

Live demo in [BSD Now Episode 009](#) .

In most of the BSDs, there are different branches (or "versions") of the OS that you can use. Often times in addition to the normal releases, there is a development version with the latest features. In FreeBSD, there are three main branches of the OS you can use: -RELEASE, -STABLE and -CURRENT. Each of them is aimed towards a different set of users and their needs. This tutorial will explain the differences between them and why you may or may not want to use a certain one. You'll also learn how to go from one to another, when possible. One of the advantages of doing a source upgrade (as is required for development branches) is that you can easily strip out parts of the OS that you don't want or need. We'll discuss that more as we go on. Let's start with a brief explanation of what each branch is and what it's mainly intended for.

-RELEASE

The -RELEASE branch is arguably the most stable branch of FreeBSD. That's exactly what it's designed to be, what the everyday end user installs and uses. When a -RELEASE is cut, the only updates it gets are either critical fixes or security flaws. Most users should use -RELEASE because its codebase is the most thoroughly tested. Binary upgrades between -RELEASE versions are also very fast and easy, using the `freebsd-update` tool. Keeping this branch up to date with the latest patches is very simple, especially since there are very few patches needed for it. Security fixes are announced on multiple mailing lists, so it's strongly recommended to subscribe to at least [the main one](#).

Patching a -RELEASE branch for security fixes (on a supported version) is as easy as:

```
# freebsd-update fetch install
```

That's it! Signed binary patches will be downloaded and applied. The announcement email will tell you if you need to rebuild any ports (as is the case for some OpenSSL updates), restart any services or reboot (usually only needed for kernel updates). FreeBSD development moves fast, and there comes a time when you need to upgrade the OS to a whole new version if you

Freebsd Tracking -STABLE and -CURRENT

Written by Friends

Friday, 27 December 2013 09:01 -

want to keep getting security fixes. If you're running a -RELEASE, you can also use the `freebsd-update` tool for major upgrades. If you wanted to go from 9.1-RELEASE to 9.2-RELEASE, all you need to do is this:

```
# freebsd-update -r 9.2-RELEASE upgrade
```

All the needed files will be downloaded and put into place. You will be asked to merge some configuration files if needed. Two reboots will be required (one for kernel, one for userland), but `freebsd-update` will tell you everything you need to do. For more info, see [this page](#).

-CURRENT

FreeBSD -CURRENT is where all the new changes go first - all the new features, all the big updates, all the freshly-written code. This is the bleeding edge of the OS' development. It is sometimes highly experimental and/or unstable. Most of the time it's not - I've heard people say "FreeBSD -CURRENT is more stable than an Ubuntu release" - but I digress. If you're using -CURRENT, you are more or less expected to be able to fix your own problems, or provide helpful information so that the developers can find and fix whatever bug you've found. People who need or want the latest and greatest code should consider running this branch. While the security team doesn't officially support -CURRENT, all needed security fixes will be committed. If the security issue only affects -CURRENT, you'll have to watch the commit logs to find out about it. If you want to keep up with this branch, you should join these two mailing lists to see changes and any workarounds needed: [freebsd-current](#) and [svn-src-head](#).

To go from -RELEASE or -STABLE to -CURRENT (note: you can't really go backwards easily) you'll need to checkout the latest SVN sources, then rebuild the kernel and userland. Let's remove your current sources if you have them, then install subversion from ports and go through the process.

```
# rm -rf /usr/src # cd /usr/ports/devel/subversion # make config-recursive install clean
```

There's also an "svnup" port in `/usr/ports/net`, and a subversion client in the base OS starting with FreeBSD 10. This is assuming you're running a version before 10, so we need to install it from the port as described above. Now we grab the source code from one of the mirrors.

```
# svn co https://svn0.us-west.freebsd.org/base/head /usr/src
```

For a list of SVN mirrors and their fingerprints, see: [svn mirrors](#). You should verify the SSL fingerprint if you're using HTTPS. After the initial checkout (which may take some time), you can sync your sources with the latest by issuing the following command:

Freebsd Tracking -STABLE and -CURRENT

Written by Friends

Friday, 27 December 2013 09:01 -

```
# svn update /usr/src
```

It's a good idea to read over `/usr/src/Makefile` the first time you do this, so you can better understand what make targets are and which ones to use. Another file you should regularly check before doing big upgrades is `/usr/src/UPDATING`. It documents needed workarounds for certain changes. After you've familiarized yourself with those two files, now might be a good time to remove any parts of the OS you're not interested in compiling or using. To do this, edit the `/etc/src.conf` file. Consult [src.conf](#) for a list of things you can remove, add or change. You can also customize the kernel, taking out things you don't want or adding things that aren't in `GENERIC`. For a brief explanation of custom kernels and how to configure one, see [this page](#).

Next, we build the userland and kernel, then do the initial merge of configuration files in case there's anything needed for a successful upgrade.

```
# cd /usr/src # make -j `sysctl -n hw.ncpu` buildworld # make -j `sysctl -n hw.ncpu` buildkernel
## Add "KERNCONF=yourkernelname" if you did a custom kernel. # make installkernel
## Add "KERNCONF=yourkernelname" if you did a custom kernel. # reboot
## It's recommended to go to single user mode, but not required. # cd /usr/src #
mergemaster -p ## If anything needs to be merged, added or removed. #
make installworld # mergemaster -iUF ## You will merge the new config files
with your old ones. # yes | make delete-old # yes | make delete-old-libs # cd /usr/obj &&
chflags -R noschg * && rm -rf * # reboot
```

Your system should now come back up as a `-CURRENT` installation! Remember to check the lists and files mentioned so far so you can stay up-to-date with the latest developments. If you went from one major version to another, for example `9.2-RELEASE` to `10-CURRENT`, you will need to rebuild all ports with `portmaster` or `portupgrade`. Systems running `-CURRENT` cannot be upgraded using the `freebsd-update` tool, you have to do source-based upgrades, recompiling the OS.

-STABLE

FreeBSD `-STABLE` is what the `-RELEASE` branch is made from. After new features have been tested for a while in `-CURRENT`, they're often merged into the `-STABLE` branch. Some big features stay in `-CURRENT` until a new major point release, but a lot of the improvements that would be beneficial to the next `-RELEASE` will be committed to `-STABLE`. Security fixes for `-STABLE` are announced on the regular mailing lists mentioned earlier, so you should really subscribe to it regardless of what branch you're running. New features in `-STABLE` become the

Freebsd Tracking -STABLE and -CURRENT

Written by Friends

Friday, 27 December 2013 09:01 -

next point release. For example, after 9.1 was released, all the new changes in -STABLE will eventually become 9.2. There's a process of BETA and RC (release candidate) tests before that happens. This branch should be considered a lot more mature and stable than -CURRENT, but there will be a few times when it may not build or work quite as expected. Problems are fixed very quickly, however. It's a good mix between "only security updates" and "new stuff every day." As recommended in the previous section, you should subscribe to a couple mailing lists to see important changes. [freebsd-stable](#) and [svn-src-stable](#) (for 9.X, replace if you're on a different major version).

If you don't already have subversion installed, let's install that and check out the source code.

```
# rm -rf /usr/src # cd /usr/ports/devel/subversion # make config-recursive install clean # svn
co https://svn0.us-west.freebsd.org/base/stable/9 /usr/src ## Replace 9 with the branch you
want.
```

For a list of SVN mirrors and their fingerprints, see: [svn mirrors](#) . You should verify the SSL fingerprint if you're using HTTPS. After the initial checkout (which may take some time), you can sync your sources with the latest by issuing the following command:

```
# svn update /usr/src
```

It's a good idea to read over /usr/src/Makefile the first time you do this, so you can better understand what make targets are and which ones to use. Another file you should regularly check before doing big upgrades is /usr/src/UPDATING. It documents needed workarounds for certain changes. After you've familiarized yourself with those two files, now might be a good time to remove any parts of the OS you're not interested in compiling or using. To do this, edit the /etc/src.conf file. Consult [src.conf](#) for a list of things you can remove, add or change. You can also customize the kernel, taking out things you don't want or adding things that aren't in GENERIC. For a brief explanation of custom kernels and how to configure one, see [this page](#)

Next, we build the userland and kernel, then do the initial merge of configuration files in case there's anything needed for a successful upgrade.

```
# cd /usr/src # make -j `sysctl -n hw.ncpu` buildworld # make -j `sysctl -n hw.ncpu` buildkernel
## Add "KERNCONF=yourkernelname" if you did a custom kernel. # make installkernel
## Add "KERNCONF=yourkernelname" if you did a custom kernel. # reboot
## It's recommended to go to single user mode, but not required. # cd /usr/src #
mergemaster -p ## If anything needs to be merged, added or removed. #
make installworld # mergemaster -iUF ## You will merge the new config files
with your old ones. # yes | make delete-old # yes | make delete-old-libs # cd /usr/obj &&
```

Freebsd Tracking -STABLE and -CURRENT

Written by Friends

Friday, 27 December 2013 09:01 -

```
chflags -R noschg * && rm -rf * # reboot
```

Your system should now come back up as a -STABLE system! Remember to check the lists and files mentioned so far so you can stay up-to-date with the latest developments. If you went from one major version to another, for example 8.4-RELEASE to 9-STABLE, you will need to rebuild all ports with portmaster or portupgrade. Systems running -STABLE cannot be upgraded using the freebsd-update tool, you have to do source-based upgrades, recompiling the OS.