (Examples include windowing systems, Internet browsers, email programs, office suites, and so forth.) In general, the project itself does not develop this software, only the framework to allow these programs to be installed (termed the *Ports Collection*). Applications may be installed either from source, if its licensing terms allow such redistribution (these are called *ports*), or as compiled binaries if allowed (these are called *packages*).

2 Release Scheduling In The Past

During the development and release of FreeBSD 5.X, many lessons were learned which only became clear in retrospect. The goals of the 5.X series were very aggressive, and included:

- To provide support for Symmetric MultiProcessing (SMP) machines;

- To increase performance by adopting a new strategy for dealing with resource contention in the kernel;
- To add several new processor architectures;
- To introduce a new threading model;
- To introduce a new scheduler;
- To add support for new technologies such as power management (especially important for laptops); and, most critically,
- Not to declare the release series as `STABLE` until all these tasks were finished.

This led to a situation where it was several years between the time that a release in the 4.X series was declared `STABLE` and that a release in the 5.X series was declared `STABLE`. This had several undesirable effects:

- The number of simultaneous feature-set changes made it very difficult to isolate one set of changes for merging back into the `STABLE` branch;
- which meant that users who absolutely had to have certain new features (for instance, to be able to run on modern hardware) wound up adopting (for instance) FreeBSD 5.2.1 even though it was advertised as a developer-only release, and regardless of the fact that a `CURRENT` release was not really suitable for their needs.
- In the cases where backmerges were made, this put the developers in a position of trying to support features on a version that they themselves were not using as their primary development platform.
- The time lag also meant that when 5.3 was finally declared as the latest `STABLE` release, the accumulated weight of changes made the upgrading process very painful.

To put it bluntly, no one was satisfied with this result.

The lessons that were learned are:

- New major releases must have a smaller number of feature-set changes and be released more often.
- To the maximum extent possible, feature-set changes should be isolated from each other. (This implies that some development must take place outside of the main tree and only be merged in when it will not destabilize other simultaneous development.)
- Major releases should be targeted at a time deadline rather than a feature-set deadline. If some feature is not ready in time, then it should be disabled by default and left for the next major release.

By releasing smaller sets of changes more often, it is also hoped that less time will be spent trying to merge features from `HEAD` back into the latest `STABLE` version (and thus trying to support those features in more than one major version); and further, that as the changes are more isolated, that the risk of introducing more bugs by doing so is much less.

Also, by focusing on a time deadline rather than a feature set, it should finally be possible for users, developers of external applications, and the FreeBSD developers themselves to be able to better plan for the future.

3 Release Scheduling Goals Going Forward

Here are the current scheduling goals for the Project:

- To release a new major release every 18 months;

- To release a new minor release every 4 months;
- To provide prebuilt packages for the most recent minor release of each major version;
- To provide security updates and other critical bug fixes for the last several minor versions of each major version (termed *security branches*).

Due to the large number of possible combinations of installable versions, it is not possible to support every version indefinitely; this is due somewhat to the availability of machine resources but primarily due to the amount of volunteer effort that is available.

Interested readers should also see:

http://www.FreeBSD.org/releng/index.html#schedule

The Release Engineering Schedule

http://www.FreeBSD.org/security/security.html#supported-branches

The Security Branch Schedule

These documents go into much greater depth about the background and rationale behind the decisions regarding the supported branches and the lifetime of each branch.

4 How Do These Factors Affect My Decision?

The major factors that should affect your decision as to which version to install include:

- What degree of stability does your installation need?
- How much effort do you wish to devote to upgrading?
- How long do you intend to stay on a particular version between upgrades?
- How important is security to your installation?
- Will you be installing from source, or from binaries?
- Are you willing to help participate in the FreeBSD development process?

Here are some rough guidelines to help you in your decision:

- If you have a short-term need, would benefit from the highest degree of stability currently available, and are not able to devote many resources to upgrading, then you will probably want to install the latest `STABLE` minor release and remain with that. Depending on your security needs, you may or may not wish to track changes to that branch as they are released.
- If you have a short-term need, and feature completeness or the best levels of security are most important to you, and you are willing to spend time upgrading, you may wish to track the latest `STABLE` branch.
- If you are not immediately planning on going into production, you are willing to work through a certain number of problems, and a new major release is upcoming within the next few months, you may wish to investigate installing that branch to help the project to test it and stabilize it to make it the best release possible in the medium to long term.

- Only if you are willing to install from source, and spend time debugging problems with the base system and follow up with problem reports, and track the mailing list that deals with such issues, should you consider tracking `HEAD`.

5 Conclusion

We hope that this article serves as a useful starting point for understanding the FreeBSD development model and deciding what version is right for your needs.