

# FreeBSD

## Ports and Packages



With time you will have many files in the directory 'distfiles', these files are compressed source code packages. To get rid of the distfiles you don't need any longer, please execute:

```
# portsclean -D
```

If you only want to clear out all distfiles unneeded by any currently installed Port, please execute:

```
# portsclean -DD
```

### Remove Ports

You may remove Ports with `pkg_deinstall`, which is a part of Portupgrade. You may also use `pkg_delete` instead. If you want an easier way, you may also use the Port `sysutils/pkg_cutleaves` which finds all packages that are no longer needed by any dependency of another program and which suggests the removal of each package individually for maximum control.

### Graphical Package Managers

Of course there are also graphical interfaces for people who don't want to use the console. Beside the ncurses-based `sysutils/pkgfe` and `sysutils/portbrowser` there is also the outstanding `sysutils/desktopbsd-tools` which is based on KDE and is the standard in DesktopBSD and PC-BSD. With these tools even a beginner can install and manage Ports with ease.

### Further Information

If you want to know more about Ports we suggest taking a look at the excellent FreeBSD Handbook:

<http://www.freebsd.org/doc/en/books/handbook/>

Further sources are here:

[http://www.freebsd.org/doc/en\\_US.ISO8859-1/books/porters-handbook/](http://www.freebsd.org/doc/en_US.ISO8859-1/books/porters-handbook/)

<http://www.lpthe.jussieu.fr/~talon/freebsdports.html>

### Portupgrade

Portupgrade was developed to simplify updating and deleting of Ports. Like nearly all Ports utilities you can find it in the category "sysutils". For installation please execute:

```
# cd /usr/ports/sysutils/portupgrade && make install clean
```

Prior to updating your ports please search the list of installed Ports with

```
# pkgdb -F
```

for possible problems and resolve them. If you execute

```
# portupgrade -a
```

the automatic update of all Ports begins where new versions are available. Use '-i' to switch to interactive mode if you want to have more control:

```
# portupgrade -ai
```

If you only want to update specific programs instead of all, use the command

```
# portupgrade pkgname
```

Use '-R' to switch to recursive mode and upgrade all dependent packages too. The switch '-P' uses Packages instead of Ports. With this option portupgrade searches the directories defined in the environment variable `PKG_PATH` for Packages. If they are not available locally, portupgrade tries to download them over the network. If they are not available locally nor on remote machines then Ports are used. Portupgrade is very powerful, absolutely read the [manual page](#) before use!

### Portaudit

To help recognize security vulnerabilities the program `security/portaudit` can help you.

```
# portaudit -Fa
```

lists all known security holes in the programs installed on the machine you execute portaudit.

Of course you can schedule runs with cron. You can also use portaudit before the installation of a Port or Package and check for possible vulnerabilities.

### Making Space

The Ports Collection (Porttree) can become very large. If you don't always use 'clean' when installing a Port you can clean out all Ports on the whole system in one go by using `portsclean` which is part of portupgrade

```
# portsclean -C
```

### Introduction

An operating system like FreeBSD is the software which enables use of a computer. It manages memory, input and output and controls the execution of programs. Without programs like browsers, text editors, mail programs etc. an operating system is of little use.

A question often asked is whether FreeBSD has enough applications. FreeBSD has more applications than any other freely available operating system including Debian (please note that Debian are very different from Ports). With **over 16'000 applications** we're confident you will find the ones you need. Don't forget that nearly all Linux applications also run on FreeBSD with very few exceptions.

You may ask yourself now how the applications are installed, updated, audited and managed in FreeBSD. The answers are in this flyer.

### Ports and Packages

To install software FreeBSD offers two complementary methods: The Ports collection for installation from source and Packages for installation of pre-compiled binaries. You may use both methods in conjunction to install your programs with local media or over a network.



The FreeBSD Package of an application consists of a single file you need to download. The package contains all commands of the application, additional configuration files and documentation. FreeBSD offers the tools `pkg_add`, `pkg_delete`, `pkg_version` and `pkg_info` to manage packages. With the command `pkg_add` you install the program you want automatically.

The FreeBSD Port of an application is a collection of files that automate the compilation of a program. The files of a Port execute all necessary steps for installation for you. With just a few commands the source code of a program is downloaded, unpacked, patched, compiled and installed. Ports can also be used to create Packages of your own.

Packages and Ports respect dependencies between applications. Let's say you want to install a program dependent on a library and both are available as Package and Port for FreeBSD. When you use `pkg_add` or the Ports system you notice the library was installed first if it wasn't already installed on the system.

Why two systems? Both systems have their advantages and which choice you make is up to you.

#### Advantages of Packages

The compressed package of an application is normally smaller than the compressed archive of the source.

Packages don't need to be compiled, which is of great advantage with large programs like KDE, GNOME, OpenOffice and others.

If you use packages you don't need to understand how to compile software on FreeBSD.

#### Advantages of Ports

Since the packages should run on most systems, compile options are used sparingly. When you install an application with Ports you can optimize the installation with compile options. Apache can be configured with many integrated options. When you use the Ports system you can overwrite the default options.

Sometimes there are multiple packages of a given application. For `ghostscript` there is a `ghostscript-package` and a `ghostscript-nox11-package` without support for X. This rough distinction is possible within the package system but quickly gets unwieldy when an application has more than two options.

The licensing of some software does not allow distribution in binary form. This software must be delivered in source code.

Some people don't trust binary distributions. When you own the source code you can audit it before compiling it and make sure it doesn't harbor nasty surprises.

Local changes and customizations also make possession of source code necessary.

### Freshports: How do I find Applications?

If you can't remember the name of an application you can go to sites like FreshMeat (<http://www.freshmeat.net/>) to look for a suitable application. Then look also on [www.freshports.org](http://www.freshports.org) where you can search by category and see which new applications have been added, which programs currently have security vulnerabilities or have other problems. Have a look, but take care, it's addictive!

### Installing the Ports Collection

Before installing Ports you must add the Ports Collection (Portstree), it consists of makefiles, patches and descriptions. It is typically installed under `/usr/ports` but you may choose another path if desired. You can use `Sysinstall` (the FreeBSD-installer), `cvsup` or `csup`. We recommend using `portsnap` instead which is in FreeBSD base since FreeBSD 6.1. It is extremely easy to use

1. The first time `portsnap` is run, it will need to download a compressed snapshot. This is approximately a 42 MB download.

```
# portsnap fetch
```

2. Once the compressed snapshot has been downloaded, a "live" copy of the ports tree can be extracted into `/usr/ports/`. This is necessary even if a ports tree has already been created in that directory (e.g., by using `cvsup`), since it establishes a baseline from which `portsnap` can determine which parts of the ports tree need to be updated later.

```
# portsnap extract
```

3. After an initial compressed snapshot of the ports tree has been downloaded and extracted into `/usr/ports/`, updating the ports tree consists of two steps: *fetching* updates to the compressed snapshot, and using them to *update* the live ports tree. These two steps can be specified to `portsnap` as a single command:

```
# portsnap fetch update
```

If you want to automatically update your Portstree you can use cron as well.

### Installing Ports

If you have an updated current Ports collection and root privileges then you can install Ports. Ports assume your system has working internet connection. Let's say you want to install your beloved Bash shell you know from your old and obsolete Linux system. Change to the directory of the Port, build it with `make`, install it and clean up the intermediate files from the working directory. You are scared? Please don't worry, it's very easy!

```
cd /usr/ports/shells/bash/ && make install clean
```

It's as easy as that, go to the directory, make it, install it and clean it. What about installing the package instead?

```
pkg_add -r bash
```

It doesn't get much easier than that. Some Ports also have additional functions and security options. Additional options can be added e.g. for [www.mozilla](http://www.mozilla.com) and [mail/sylpheed-claws](http://mail.sylpheed-claws.com). When a Port has additional options then these are displayed when making the Port.

### Updating Ports

People new to this system are prone to constantly feel the need to update their Ports. This is not necessary.

After updating your Ports tree with

```
# portsnap fetch update
```

**Please make absolutely sure you have read and understood `/usr/ports/UPDATING` before updating a Port!**

This file describes potential problems and their solutions for updating a Port. Many problems with programs not working after an unplanned update can be avoided by reading this file beforehand.

Large programs like KDE and GNOME can be also very easily updated with

```
# pkg_delete -r and a subsequent # pkg_add
```

A list of all installed Ports and their version status can be displayed with:

```
# pkg_version -v
```

Of course there are many different tools available to update or manage your Ports, like `portmanager`, `portmaster` and `portupgrade`. Lacking space to describe them all we concentrate on the latter.

## FreeBSD: The Power To Serve