Introduction to FreeBSD Ports

(25 years and counting...)

linux.conf.au 2020 - FreeBSD Miniconf

Ben Woods, 14th January 2020

About Me

Ben Woods — woodsb02@FreeBSD.org

FreeBSD ports committer:

- Since 2016-05-09
- Ports maintained: 59
- Commits to head: 440
- I got involved to scratch my own itch - to help improve programs I used
- This is a hobby for me

Day Job ($):

- Industrial Control Systems Engineer
- Australian Oil & Gas company
- I have been unsuccessful at getting these 2 to intersect (so far...)

Agenda

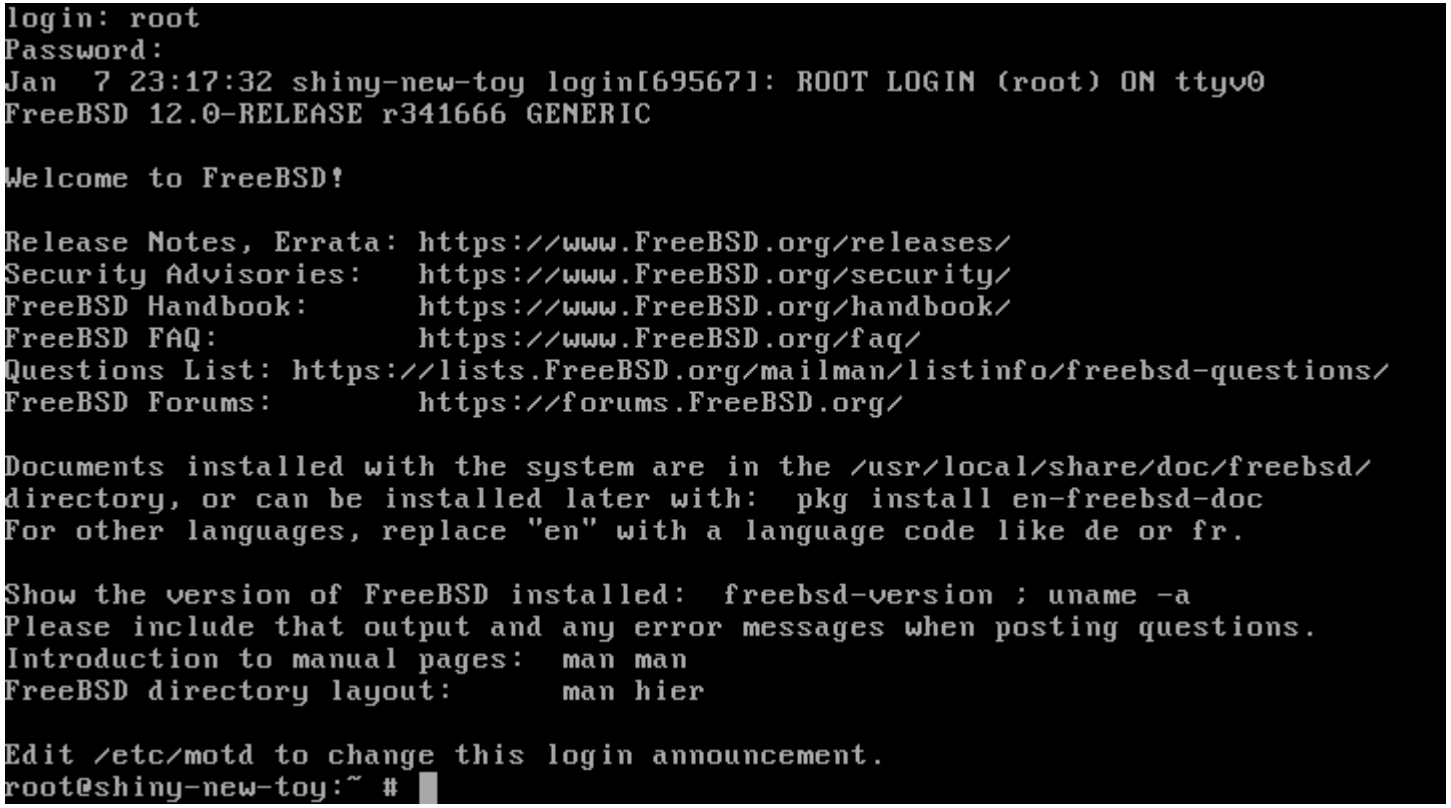
1. What are ports?
2. How to use ports
3. Ports tree internals - how it works
4. Ports development - some examples
5. How you can contribute
6. Becoming a ports committer
7. Poudriere

What are ports?

Section 1

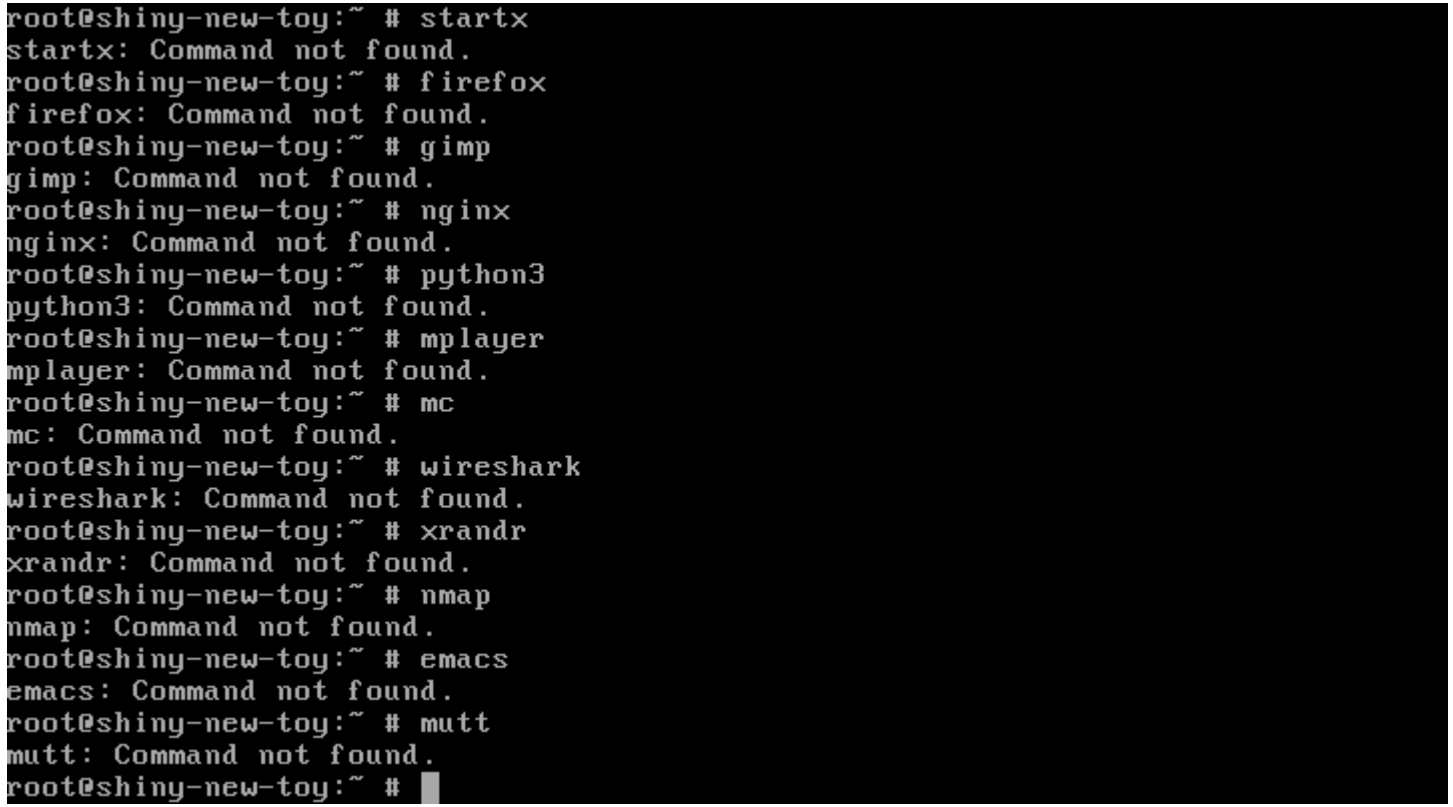
You've just finished installing FreeBSD...

Now what?



FreeBSD has many system tools in the base system...

But perhaps you want to run some other programs?



Ports — The FreeBSD **App Store**™ *

* Ports existed long before the term "App Store" was claimed as trademark

But what are ports?

Formulas to simplify installing third-party programs on FreeBSD

The formulas include all the typical steps required to install software:

- User selection of build options
- Fetch
- Extract
- Patch
- Configure
- Build (compile)
- Stage
- Package
- Install

Bonus: They also provide a consistent way to install all apps

Who makes these apps?

The apps themselves are developed by "others" (yay for open source!)

The FreeBSD project maintains:

- the formulas
- any modifications necessary to make the programs work on FreeBSD *

* Typically to remove "Linuxisms"

FreeBSD Ports History

2019 marked the 25th anniversary of FreeBSD ports!

- Created by Jordan Hubbard →
- 1994-08-21 - First ports framework committed
- 1993-08-26 - pkg_install tool was committed

https://en.m.wikipedia.org/wiki/FreeBSD_Ports#History

Photo by Faces of Open Source / Peter Adams

Quote from Jordan Hubbard (1/5)

> *“*
>
> *The ports collection arose out of nothing more significant than sheer laziness on my part. I was installing a lot of FreeBSD machines from scratch in those days, particularly as I was provisioning a lot of new servers to deal with our growth, and*
>
> ***I got really tired of grabbing source tarballs from various FTP sites, unpacking them, applying any necessary patches, then configuring, building and installing them.***
>
> *It's hard to remember this now, but in the days before pretty much everything you might want was packaged in binary form, you had to build everything from source, and there were no "recipes" for doing this - you just did it all by hand using memory and tribal knowledge about what configuration flags to use and environment variables to set.*
>
> *”*

Quote from Jordan Hubbard (2/5)

> “
>
> *All of this was incredibly labor-intensive, and I was already something of a ninja at using make(1), having written at least one version of make(1) myself earlier in the same decade, so I decided to use BSD make's robust macro (.mk) mechanism to see if I could generalize and automate all of this work I was doing.*
>
> ***From the initial idea to first POC implementation took about 24 hours***
>
> *and I got the first 10 ports done the next day, then I went back and refactored the macro files and went through a few more iterations before committing the first version of the macros as well as my reference ports*
>
> *(see https://www.slideshare.net/JordanHubbard/clipboards/my-clips?rftp=success_toast for the first 70 ports).*
>
> ”

Quote from Jordan Hubbard (3/5)

> “
>
> *Once we got to 300 or so ports, I managed to hand responsibility for the whole thing to **Satoshi Asami**, who was the project's **first portsmeister** and responsible for truly refining and scaling the ports collection up significantly, as well as recruiting the first group of ports committers.*
>
> *The project owes Satoshi and the initial ports committers a significant debt of gratitude for really taking it to the next level!*
>
> ***I can't believe there are over 32,000 ports today.***
>
> ”

Quote from Jordan Hubbard (4/5)

> “
>
> ***Not so fun fact:*** *I wrote the package management tools and started publishing binary packages before I wrote the ports collection, so I later had to go to a considerable amount of work to make the two systems work together. With the benefit of hindsight I would have done ports first and packages second, yielding a better architecture, but we eventually fixed that!*
>
> ”

Quote from Jordan Hubbard (5/5)

> “
>
> ***Fun fact:*** *I live with 14 cats and 7 dogs, so I have replaced software husbandry with animal husbandry, though the pets are a lot cuter than a lot of software I've seen over the years.*
>
> ”

Similar Frameworks

Other *nix systems also created similar "app stores":

- 1994 - Debian created dpkg
- 1997 - NetBSD gained pkgsrc
- 1997 - RedHat's RPM tool was created
- 1998 - Debian created APT
- 1999 - YUM was created
- 2002 - Arch Linux's Pacman were released
- 2002 - Gentoo's Portage was released - inspired by FreeBSD ports *
- 2005 - The Arch User Repository (AUR) was created

> “
>
> *Most of all, I was impressed with FreeBSD's ports system, the technology used to maintain and upgrade the system. Unlike the Linux approach, ports didn't use binary packages but instead automatically compiled everything locally from their original sources.*
>
> ”

* https://web.archive.org/web/20051126225230/http://www.gentoo.org/doc/en/articles/making-the-distro-p3.xml

What's in FreeBSD Base?

- kernel and device drivers
- man pages
- userland utilities (command line):

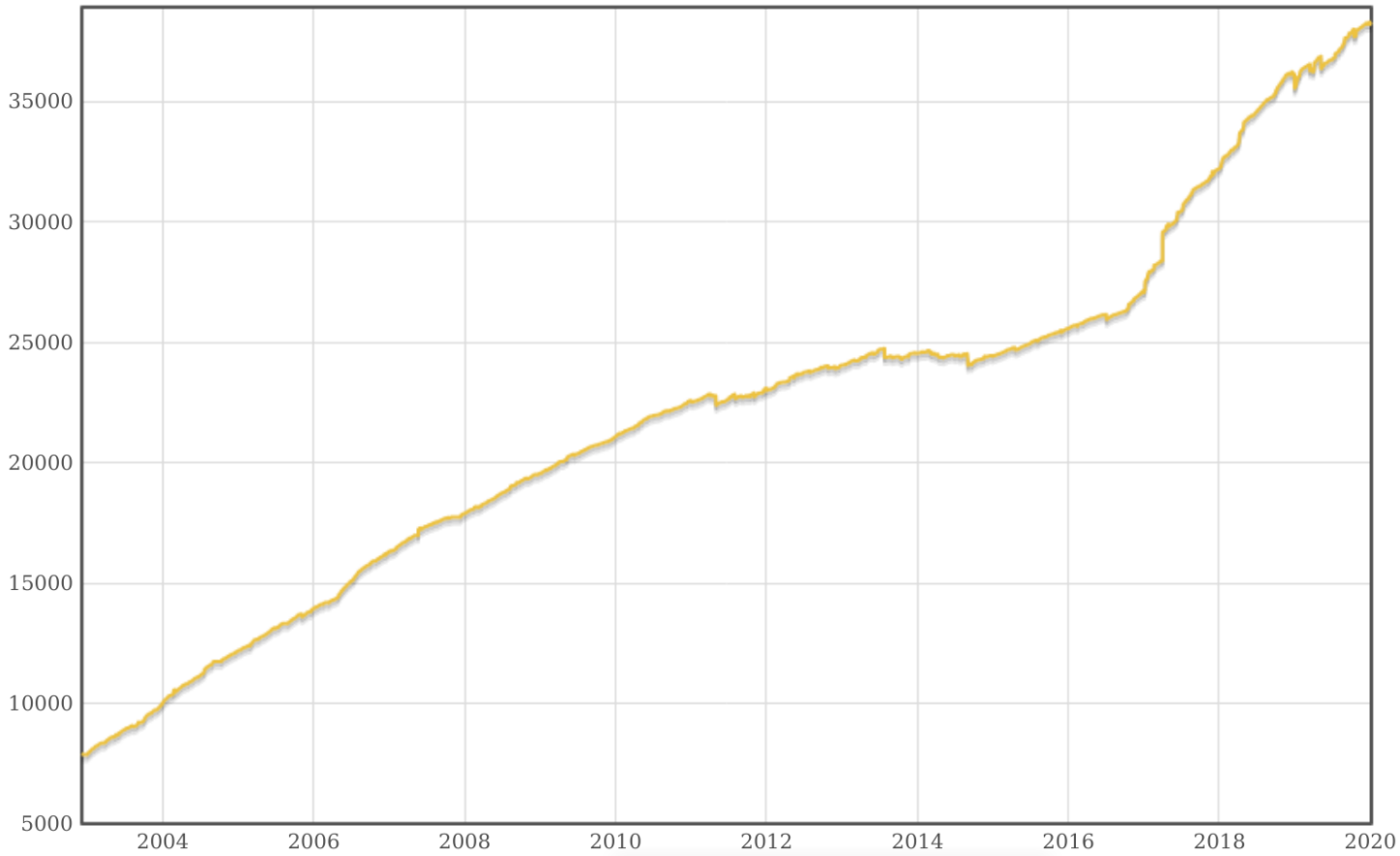
Category	Examples
CLI tools	`sh(1)`, `ls(1)`, `find(1)`, `echo(1)`, `sed(1)`, `awk(1)`, `grep(1)`
Text editors	`vi(1)`, `ee(1)`, `ed(1)`
System admin	`gpart(8)`, `zfs(8)`, `ps(1)`, `top(1)`, `mount(8)`
Network config	`ifconfig(8)`, `netstat(1)`, `route(8)`, `arp(8)`
Virtualisation / Containers	`bhyve(8)`, `jail(8)`
Remote access client and server	`ssh(1)`, `sshd(8)`
Download clients	`fetch(1)`, `openssl(1)`, `ftp(1)`, `nc(1)`
Developer tools	`cc(1)`, `make(1)`
…	

What's Available in FreeBSD Ports?

Third-party software… 38,290 different programs!

Category	Examples
Graphical desktop environments	`KDE, GNOME, XFCE`
Web browsers	`Chromium, Firefox`
Office tools	`LibreOffice, Evince, Xpdf`
Graphics tools	`GIMP, Inkscape`
Audio / Video tools	`VLC, MPlayer, Handbrake`
Chat	`Irssi, Quassel, Pidgin`
Web servers	`Apache, Nginx, PHP, Wordpress`
Mail servers	`Postfix, Exim, Dovecot, Rspamd`
Databases	`MySQL, PostgreSQL, SQLite`
Developer tools & languages	`Python, Ruby, Java, Perl, Git`
Configuration Management	`Ansible, Puppet, Salt, Chef`
…	

Number of ports... it keeps growing!



Graph courtesy of FreshPorts.org

How to use ports

Section 2

Don't use ports! *

(Directly)

* My personal opinion

Don't Use Ports For *

Installing programs

- Use packages for this
- Don't use a combination of ports and packages

Why

- Using ports directly does work in simple cases; many people still do this
- The big issue:
  - Output of a port build varies depending on other installed programs
  - This can result in bugs; build or runtime dependency errors
  - Hard for ports maintainers to find/fix these bugs (*"works for me"* ™)

* My personal opinion

Do Use Ports For *

Building packages

- Do so in a clean environment - use poudriere(8)
- Then use these packages to manage installed programs on your machines
- The FreeBSD project does this for you
- Only build your own packages if you need to customise the build options

* My personal opinion

Ports vs Packages

Ports

- Formulas for how to download, build and install programs
- Typically compile the programs from source (time consuming)
- Download the program from the upstream code repo / website

Packages

- Pre-compiled binaries (ready to install)
- Generated using the formulas in ports
- Downloaded from the configured package repo (not the upstream site)
- Available from official FreeBSD package repo's (updated every few days)
- You can also make your own package repo's using poudriere(8)

How to use packages - catalogue

It's easy!

```
# pkg update
Updating FreeBSD repository catalogue...
Fetching meta.txz: 100%    944 B   0.9kB/s    00:01
Fetching packagesite.txz: 100%    6 MiB  83.5kB/s    01:18
Processing entries: 100%
FreeBSD repository update completed. 31770 packages processed.
All repositories are up to date.

$ pkg search tmux
py27-libtmux-0.8.2             Library for interfacing with tmux
py27-tmuxp-1.5.4               Session manager for tmux
py37-libtmux-0.8.2             Library for interfacing with tmux
py37-tmuxp-1.5.4               Session manager for tmux
rubygem-tmuxinator-1.1.1       Manage complex tmux sessions easily
tmux-3.0a                      Terminal Multiplexer
tmux-mem-cpu-load-3.4.0_4      CPU, RAM, and load monitor for use with tmux
tmux23-2.3_1                   Terminal Multiplexer (old stable version 2.3)
```

```
# pkg install tmux
Updating FreeBSD repository catalogue...
FreeBSD repository is up to date.
All repositories are up to date.
Updating database digests format: 100%
The following 3 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
        tmux: 3.0a
        utf8proc: 2.4.0
        libevent: 2.1.11

Number of packages to be installed: 3

The process will require 4 MiB more space.
676 KiB to be downloaded.

Proceed with this action? [y/N]: y
[1/3] Fetching tmux-3.0a.txz: 100%  303 KiB 155.3kB/s    00:02
[2/3] Fetching utf8proc-2.4.0.txz: 100%   59 KiB  60.6kB/s    00:01
[3/3] Fetching libevent-2.1.11.txz: 100%  314 KiB 160.8kB/s    00:02
Checking integrity... done (0 conflicting)
[1/3] Installing utf8proc-2.4.0...
[1/3] Extracting utf8proc-2.4.0: 100%
[2/3] Installing libevent-2.1.11...
[2/3] Extracting libevent-2.1.11: 100%
[3/3] Installing tmux-3.0a...
[3/3] Extracting tmux-3.0a: 100%
```

How to use packages - upgrade

```
# pkg upgrade
Updating FreeBSD repository catalogue...
FreeBSD repository is up to date.
All repositories are up to date.
Checking for upgrades (2 candidates): 100%
Processing candidates (2 candidates): 100%
The following 1 package(s) will be affected (of 0 checked):

Installed packages to be UPGRADED:
        ca_root_nss: 3.48 -> 3.49

Number of packages to be upgraded: 1

290 KiB to be downloaded.

Proceed with this action? [y/N]: y
[1/1] Fetching ca_root_nss-3.49.txz: 100%  290 KiB 296.6kB/s    00:01
Checking integrity... done (0 conflicting)
[1/1] Upgrading ca_root_nss from 3.48 to 3.49...
[1/1] Extracting ca_root_nss-3.49: 100%
```

How to use packages - repo's

- A fresh FreeBSD install uses the "quarterly" package repo
- Daily ports development occurs in head (still quite stable)
- The quarterly ports branch is copied from head
  (at the beginning of each quarter)
- Only security updates and bug fixes are backported
  (from head to the quarterly branch)
- You get to choose which repo to use

Cross-reference:

Packages	Ports	Goal
`latest`	`head`	Cutting edge
`quarterly`	`branches/2020Q1`	Stable

How to use packages - repo's

Instructions for configuring package repo's are in **/etc/pkg/FreeBSD.conf**:

```
# $FreeBSD: releng/12.1/usr.sbin/pkg/FreeBSD.conf.quarterly 346780 2019-04-27 04:00:50Z cperciva $
#
# To disable this repository, instead of modifying or removing this file,
# create a /usr/local/etc/pkg/repos/FreeBSD.conf file:
#
#   mkdir -p /usr/local/etc/pkg/repos
#   echo "FreeBSD: { enabled: no }" > /usr/local/etc/pkg/repos/FreeBSD.conf
#

FreeBSD: {
  url: "pkg+http://pkg.FreeBSD.org/${ABI}/quarterly",
  mirror_type: "srv",
  signature_type: "fingerprints",
  fingerprints: "/usr/share/keys/pkg",
  enabled: yes
}
```

How to use packages - repo's

If you want to use the latest packages:

- Create the file **/usr/local/etc/pkg/repos/FreeBSD.conf**

```
FreeBSD: {
  url: "pkg+http://pkg.FreeBSD.org/${ABI}/latest",
}
```

- Then run the commands:

```
# pkg update
Updating FreeBSD repository catalogue...
pkg: Repository FreeBSD has a wrong packagesite, need to re-create database
Fetching meta.txz: 100%    944 B   0.9kB/s    00:01
Fetching packagesite.txz: 100%    6 MiB 923.8kB/s    00:07
Processing entries: 100%
FreeBSD repository update completed. 31587 packages processed.
All repositories are up to date.

# pkg upgrade
...
```

How to use ports (if you must)

Read the documentation in the FreeBSD Handbook Ports Chapter

Step 1 - install pkg(8) and ca_root_nss

- **pkg:** ports create packages and therefore depend on pkg
- **ca_root_nss:** verify secure connection for downloading the ports tree

```
# pkg install -y ca_root_nss
Bootstrapping pkg from pkg+http://pkg.FreeBSD.org/FreeBSD:12:amd64/quarterly, please wait...
Verifying signature with trusted certificate pkg.freebsd.org.2013102301... done
Installing pkg-1.12.0...
Extracting pkg-1.12.0: 100%
Updating FreeBSD repository catalogue...
FreeBSD repository is up to date.
All repositories are up to date.
Updating database digests format: 100%
Checking integrity... done (0 conflicting)
The following 1 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
        ca_root_nss: 3.48

Number of packages to be installed: 1
[1/1] Installing ca_root_nss-3.48...
[1/1] Extracting ca_root_nss-3.48: 100%
...
```

Step 2 - Download the Ports tree

First you should delete the ports tree installed with FreeBSD

- It does not include easy methods to update to the latest version
- It is a static copy of the tree at the time of this FreeBSD release

```
# rm -rf /usr/ports/*
# chown -R myuser /usr/ports /var/db/ports /var/db/portsnap
```

After that, there are 2 methods to download a fresh ports tree:

- portsnap(8)
  - First download is *much* faster
- svn(1) ← this is my recommendation
  - Allows you to choose which ports branch to use: head or quarterly
  - Provides version control functionality
    (track local changes against official tree, generate diffs, etc.)

Download the Ports tree - Portsnap

```
$ portsnap fetch
Looking up portsnap.FreeBSD.org mirrors... 6 mirrors found.
Fetching public key from your-org.portsnap.freebsd.org... done.
Fetching snapshot tag from your-org.portsnap.freebsd.org... done.
Fetching snapshot metadata... done.
Fetching snapshot generated at Fri Jan 10 08:02:17 AWST 2020:
6ab9c7ca58b774f975aa2d6d3407a2bc6292f878cfa8b9          84 MB 2198 kBps    39s
Extracting snapshot... done.
Verifying snapshot integrity... done.
Fetching snapshot tag from your-org.portsnap.freebsd.org... done.
Fetching snapshot metadata... done.
Updating from Fri Jan 10 08:02:17 AWST 2020 to Fri Jan 10 12:10:13 AWST 2020.
Fetching 5 metadata patches... done.
Applying metadata patches... done.
Fetching 0 metadata files... done.
Fetching 9 patches.
(9/9) 100.00%  done.
done.
Applying patches...
done.
Fetching 0 new ports or files... done.
```

Download the Ports tree - Portsnap

```
$ portsnap extract
/usr/ports/.arcconfig
/usr/ports/.gitattributes
/usr/ports/.gitauthors
/usr/ports/.gitignore
/usr/ports/.gitmessage
/usr/ports/CHANGES
...
/usr/ports/x11/yelp/
/usr/ports/x11/zenity/
Building new INDEX files... done.
```

Updating the Ports tree - Portsnap

```
$ portsnap fetch update
Looking up portsnap.FreeBSD.org mirrors... 6 mirrors found.
Fetching snapshot tag from your-org.portsnap.freebsd.org... done.
Fetching snapshot metadata... done.
Updating from Fri Jan 10 10:37:25 AWST 2020 to Fri Jan 10 11:08:28 AWST 2020.
Fetching 5 metadata patches... done.
Applying metadata patches... done.
Fetching 0 metadata files... done.
Fetching 7 patches.
(7/7) 100.00%  done.
done.
Applying patches...
done.
Fetching 0 new ports or files... done.
Removing old files and directories... done.
Extracting new files:
/usr/ports/databases/rubygem-gdbm/
/usr/ports/security/krb5-devel/
/usr/ports/shells/ksh93-devel/
/usr/ports/shells/ksh93/
/usr/ports/sysutils/cfengine-devel/
/usr/ports/sysutils/cfengine-masterfiles-devel/
/usr/ports/www/p5-Mojolicious/
Building new INDEX files... done.
```

Download the Ports tree - Subversion

2 compatible options:

- install **devel/subversion** from ports/packages (recursive)
- use **svnlite(1)** included in FreeBSD base - slightly less functionality

```
$ svnlite checkout https://svn.FreeBSD.org/ports/head /usr/ports
A    science
A    science/vipster
A    science/kalzium
A    science/step
...
A    Keywords/terminfo.ucl
A    Keywords/glib-schemas.ucl
A    README
A    .arcconfig
 U   .
Checked out revision 522553.
```

Updating the Ports tree - Subversion

```
$ cd /usr/ports
$ svnlite up
Updating '.':
A    graphics/blend2d
A    graphics/blend2d/files
A    devel/asmjit
A    devel/asmjit/files
D    net/ntpsec/files
U    security/signify/Makefile
U    security/signify/distinfo
U    security/signify/files/patch-Makefile
...
U    net/ntpsec/Makefile
U    net/ntpsec/distinfo
Updated to revision 522559.
```

1st Port Install: ports-mgmt/dialog4ports

Allows user to select build-time options when compiling other ports

```
$ cd /usr/ports/ports-mgmt/dialog4ports
$ make install clean
===>  License BSD2CLAUSE accepted by the user
===>   dialog4ports-0.1.6 depends on file: /usr/local/sbin/pkg - found
=> dialog4ports-0.1.6.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
=> Attempting to fetch https://files.etoilebsd.net/dialog4ports/dialog4ports-0.1.6.tar.gz
dialog4ports-0.1.6.tar.gz                          10 kB   20 MBps    00s
===> Fetching all distfiles required by dialog4ports-0.1.6 for building
===>  Extracting for dialog4ports-0.1.6
=> SHA256 Checksum OK for dialog4ports-0.1.6.tar.gz.
===>  Patching for dialog4ports-0.1.6
===>  Applying FreeBSD patches for dialog4ports-0.1.6
===>  Configuring for dialog4ports-0.1.6
===>  Building for dialog4ports-0.1.6
--- dialog4ports.o ---
--- mixedlist.o ---
--- dialog4ports.1.gz ---
--- dialog4ports.o ---
cc  -O2 -pipe  -fstack-protector-strong -fno-strict-aliasing  -Wall -pedantic -c dialog4ports.c -o dialog4ports.o
...
```

1st Install: ports-mgmt/dialog4ports

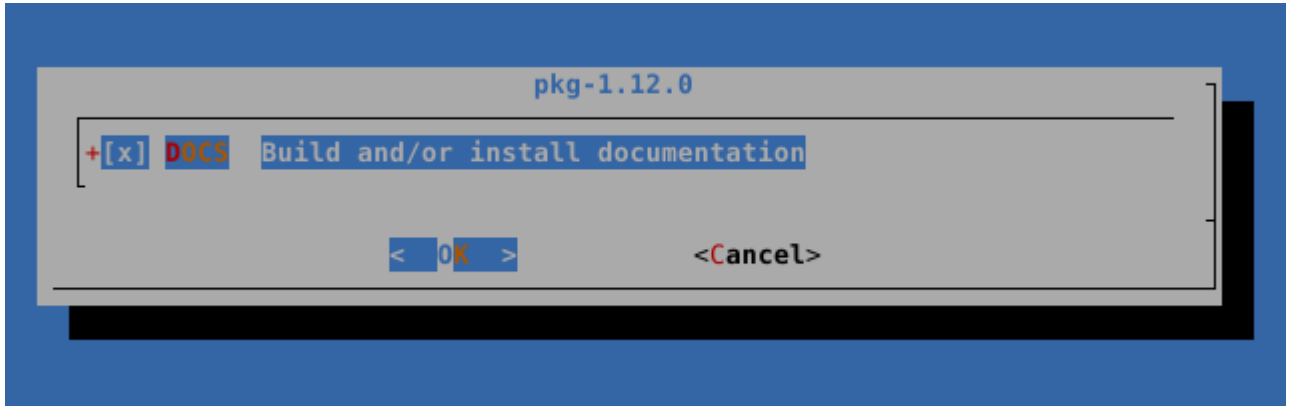
```
...
--- mixedlist.o ---
cc  -O2 -pipe  -fstack-protector-strong -fno-strict-aliasing  -Wall -pedantic -c mixedlist.c -o mixedlist.o
--- dialog4ports.1.gz ---
gzip -cn dialog4ports.1 > dialog4ports.1.gz
--- dialog4ports ---
cc -O2 -pipe  -fstack-protector-strong -fno-strict-aliasing  -Wall -pedantic dialog4ports.o mixedlist.o -o dialog4pc
===>  Staging for dialog4ports-0.1.6
===>   Generating temporary packing list
install  -s -m 555 dialog4ports /usr/ports/ports-mgmt/dialog4ports/work/stage/usr/local/bin
install  -m 0644 dialog4ports.1.gz /usr/ports/ports-mgmt/dialog4ports/work/stage/usr/local/man/man1
====> Compressing man pages (compress-man)
===>  Installing for dialog4ports-0.1.6
===>  Checking if dialog4ports is already installed
===>   Registering installation for dialog4ports-0.1.6
Installing dialog4ports-0.1.6...
===>  Cleaning for dialog4ports-0.1.6
```

2nd Port Install: ports-mgmt/pkg

Let's try building our own copy from source

```
$ cd /usr/ports/ports-mgmt/pkg
$ make install clean
```

This time we are presented with an options dialog before proceeding with the build:



2nd Port Install: ports-mgmt/pkg

```
===>  License BSD2CLAUSE accepted by the user
===> Fetching all distfiles required by pkg-1.12.0 for building
===>  Extracting for pkg-1.12.0
===>  License BSD2CLAUSE accepted by the user
===> Fetching all distfiles required by pkg-1.12.0 for building
=> SHA256 Checksum OK for freebsd-pkg-1.12.0_GH0.tar.gz.
===>  Patching for pkg-1.12.0
===>  Applying FreeBSD patches for pkg-1.12.0
===>  Configuring for pkg-1.12.0
...
===>  Installing for pkg-1.12.0
===>  Checking if pkg is already installed
===>   pkg-1.12.0 is already installed
      You may wish to ``make deinstall'' and install this port again
      by ``make reinstall'' to upgrade it properly.
      If you really wish to overwrite the old port of pkg
      without deleting it first, set the variable "FORCE_PKG_REGISTER"
      in your environment or the "make install" command line.
*** Error code 1

Stop.
...
```

2nd Port Install: ports-mgmt/pkg

```
$ make reinstall
===>  Deinstalling for pkg
===>   Deinstalling pkg-1.12.0
Updating database digests format: 100%
Checking integrity... done (0 conflicting)
Deinstallation has been requested for the following 1 packages (of 0 packages in the universe):

Installed packages to be REMOVED:
        pkg-1.12.0

Number of packages to be removed: 1

The operation will free 27 MiB.
[1/1] Deinstalling pkg-1.12.0...
[1/1] Deleting files for pkg-1.12.0: 100%
===>  Installing for pkg-1.12.0
===>  Checking if pkg is already installed
===>   Registering installation for pkg-1.12.0
Installing pkg-1.12.0...
...
```

Finding Ports

Can use FreshPorts.org:

The FreshPorts Search

Port Name | containing | lsof | 10 results

Number of ports: 2

lsof Lists information about open files (similar to fstat(1))
4.93.2_9,8 sysutils
IGNORE: requires kernel sources
Maintainer: ler@FreeBSD.org
Last Update: 2020-01-10 00:02:32
SVN Revision: 522545
License: lsof
SVNWeb : Homepage

Dependency lines:

- `lsof>0:sysutils/lsof`

To install the port: `cd /usr/ports/sysutils/lsof/ && make install clean`
A package is not available for ports marked as: Forbidden / Broken / Ignore / Restricted

PKGNAME: lsof

Flavors: there is no flavor information for this port.

---

p5-Unix-Lsof Unix::Lsof -- a wrapper to the Unix lsof utility
0.0.5_2 sysutils
BROKEN: unfetchable
IGNORE: is marked as broken: unfetchable
Maintainer: gjvc@gjvc.com
Last Update: 2019-11-06 16:26:35
SVN Revision: 516897
License: not specified in port
SVNWeb : Homepage

Dependency lines:

- `p5-Unix-Lsof>0:sysutils/p5-Unix-Lsof`

To install the port: `cd /usr/ports/sysutils/p5-Unix-Lsof/ && make install clean`
A package is not available for ports marked as: Forbidden / Broken / Ignore / Restricted

PKGNAME: p5-Unix-Lsof

Flavors: there is no flavor information for this port.

Number of ports: 2

Finding Ports

Can search the ports tree directly:

```
$ cd /usr/ports
$ make fetchindex
/usr/bin/env  fetch -am -o /usr/ports/INDEX-12.bz2 https://www.FreeBSD.org/ports/INDEX-12.bz2
/usr/ports/INDEX-12.bz        2328 kB  897 kBps    03s

$ make search name=lsof
Port:   lsof-4.93.2_8,8
Path:   /usr/ports/sysutils/lsof
Info:   Lists information about open files (similar to fstat(1))
Maint:  ler@FreeBSD.org
B-deps:
R-deps:
WWW:    https://people.freebsd.org/~abe/

Port:   p5-Unix-Lsof-0.0.5_2
Path:   /usr/ports/sysutils/p5-Unix-Lsof
Info:   Unix::Lsof -- a wrapper to the Unix lsof utility
Maint:  gjvc@gjvc.com
B-deps: p5-IPC-Run3-0.048_1 perl5-5.30.1
R-deps: p5-IPC-Run3-0.048_1 perl5-5.30.1
WWW:    https://metacpan.org/release/Unix-Lsof
```

Individual make steps

Step	Command	Default Target
Select build options	`make config[-recursive]`	↓
Port formula checks	`make check-sanity`	↓
Fetch	`make fetch[-recursive]`	↓
Checksum	`make checksum[-recursive]`	↓
Extract	`make extract`	↓
Patch	`make patch`	↓
Configure	`make configure`	↓
Build (compile)	`make build`	↓
Stage	`make stage`	↓
Package	`make package[-recursive]`	← `make`
Install	`make install`	
Clean Up WRKDIR	`make clean[-depends]`	
Uninstall	`make deinstall[-all]`	
Re-Install	`make reinstall`	
Reset build options	`make rmconfig[-recursive]`	

Ports tree internals - how it works

Section 3

A bunch of Makefiles

Providing standard macros for common tasks

What are Makefiles

- Instructions which direct the make(1) program how to compile, link and install a program
- A Makefile consists of "rules" in the following form:

```
target: dependencies
    system command(s)
```

- Target = name of an action to carry out, or name of a file to be generated
- Dependencies = list of other targets or files which are pre-requisites for this target
- System commands = this recipe, or list of commands that make(1) should run for this target

The make(1) command looks for a file by the name **Makefile** in the current directory to execute:

```
$ make [options] [target1 target2 ...]
```

Read more about Makefiles here: https://en.wikipedia.org/wiki/Makefile

Ports Framework - Powerful Makefiles

The magic is all in this one line in each port:

```
.include <bsd.port.mk>
```

This does all of the work:

- fetch
- checksum
- extract
- patch
- configure
- build
- stage
- package
- install
- ...

Benefits of this approach:

- Standardises the code for these steps
- Prevents duplication within every port

Individual ports simply:

- define variables
- define / over-ride make targets
- .include < bsd.port.mk >

make(1) searches the default system include path and finds the following matching Makefile:

- **/usr/share/mk/bsd.port.mk**

This Makefile determines the root directory of your ports tree and then includes

- **${PORTSDIR}/Mk/bsd.port.mk**

Ports Framework - Mk/* Makefiles

bsd.port.mk will include other Makefiles which make up the broader ports framework

- The variables defined in a port determine which other Makefiles to include
- e.g. USES=gnome → Mk/Uses/gnome.mk

This broader ports framework can be found in **/usr/ports/Mk/**

```
$ wc -l /usr/ports/Mk/*
      68 Mk/bsd.ccache.mk                28 Mk/bsd.local.mk                312 Mk/bsd.sanity.mk
     132 Mk/bsd.commands.mk             204 Mk/bsd.ocaml.mk               1338 Mk/bsd.sites.mk
     141 Mk/bsd.default-versions.mk      47 Mk/bsd.octave.mk                12 Mk/bsd.ssp.mk
     222 Mk/bsd.destdir.mk             549 Mk/bsd.options.desc.mk          656 Mk/bsd.tex.mk
     215 Mk/bsd.gcc.mk                 654 Mk/bsd.options.mk               518 Mk/bsd.wx.mk
     392 Mk/bsd.gecko.mk              5393 Mk/bsd.port.mk                    0 Mk/Scripts
     671 Mk/bsd.gstreamer.mk            23 Mk/bsd.port.options.mk            0 Mk/Uses
     475 Mk/bsd.java.mk                  7 Mk/bsd.port.post.mk               0 Mk/Wrappers
      99 Mk/bsd.ldap.mk                  7 Mk/bsd.port.pre.mk            14310 total
     424 Mk/bsd.licenses.db.mk         516 Mk/bsd.port.subdir.mk
     817 Mk/bsd.licenses.mk            390 Mk/bsd.ruby.mk
```

Ports Framework - Mk/Uses/* Makefiles

```
 59 7z.mk                  55 eigen.mk            76 grantlee.mk     21 metaport.mk       39 scons.mk
 38 ada.mk                126 elixir.mk           31 groff.mk       234 mono.mk          112 sdl.mk
 30 alias.mk              150 emacs.mk           188 gssapi.mk      38 motif.mk           22 shared-mime-info.mk
433 apache.mk             138 erlang.mk          159 horde.mk      177 mysql.mk         108 shebangfix.mk
 93 autoreconf.mk          17 fakeroot.mk         69 iconv.mk       93 ncurses.mk        24 sqlite.mk
 41 azurepy.mk             62 fam.mk              50 imake.mk       29 ninja.mk         129 ssl.mk
245 bdb.mk                 24 firebird.mk         38 jpeg.mk        82 objc.mk           36 tar.mk
 33 bison.mk               76 fonts.mk           856 kde.mk         30 openal.mk        234 tcl.mk
 52 blaslapack.mk          46 fortran.mk          83 kmod.mk        63 pathfix.mk        18 terminfo.mk
192 cabal.mk              206 fpc.mk             138 lazarus.mk    177 pear.mk          10 tk.mk
320 cargo.mk               26 fuse.mk             22 lha.mk        335 perl5.mk          31 uidfix.mk
 27 charsetfix.mk          136 gem.mk             16 libarchive.mk 180 pgsql.mk         221 uniquefiles.mk
143 cmake.mk               28 gettext-runtime.mk  16 libedit.mk    479 php.mk           30 varnish.mk
297 compiler.mk            25 gettext-tools.mk    84 libtool.mk     33 pkgconfig.mk      51 waf.mk
 42 corosync.mk            21 gettext.mk         267 linux.mk      301 pyqt.mk          178 webplugin.mk
 41 cpe.mk                 107 ghostscript.mk     31 localbase.mk   761 python.mk        111 xfce.mk
 77 cran.mk                 57 gl.mk              110 lua.mk         63 qmail.mk         188 xorg-cat.mk
 22 desktop-file-utils.mk   21 gmake.mk           107 lxqt.mk       152 qmake.mk         160 xorg.mk
 20 desthack.mk            524 gnome.mk            22 makeinfo.mk   387 qt-dist.mk        27 zip.mk
 37 display.mk              95 gnustep.mk          23 makeself.mk   397 qt.mk             95 zope.mk
 46 dos2unix.mk            180 go.mk              229 mate.mk        24 readline.mk    13168 total
117 drupal.mk               35 gperf.mk            71 meson.mk       42 samba.mk
```

Ports Categories

- All ports must be assigned to one or more categories
- The port files go in a subdirectory under the primary category

Physical categories (corresponding subdirectory in the ports tree):

```
accessibility   comms           french      java        net-mgmt    security        x11-drivers
arabic          converters      ftp         korean      net-p2p     shells          x11-fm
archivers       databases       games       lang        news        sysutils        x11-fonts
astro           deskutils       german      mail        polish      textproc        x11-servers
audio           devel           graphics    math        ports-mgmt  ukrainian       x11-themes
benchmarks      dns             hebrew      misc        portuguese  vietnamese      x11-toolkits
biology         editors         hungarian   multimedia  print       www             x11-wm
cad             emulators       irc         net         russian     x11
chinese         finance         japanese    net-im      science     x11-clocks
```

Virtual categories (no subdirectory):

```
afterstep       gnome           ipv6        lisp        perl5       rubygems        tk
docs            gnustep         kde         mbone       plan9       scheme          windowmaker
elisp           hamradio        kld         parallel    python      spanish         xfce
geography       haskell         linux       pear        ruby        tcl             zope
```

Category Makefiles

Each physical category has a Makefile in it's sub-directory to define the ports in that category

E.g. /usr/ports/arabic/Makefile

```
# $FreeBSD: head/arabic/Makefile 488807 2018-12-31 17:55:45Z rene $
#

    COMMENT = Arabic language support

    SUBDIR += ae_fonts_mono
    SUBDIR += ae_fonts_ttf
    SUBDIR += arabtex
    SUBDIR += aspell
    SUBDIR += kacst_fonts
    SUBDIR += khotot
    SUBDIR += libitl
    SUBDIR += libreoffice

.include <bsd.port.subdir.mk>
```

Ports development - some examples

Section 4

Read the Documentation

FreeBSD Porter's Handbook

Creating a new port

textproc/aha

Check the port doesn't already exist

- I've made this mistake on a few occasions!
- Search ports tree - can also use FreshPorts.org
    - Be careful of ports which might have a prefix in their origin
      e.g. Python ports begin with *py-*
      e.g. Perl ports begin with *p5-*
- Check if someone has already submitted the new port for consideration
    - Check BugZilla
    - Check Phabricator

Create the index entry

Decide which category the port belongs to.

- This program manipulates text, so I opted to use the **textproc** category

```
$ cd /usr/ports/textproc
$ vim Makefile
$ svn diff Makefile
Index: Makefile
===================================================================
--- Makefile    (revision 522140)
+++ Makefile    (working copy)
@@ -35,6 +35,7 @@
     SUBDIR += af-aspell
     SUBDIR += aft
     SUBDIR += agrep
+    SUBDIR += aha
     SUBDIR += aiksaurus
     SUBDIR += align
     SUBDIR += am-aspell
```

Create the port skeleton

```
$ mkdir /usr/ports/textproc/aha
$ cd /usr/ports/textproc/aha
$ vim Makefile
$ vim pkg-descr
```

Makefile

```
# $FreeBSD$

PORTNAME=       aha
PORTVERSION=    0.5
CATEGORIES=     textproc devel

MAINTAINER=     woodsb02@FreeBSD.org
COMMENT=        Ansi HTML Adapter

LICENSE=                LGPL20+ MPL11
LICENSE_COMB=           dual

USE_GITHUB=     yes
GH_ACCOUNT=     theZiz

.include <bsd.port.mk>
```

pkg-descr

```
Converts ANSI escape sequences of a unix terminal to HTML code.

WWW: https://github.com/theZiz/aha
```

distinfo & make makesum

We will auto-generate the contents of the **distinfo** file

```
$ make makesum
===>  License LGPL20+ MPL11 accepted by the user
===>  License LGPL20+ MPL11 accepted by the user
===>   aha-0.5 depends on file: /usr/local/sbin/pkg - found
=> theZiz-aha-0.5_GH0.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
=> Attempting to fetch https://codeload.github.com/theZiz/aha/tar.gz/0.5?dummy=/theZiz-aha-0.5_GH0.tar.gz
fetch: https://codeload.github.com/theZiz/aha/tar.gz/0.5?dummy=/theZiz-aha-0.5_GH0.tar.gz: size of remote file is no
theZiz-aha-0.5_GH0.tar.gz                      367 kB 1374 kBps    00s
===> Fetching all distfiles required by aha-0.5 for building

$ cat distinfo
TIMESTAMP = 1578635903
SHA256 (theZiz-aha-0.5_GH0.tar.gz) = 6f8b044bee9760a1a85dffbc362e532d7dd91bb20b7ed4f241ff1119ad74758f
SIZE (theZiz-aha-0.5_GH0.tar.gz) = 375851
```

Read the upstream build instructions

```
$ make extract
===>  License LGPL20+ MPL11 accepted by the user
===>   aha-0.5 depends on file: /usr/local/sbin/pkg - found
===> Fetching all distfiles required by aha-0.5 for building
===>  Extracting for aha-0.5
=> SHA256 Checksum OK for theZiz-aha-0.5_GH0.tar.gz.

$ cd work/aha-0.5/
$ ls
aha.1      aha.c      CHANGELOG      Makefile     README.md      screenshot.png
$ less README.md
```

```
Compilation / Installation
==========================

Aha has no dependencies except for a C compiler and `make`.

To compile just type `make`.
To install aha to `/usr/local/` type `make install`.
You can change the installation directory with `make install PREFIX=/your/path`.
You can override the man directory with `make install MANDIR=/your/path/man`
```

No configure stage, and no dependencies - unusual! But an easy example 😉

Reviewing the upstream Makefile

```
.PHONY: all clean install

PREFIX?=/usr/local

DATAROOTDIR?=$(PREFIX)/share
MANDIR?=$(DATAROOTDIR)/man

BINMODE?=0755
MANMODE?=644

CFLAGS += -Wall -Wextra

all: aha

aha: aha.c
	$(CC) -std=c99 $(CFLAGS) $(LDFLAGS) $(CPPFLAGS) aha.c -o aha

clean:
	rm -f aha

install: aha
	install -d $(DESTDIR)$(PREFIX)/bin
	install -m $(BINMODE) aha $(DESTDIR)$(PREFIX)/bin
	install -d $(DESTDIR)$(MANDIR)/man1
	install -m $(MANMODE) aha.1 $(DESTDIR)$(MANDIR)/man1
```

Looks like it should just work ™

portlint(1)

> “
>
> *portlint tries to verify the content of a port directory. The purpose of portlint can be separated into two parts:*
>
> *(1) to let the submitters easily polish their own port directory, and*
>
> *(2) to decrease the labor of > the committers.*
>
> ”

When you think you are ready to start testing your port, first run portlint(1)

```
$ portlint -AC
looks fine.
```

Test build

```
$ cd /usr/ports/textproc/aha
$ make clean stage
===>  Cleaning for aha-0.5
===>  License LGPL20+ MPL11 accepted by the user
===>   aha-0.5 depends on file: /usr/local/sbin/pkg - found
===> Fetching all distfiles required by aha-0.5 for building
===>  Extracting for aha-0.5
=> SHA256 Checksum OK for theZiz-aha-0.5_GH0.tar.gz.
===>  Patching for aha-0.5
===>  Configuring for aha-0.5
===>  Building for aha-0.5
--- aha ---
cc -std=c99 -O2 -pipe  -fstack-protector-strong -fno-strict-aliasing  -Wall -Wextra  -fstack-protector-strong   aha.
aha.c:1028:35: warning: unknown warning group '-Wimplicit-fallthrough=', ignored [-Wunknown-warning-option]
                       #pragma GCC diagnostic ignored "-Wimplicit-fallthrough="
                                                      ^
1 warning generated.
...
```

We can ignore this warning - difference in GCC vs LLVM compile warning syntax

Test build

The port make continues

```
...
===>  Staging for aha-0.5
===>   Generating temporary packing list
install -d /usr/ports/textproc/aha/work/stage/usr/local/bin
install -m 0755 aha /usr/ports/textproc/aha/work/stage/usr/local/bin
install -d /usr/ports/textproc/aha/work/stage/usr/local/share/man/man1
install -m 644 aha.1 /usr/ports/textproc/aha/work/stage/usr/local/share/man/man1
====> Compressing man pages (compress-man)
====> Running Q/A tests (stage-qa)
Warning: 'bin/aha' is not stripped consider trying INSTALL_TARGET=install-strip or using ${STRIP_CMD}
```

- At the end of the "stage" phase of a port build, the ports frameworks runs some quality checks
- We need to fix this strip warning.

Fix strip warning

- Recommended by `stage-qa` step
- The upstream Makefile does not include an install-strip target
- Add the following lines to the port Makefile to strip the compiled binaries

```
post-install:
        ${STRIP_CMD} ${STAGEDIR}${PREFIX}/bin/aha
```

Note: the **post-install** target is called after stage and before package or install (unintuitive)

You can check the values of these Makefile variables:

```
$ make -V STRIP_CMD
/usr/bin/strip

$ make -V STAGEDIR
/usr/ports/textproc/aha/work/stage

$ make -V PREFIX
/usr/local
```

pkg-plist & make makeplist

We will auto-generate the contents of the **pkg-plist** file

```
$ make makeplist
/you/have/to/check/what/makeplist/gives/you
bin/aha
share/man/man1/aha.1
```

Create pkg-plist:

- Copy the contents of the auto-generated plist file
  - Normally add this file list to the port **pkg-plist** file
  - If only a few files, instead set **PLIST_FILES** parameter in port Makefile
- Manually edit the auto-generated pkg-plist to correct any errors
  - e.g. files which only get installed if certain options are enabled
  - In this case we only need to delete the first line

```
PLIST_FILES=    bin/aha \
                share/man/man1/aha.1
```

portlint(1)

Another round of portlint to check pkg-plist

```
$ portlint -AC
FATAL: /usr/ports/textproc/aha/pkg-plist: [3]: \
       Man pages must be installed into ``man'' not ``share/man''.
FATAL: /usr/ports/textproc/aha/pkg-plist: [3]: \
       Unpacked man file aha.1 listed.  Must be gzipped.
```

Looks like we need to customise the installation of the manpage...

We will also need to change pkg-plist:

```
 GH_ACCOUNT=    theZiz

 PLIST_FILES=   bin/aha \
-               share/man/man1/aha.1
+               man/man1/aha.1.gz

 post-install:
        ${STRIP_CMD} ${STAGEDIR}${PREFIX}/bin/aha
```

Fixing man page location

Remember the note in the upstream build instructions:

```
You can override the man directory with `make install MANDIR=/your/path/man`
```

And the line in the upstream Makefile:

```
MANDIR?=$(DATAROOTDIR)/man
```

Adding the following line to our port Makefile will fix this:

```
MAKE_ARGS=      MANDIR="${MANPREFIX}/man"
```

You can check the value of MANPREFIX set by the Ports framework:

```
$ cd /usr/ports/textproc/aha
$ make -V MANPREFIX
/usr/local
```

Note that the ports framework auto-zips manpages if they are in the right location, so we don't have to worry about this step.

make clean install

```
===>  License LGPL20+ MPL11 accepted by the user
===>   aha-0.5 depends on file: /usr/local/sbin/pkg - found
===> Fetching all distfiles required by aha-0.5 for building
===>  Extracting for aha-0.5
=> SHA256 Checksum OK for theZiz-aha-0.5_GH0.tar.gz.
===>  Patching for aha-0.5
===>  Configuring for aha-0.5
===>  Building for aha-0.5
--- aha ---
cc -std=c99 -O2 -pipe  -fstack-protector-strong -fno-strict-aliasing  -Wall -Wextra  -fstack-protector-strong   aha.
aha.c:1028:35: warning: unknown warning group '-Wimplicit-fallthrough=', ignored [-Wunknown-warning-option]
                       #pragma GCC diagnostic ignored "-Wimplicit-fallthrough="
                                                      ^
1 warning generated.
...
```

Same warning as before - safe to ignore...

make clean install

The port make continues

```
...
===>  Staging for aha-0.5
===>   Generating temporary packing list
install -d /usr/ports/textproc/aha/work/stage/usr/local/bin
install -m 0755 aha /usr/ports/textproc/aha/work/stage/usr/local/bin
install -d /usr/ports/textproc/aha/work/stage/usr/local/man/man1
install -m 644 aha.1 /usr/ports/textproc/aha/work/stage/usr/local/man/man1
/usr/bin/strip /usr/ports/textproc/aha/work/stage/usr/local/bin/aha
====> Compressing man pages (compress-man)
====> Running Q/A tests (stage-qa)
===>  Installing for aha-0.5
===>  Checking if aha is already installed
===>   Registering installation for aha-0.5
Installing aha-0.5...
```

It worked!

make deinstall clean

```
===>  Deinstalling for aha
===>   Deinstalling aha-0.5
Updating database digests format: 100%
Checking integrity... done (0 conflicting)
Deinstallation has been requested for the following 1 packages (of 0 packages in the universe):

Installed packages to be REMOVED:
        aha-0.5

Number of packages to be removed: 1
[1/1] Deinstalling aha-0.5...
[1/1] Deleting files for aha-0.5: 100%
===>  Cleaning for aha-0.5
```

Makefile (final)

```
# $FreeBSD$

PORTNAME=       aha
PORTVERSION=    0.5
CATEGORIES=     textproc devel

MAINTAINER=     woodsb02@FreeBSD.org
COMMENT=        Ansi HTML Adapter

LICENSE=        LGPL20+ MPL11
LICENSE_COMB=   dual

USE_GITHUB=     yes
GH_ACCOUNT=     theZiz

MAKE_ARGS=      MANDIR="${MANPREFIX}/man"
PLIST_FILES=    bin/aha \
                man/man1/aha.1.gz

post-install:
        ${STRIP_CMD} ${STAGEDIR}${PREFIX}/bin/aha

.include <bsd.port.mk>
```

svn add

Mark the new port files to be under revision control

```
$ cd /usr/ports/textproc/aha
$ make clean
===>  Cleaning for aha-0.5

$ cd ..
$ svn add aha
A         aha
A         aha/Makefile
A         aha/distinfo
A         aha/pkg-descr

$ cd ..
$ svn status
M       textproc/Makefile
A       textproc/aha
A       textproc/aha/Makefile
A       textproc/aha/distinfo
A       textproc/aha/pkg-descr
```

Submit the new port

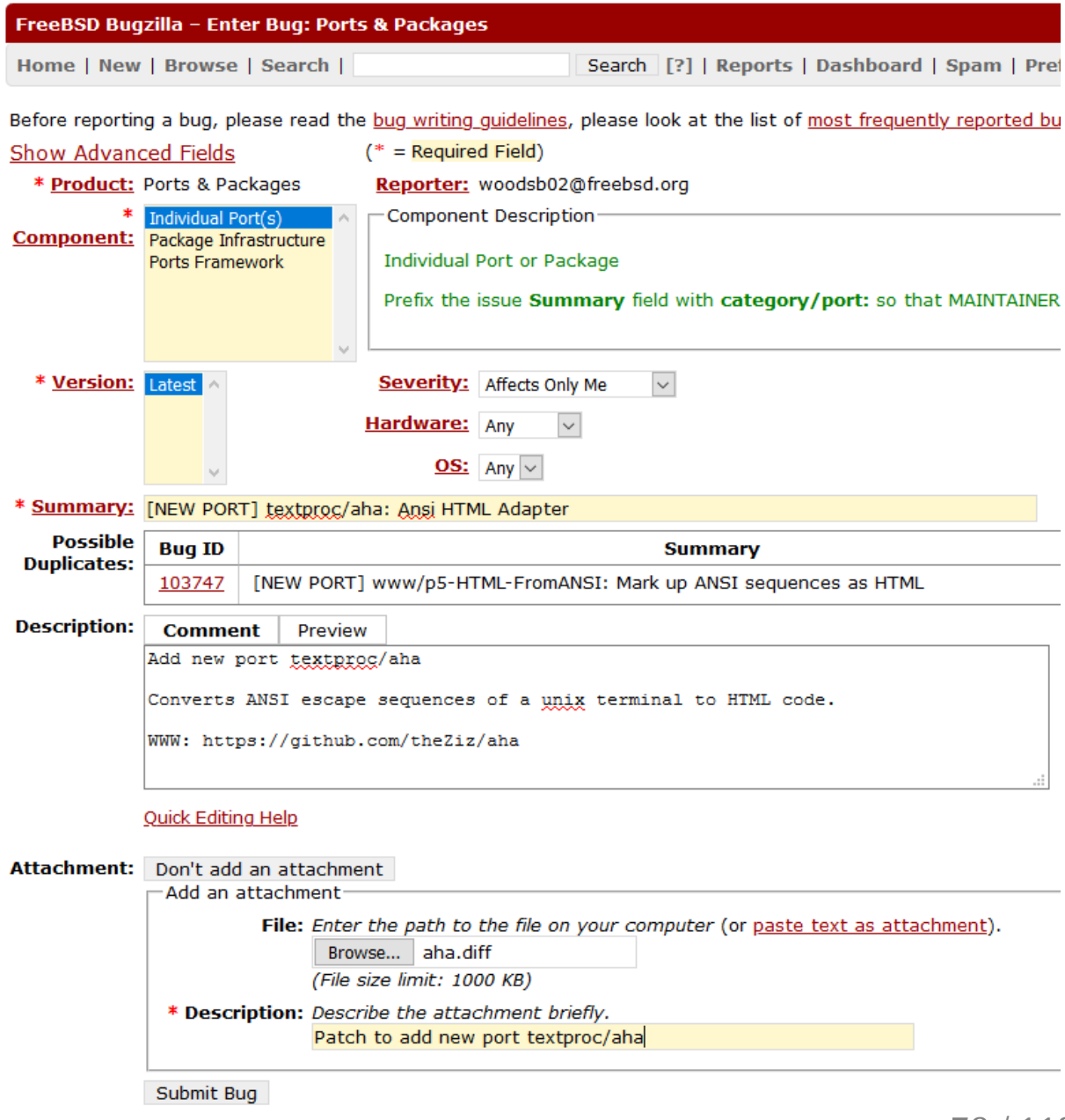
File a bug report on Bugzilla

Generate a diff and attach it to the bug report

Diff should be from the base of the ports tree

Be sure the diff includes all changes
(including the category Makefile)

```
$ cd /usr/ports
$ svn diff textproc > ~/aha.diff
```



Commit the new port - ports committer

Only ports committers can do this *

```
$ svn commit textproc/Makefile textproc/aha
Sending        textproc/Makefile
Adding         textproc/aha
Adding         textproc/aha/Makefile
Adding         textproc/aha/distinfo
Adding         textproc/aha/pkg-descr
Transmitting file data ....done
Committing transaction...
Committed revision 522747.
```

* Or a committer for one of the other FreeBSD repo's (src, docs) with approval from a ports committer

Commit the new port - ports committer

Add a commit message

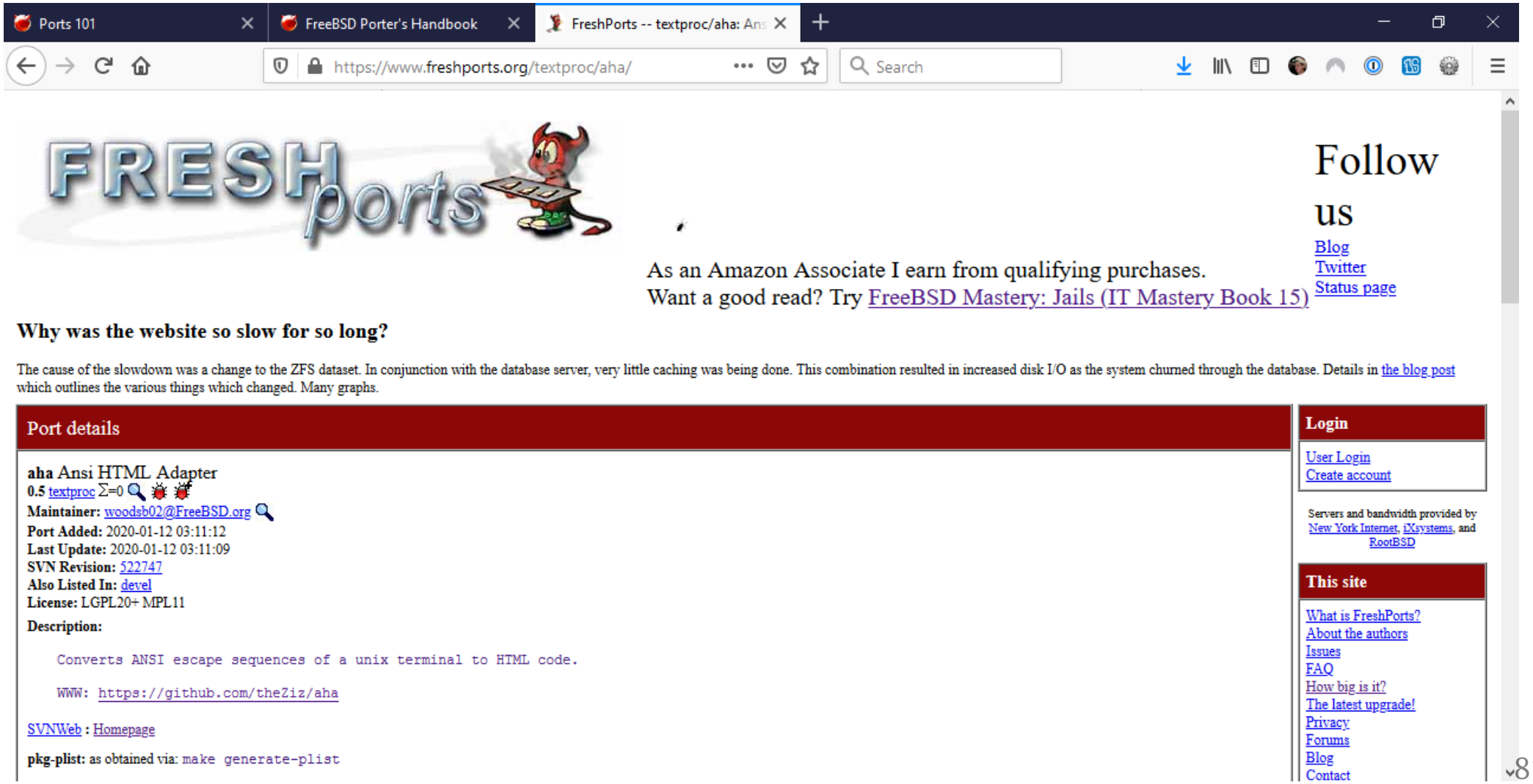
```
Add new port textproc/aha

Converts ANSI escape sequences of a unix terminal to HTML code.

WWW: https://github.com/theZiz/aha
--This line, and those below, will be ignored--
> Description of fields to fill in above:                    76 columns --|
> PR:                       If and which Problem Report is related.
> Submitted by:             If someone else sent in the change.
> Reported by:              If someone else reported the issue.
> Reviewed by:              If someone else reviewed your modification.
> Approved by:              If you needed approval for this commit.
> ...
> Empty fields above will be automatically removed.

M    textproc/Makefile
A    textproc/aha
AM   textproc/aha/Makefile
AM   textproc/aha/distinfo
AM   textproc/aha/pkg-descr
```

Behold - your new port on FreshPorts.org



Updating an existing port

print/py-weasyprint

Update PORTVERSION

Browse the upstream repo to determine the latest *stable* version

- https://weasyprint.org/news/
- https://github.com/Kozea/WeasyPrint/releases

```
$ cd /usr/ports/print/py-weasyprint
$ vim Makefile
$ svn diff Makefile
Index: Makefile
===================================================================
--- Makefile    (revision 522140)
+++ Makefile    (working copy)
@@ -2,7 +2,7 @@
 # $FreeBSD$

 PORTNAME=      weasyprint
-PORTVERSION=   47
+PORTVERSION=   51
 CATEGORIES=    print python
 MASTER_SITES=  CHEESESHOP
 PKGNAMEPREFIX= ${PYTHON_PKGNAMEPREFIX}
```

make makesum

Download the updated sources and record their size and checksum

```
$ make makesum
===>  License BSD3CLAUSE accepted by the user
===>  License BSD3CLAUSE accepted by the user
===>   py37-weasyprint-51 depends on file: /usr/local/sbin/pkg - found
=> WeasyPrint-51.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
=> Attempting to fetch https://files.pythonhosted.org/packages/source/W/WeasyPrint/WeasyPrint-51.tar.gz
WeasyPrint-51.tar.gz                               301 kB 7173 kBps    00s
===> Fetching all distfiles required by py37-weasyprint-51 for building

$ svn diff distinfo
Index: distinfo
===================================================================
--- distinfo    (revision 522140)
+++ distinfo    (working copy)
@@ -1,3 +1,3 @@
-TIMESTAMP = 1558358103
-SHA256 (WeasyPrint-47.tar.gz) = af8d39f45027ecd7fa47272b93558d851a49b3dad238a52478475a3733ffa141
-SIZE (WeasyPrint-47.tar.gz) = 300340
+TIMESTAMP = 1578804473
+SHA256 (WeasyPrint-51.tar.gz) = b3e971973a4f03c1430e6b838b75b5b57630415fcae8666d2be1347630ff6d6a
+SIZE (WeasyPrint-51.tar.gz) = 308411
```

Read changelogs / release notes for other changes

- Check for changes to the compile process
  - e.g. move from GNU autotools to CMake
  - None noted in this case
- Check for changes to dependencies
  - Some differences noted in this case
    https://weasyprint.readthedocs.io/en/latest/install.html

Modify Makefile to reflect build / dependency changes

```
--- print/py-weasyprint/Makefile    (revision 522140)
+++ print/py-weasyprint/Makefile    (working copy)
@@ -14,15 +14,18 @@
 LICENSE=    BSD3CLAUSE

 BUILD_DEPENDS=    ${PYTHON_PKGNAMEPREFIX}pytest-runner>=0.1:devel/py-pytest-runner@${PY_FLAVOR}
-RUN_DEPENDS=    ${PYTHON_PKGNAMEPREFIX}cssselect>=0.1:www/py-cssselect@${PY_FLAVOR} \
+RUN_DEPENDS=    ${PYTHON_PKGNAMEPREFIX}cairo>=1.15.4:graphics/py-cairo@${PY_FLAVOR} \
+        ${PYTHON_PKGNAMEPREFIX}cairocffi>=0.9.0:graphics/py-cairocffi@${PY_FLAVOR} \
+        ${PYTHON_PKGNAMEPREFIX}cairosvg>=2.4.0:graphics/py-cairosvg@${PY_FLAVOR} \
+        ${PYTHON_PKGNAMEPREFIX}cffi>=0.6:devel/py-cffi@${PY_FLAVOR} \
+        ${PYTHON_PKGNAMEPREFIX}cssselect2>=0.1:textproc/py-cssselect2@${PY_FLAVOR} \
         ${PYTHON_PKGNAMEPREFIX}html5lib>=0.999999999:www/py-html5lib@${PY_FLAVOR} \
-        ${PYTHON_PKGNAMEPREFIX}pdfrw>=0.4:textproc/py-pdfrw@${PY_FLAVOR} \
-        ${PYTHON_PKGNAMEPREFIX}pyphen>=0.8:textproc/py-pyphen@${PY_FLAVOR} \
+        ${PYTHON_PKGNAMEPREFIX}pyphen>=0.9.1:textproc/py-pyphen@${PY_FLAVOR} \
-        ${PYTHON_PKGNAMEPREFIX}tinycss>=0.4:textproc/py-tinycss@${PY_FLAVOR}
+        ${PYTHON_PKGNAMEPREFIX}tinycss2>=1.0.0:textproc/py-tinycss2@${PY_FLAVOR}

 USES=        gnome python:3.6+
 USE_PYTHON=    distutils autoplist
-USE_GNOME=    pango
+USE_GNOME=    gdkpixbuf2 pango

 NO_ARCH=    yes
```

Test build and check for pkg-plist changes

```
$ make install
...
=========================================================================
====> Running Q/A tests (stage-qa)
Warning: You have disabled the licenses framework with DISABLE_LICENSES, unable to run checks
====> Checking for pkg-plist issues (check-plist)
===> Parsing plist
===> Checking for items in STAGEDIR missing from pkg-plist
===> Checking for items in pkg-plist which are not in STAGEDIR
===> No pkg-plist issues found (check-plist)
=>> Checking for staging violations... done
=======================<phase: package        >============================
===>  Building package for py37-weasyprint-51
=========================================================================
=>> Recording filesystem state for preinst... done
=======================<phase: install        >============================
===>  Installing for py37-weasyprint-51
===>  Checking if py37-weasyprint is already installed
===>   Registering installation for py37-weasyprint-51
[my.fqdn.com] Installing py37-weasyprint-51...
```

Looks good - no changes to pkg-plist required (expected, python ports auto-generated their plist)

Submit the port update

File a bug report on Bugzilla

Generate a diff and attach it to the bug report

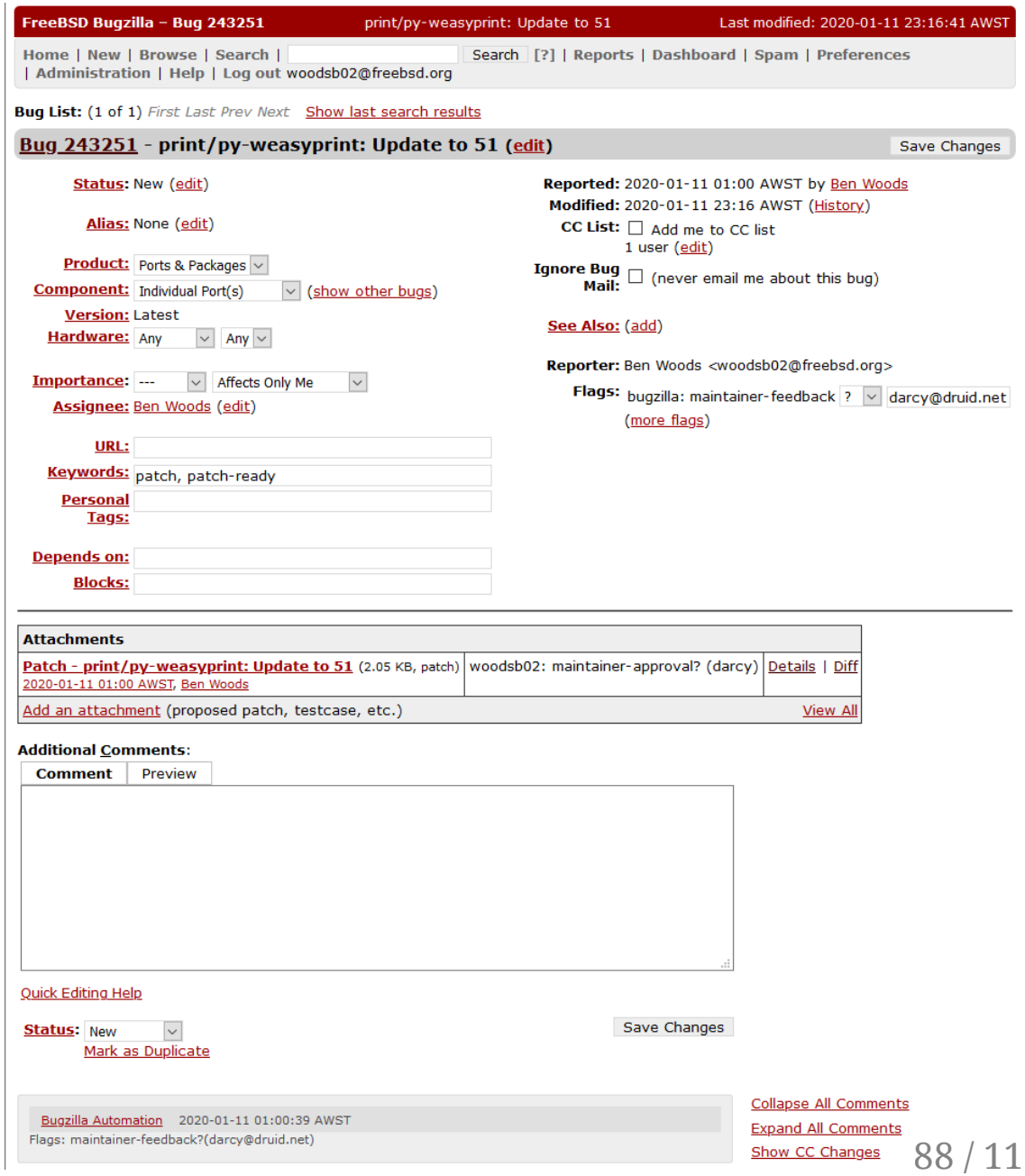
Diff should be from the base of the ports tree

Be sure the diff includes all changes
(including the category Makefile)

- Mark the patch as "maintainer-approval?"

```
$ cd /usr/ports
$ svn status print
M       print/py-weasyprint/Makefile
M       print/py-weasyprint/distinfo

$ svn diff print > ~/weasyprint.diff
```



Commit the port update - ports committer

Only ports committers can do this *

Must await maintainer approval (maintainer-timeout = 2 weeks)

```
$ fetch -o weasyprint.diff "https://bz-attachments.freebsd.org/attachment.cgi?id=210598&action=diff&format=raw&heade
$ svn patch weasyprint.diff
$ rm weasyprint.diff
$ svn status print
M       print/py-weasyprint/Makefile
M       print/py-weasyprint/distinfo

$ svn diff print | less
$ svn commit print/py-weasyprint
Sending        print/py-weasyprint/Makefile
Sending        print/py-weasyprint/distinfo
Transmitting file data ..done
Committing transaction...
Committed revision 522774.
```

* Or a committer for one of the other FreeBSD repo's (src, docs) with approval from a ports committer

Commit the port update - ports committer

Add a commit message

```
print/py-weasyprint: Update to 51

Changes this release:
  https://weasyprint.readthedocs.io/en/latest/changelog.html#version-51

PR:             243251
Approved by:    D'Arcy J.M. Cain <darcy@druid.net> (maintainer)
--This line, and those below, will be ignored--
> Description of fields to fill in above:                    76 columns --|
> PR:                       If and which Problem Report is related.
> Submitted by:             If someone else sent in the change.
> Reported by:              If someone else reported the issue.
> Reviewed by:              If someone else reviewed your modification.
> Approved by:              If you needed approval for this commit.
> ...
> Empty fields above will be automatically removed.

M    print/py-weasyprint/Makefile
M    print/py-weasyprint/distinfo
```

Commit the port update - ports committer

Close-out steps:

- If this is a security update or serious bug fix, you should MFH
  (merge the change to the ports quarterly branch)
    - Must first be approved by either ports-secteam or portmgr
- Then close the bug report
    - Also update other bug meta-data such as maintainer feedback/approval flags

How you can contribute

Section 5

Submit patches

Use BugZilla

What needs work?

- Broken ports
  - https://www.freshports.org/ports-broken.php
- Outdated ports
  - https://portscout.freebsd.org/ ← Tick the "with out-of-date only" option
- No maintainer ← adopt a port!
  - https://portscout.freebsd.org/ports@freebsd.org.html
- Submit new ports
  - Is your favourite program missing from ports?

Becoming a ports committer

Section 6

Submit *more* patches

Eventually you will get recognised

(Or become too much of a burden, and you will be punished with a commit bit)

No hard rules, but good guides:

- Quality over quantity
- That said, it helps if you have submitted 100 bug requests with ports patches

The process for new ports committers

1. Existing ports committer nominates someone for a commit bit
2. portmgr vote during their regular meetings
3. If outcome is positive, the commit bit will be granted:
    - 2-3 mentors will be allocated
      Hopefully in different time zones - fast response!
    - A FreeBSD.org account will be created
      (includings email address, ssh access)
    - Commit privileges will be granted to the ports tree
      Username and mentors will be added in svn to ports/svnadmin/conf/access
    - Added to the official FreeBSD Developers list
      https://www.freebsd.org/doc/en_US.ISO8859-1/articles/contributors/staff-committers.html
4. All commits must be approved by a mentor, until they decide you are ready to operate independently

My experience

- First bug report: 2013-07-03 - net/samba4 rc script syntax errors
- First bug report with a patch: 2014-06-26 - [NEW PORT] multimedia/plexhometheater
- First proposed for a commit bit: 2015-04-14 by adamw
  - Result: Denied 2015-04-20

> “
>
> *However, portmgr really wants to encourage Ben to follow his current path and keep on accumulating knowledge, especially by adopting some more ports and keeping on submitting good PRs. If Ben continues on the track he is on, portmgr is confident he will earn a commit bit in a near future.*
>
> *Again, thanks Adam for sponsoring Ben and portmgr really hopes to hear about him again soon.*
>
> ”

My experience

- Second proposal for a commit bit: 2016-04-23 by adamw
- Granted: 2016-05-09 in an email from Frederic Culot culot@freebsd.org

> *“*
>
> *Hi Ben, Adam, Kubilay and Mathieu*
>
> *I'm pleased to let you know that we would be happy to have Ben join the FreeBSD developer community as a ports committer. We're also confident that Adam, Kubilay and Mathieu will make fine mentors.*
>
> *...*
>
> *Adam, Kubilay and Mathieu will follow up with more instructions after the account has been created. Welcome aboard!*
>
> *Frederic*
>
> *on behalf of portmgr@*
>
> *”*

My experience

- Granted ports access by adamw in r414872

> “
>
> *Please welcome Ben Woods to our noble ranks. He submitted far too many PRs and, as penance, must now commit them himself. I will be mentoring him along with koobs and mat.*
>
> *Approved by: portmgr (implicit)*
>
> ”

- Released from mentorship: 2016-12-27 by adamw in r429654

Thank you *very much* to my mentors

You know who you are *

* For clarity: Adam Weinberger (adamw), Mathieu Arnold (mat), Kubilay Kocak (koobs) ← Legends!

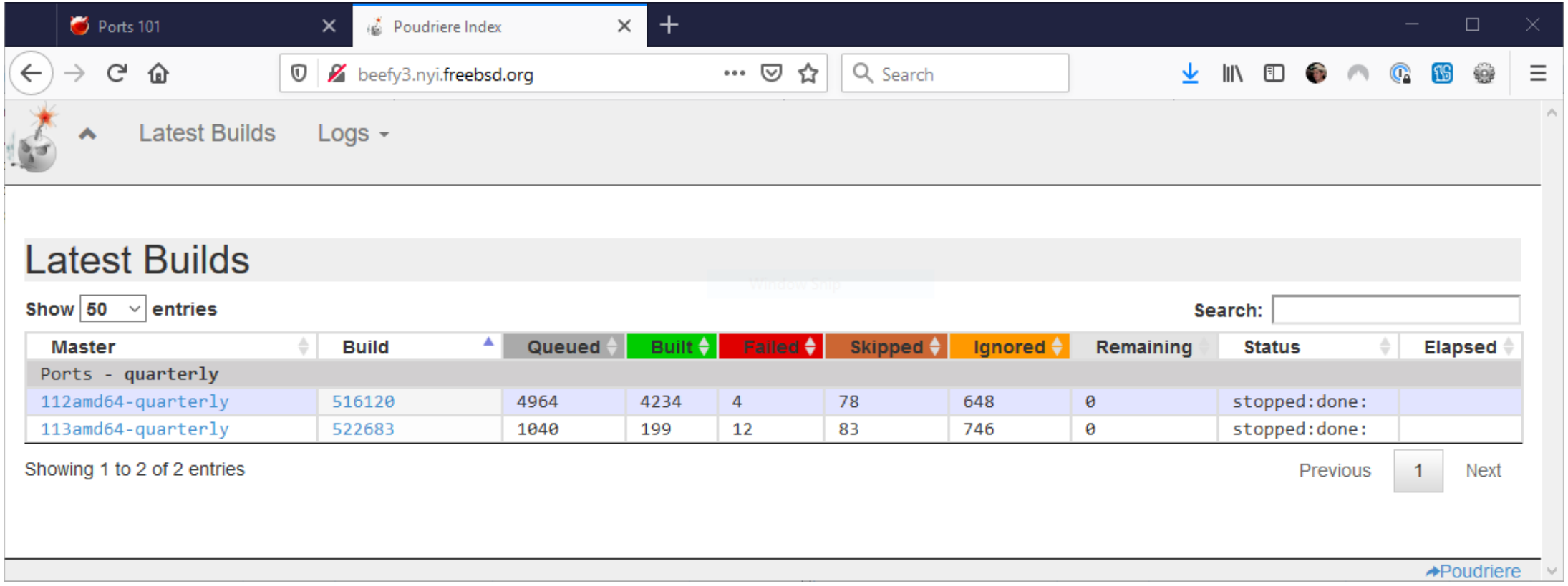
Poudriere(8)

"poo-dree-year" - french word for powder keg

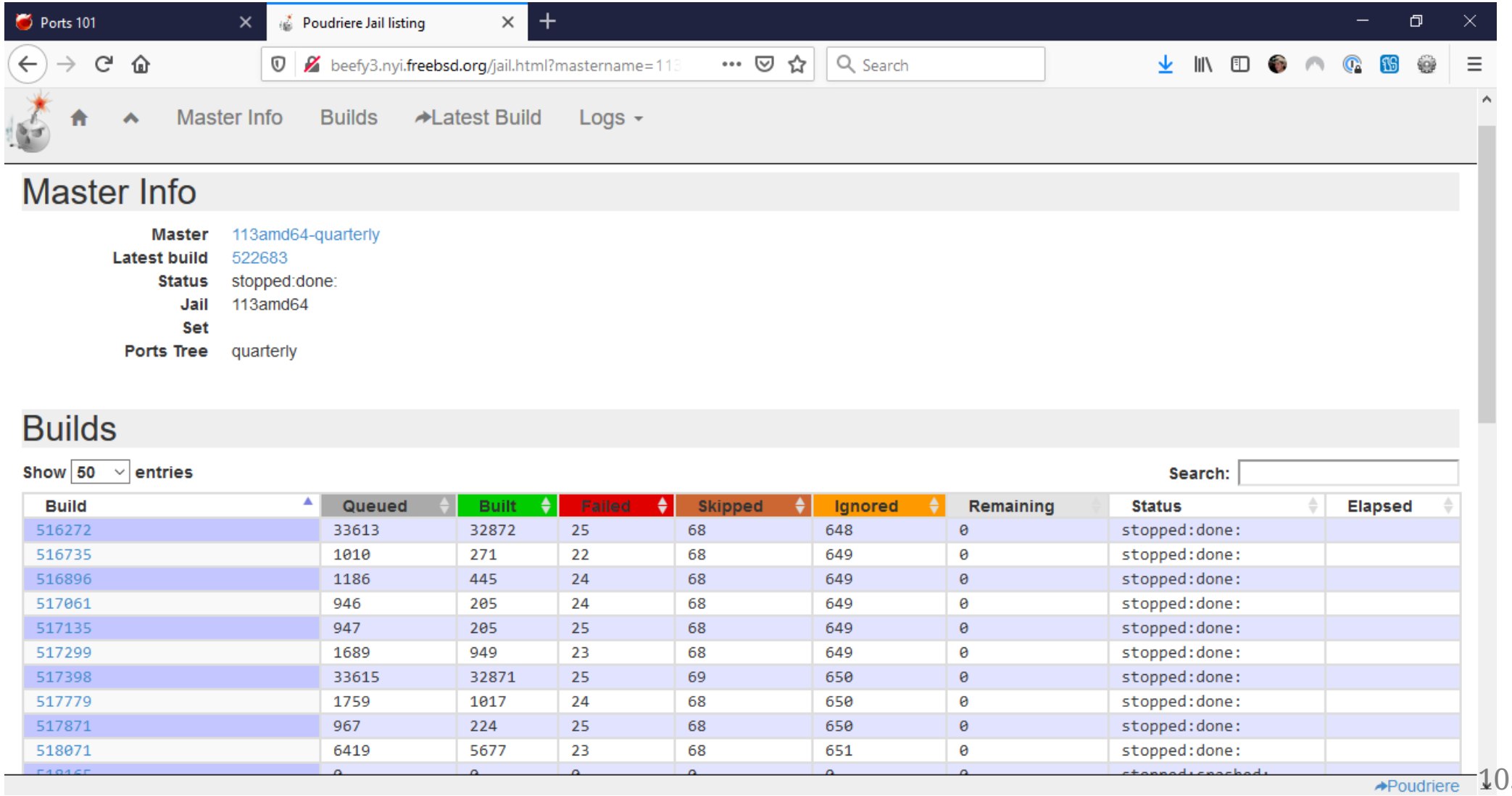
What is poudriere(8)

- Tool to bulk build ports for FreeBSD using ZFS & jails - parallel, clean env, limit networking post-fetch
- # parallel port builds = # cpu threads (individual port builds are not parallel; make -j1)
- Creates a package repository - you can point your pkg(8) tool at it
- Also helpful for testing ports
- Only depends on the tools in FreeBSD base - it's mainly shell scripts
- Can build packages for different FreeBSD versions than the host
    - Can build FreeBSD 11 packages on a FreeBSD 12 host
    - Can build i386 packages from an amd64 host
- Can push your hardware to it's limits - an accidental performance & scheduler test suite!
    - CPU utilisation
    - Disk I/O
- Provides a web interface to show status of builds (optional)

Poudriere Web - Summary Page

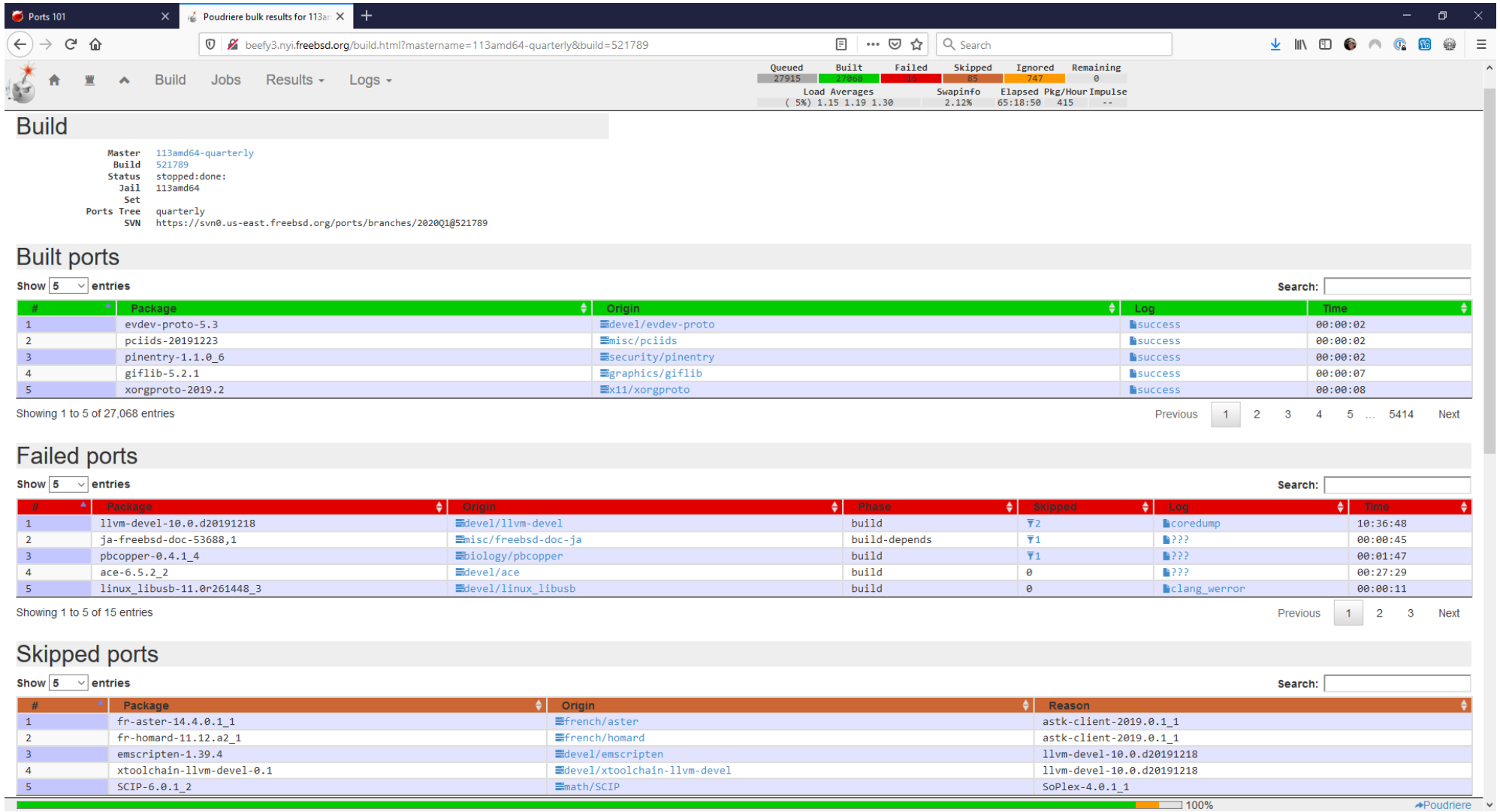


Poudriere Web - Build List



Build	Queued	Built	Failed	Skipped	Ignored	Remaining	Status	Elapsed
516272	33613	32872	25	68	648	0	stopped:done:	
516735	1010	271	22	68	649	0	stopped:done:	
516896	1186	445	24	68	649	0	stopped:done:	
517061	946	205	24	68	649	0	stopped:done:	
517135	947	205	25	68	649	0	stopped:done:	
517299	1689	949	23	68	649	0	stopped:done:	
517398	33615	32871	25	69	650	0	stopped:done:	
517779	1759	1017	24	68	650	0	stopped:done:	
517871	967	224	25	68	650	0	stopped:done:	
518071	6419	5677	23	68	651	0	stopped:done:	

Poudriere Web - Build Details



Setup poudriere(8)

```
# pkg install poudriere
# vim /usr/local/etc/poudriere.conf
# zfs create -o compression=lz4 -o atime=off \
  -o mountpoint=/usr/local/poudriere zroot/poudriere
# zfs create zroot/poudriere/jails
# zfs create zroot/poudriere/ports
# mkdir /usr/ports/distfiles
# poudriere jail -c -j 12amd64 -v 12.1-RELEASE
# poudriere ports -c -m null -M /usr/ports
# vim /usr/local/etc/poudriere.d/port-list
# mkdir -p /usr/local/etc/ssl/keys
# mkdir -p /usr/local/etc/ssl/certs
# openssl genrsa -out \
  /usr/local/etc/ssl/keys/poudriere.key 4096
# chmod 0600 /usr/local/etc/ssl/keys/poudriere.key
# openssl rsa -in \
  /usr/local/etc/ssl/keys/poudriere.key -pubout \
  -out /usr/local/etc/ssl/certs/poudriere.cert
# poudriere bulk -j 12amd64 -f \
  /usr/local/etc/poudriere.d/port-list
```

/usr/local/etc/poudriere.conf

```
ZPOOL=zroot
ZROOTFS=/poudriere
FREEBSD_HOST=https://download.FreeBSD.org
RESOLV_CONF=/etc/resolv.conf
BASEFS=/usr/local/poudriere
USE_PORTLINT=yes
USE_TMPFS=yes
DISTFILES_CACHE=/usr/ports/distfiles
PKG_REPO_SIGNING_KEY=/usr/local/etc/ssl/keys/poudriere
PARALLEL_JOBS=8
ALLOW_MAKE_JOBS_PACKAGES="*webkit* libreoffice*"
URL_BASE=https://yourdomain.com/poudriere/
BUILDER_HOSTNAME=pkg.yourdomain.com
```

/usr/local/etc/poudriere.d/port-list

```
audio/lame
multimedia/ffmpeg
```

Poudriere Console Output - Bulk

```
root@shiny-new-toy:/usr/home/woodsb02 # poudriere bulk -j 12amd64 -f /usr/local/etc/poudriere.d/port-list
[00:00:00] Creating the reference jail... done
[00:00:00] Mounting system devices for 12amd64-default
[00:00:00] Mounting ports/packages/distfiles
[00:00:00] Using packages from previously failed build: /usr/local/poudriere/data/packages/12amd64-default/.building
[00:00:00] Mounting packages from: /usr/local/poudriere/data/packages/12amd64-default
/etc/resolv.conf -> /usr/local/poudriere/data/.m/12amd64-default/ref/etc/resolv.conf
[00:00:00] Starting jail 12amd64-default
[00:00:01] Logs: /usr/local/poudriere/data/logs/bulk/12amd64-default/2020-01-13_23h10m18s
[00:00:01] WWW: https://yourdomain.com/poudriere//build.html?mastername=12amd64-default&build=2020-01-13_23h10m18s
[00:00:01] Loading MOVED for /usr/local/poudriere/data/.m/12amd64-default/ref/usr/ports
[00:00:02] Ports supports: FLAVORS SELECTED_OPTIONS
[00:00:02] Gathering ports metadata
[00:01:03] Calculating ports order and dependencies
[00:01:03] pkg package missing, skipping sanity
[00:01:03] Skipping incremental rebuild and repository sanity checks
[00:01:03] Cleaning the build queue
[00:01:03] Sanity checking build queue
[00:01:03] Processing PRIORITY_BOOST
[00:01:03] Balancing pool
[00:01:03] Recording filesystem state for prepkg... done
[00:01:03] Building 154 packages using 8 builders
[00:01:03] Starting/Cloning builders
[00:01:07] Hit CTRL+t at any time to see build progress and stats
[00:01:07] [01] [00:00:00] Building ports-mgmt/pkg | pkg-1.12.0
load: 6.94  cmd: sh 85164 [piperd] 46.12r 0.00u 0.00s 0% 3732k
[12amd64-default] [2020-01-13_23h10m18s] [parallel_build:] Queued: 154 Built: 0   Failed: 0   Skipped: 0   Ignored: 0   Tobuild: 154  Time: 00:01:53
        [01]: ports-mgmt/pkg            | pkg-1.12.0               build          (00:00:31 / 00:00:46)
[00:01:53] Logs: /usr/local/poudriere/data/logs/bulk/12amd64-default/2020-01-13_23h10m18s
[00:01:53] WWW: https://yourdomain.com/poudriere//build.html?mastername=12amd64-default&build=2020-01-13_23h10m18s
[00:03:52] [01] [00:02:45] Finished ports-mgmt/pkg | pkg-1.12.0: Success
[00:03:52] [01] [00:00:00] Building print/indexinfo | indexinfo-0.3.1
[00:03:52] [02] [00:00:00] Building lang/perl5.30 | perl5-5.30.1
[00:03:52] [03] [00:00:00] Building devel/pkgconf | pkgconf-1.6.3,1
[00:03:52] [04] [00:00:00] Building converters/libiconv | libiconv-1.14_11
[00:03:52] [05] [00:00:00] Building devel/autoconf-wrapper | autoconf-wrapper-20131203
[00:03:52] [06] [00:00:00] Building textproc/xmlcatmgr | xmlcatmgr-2.2_2
[00:03:52] [07] [00:00:00] Building devel/libpthread-stubs | libpthread-stubs-0.4
[00:03:52] [08] [00:00:00] Building archivers/lzo2 | lzo2-2.10_1
```

Nginx Config

/usr/local/etc/nginx/nginx.conf

```
server {
    listen 80 default;
    server_name server_domain_or_IP;
    root /usr/local/share/poudriere/html;

    location /data {
        alias /usr/local/poudriere/data/logs/bulk;
        autoindex on;
    }

    location /packages {
        root /usr/local/poudriere/data;
        autoindex on;
    }
}
```

You should enable TLS (exercise left to the reader)

/usr/local/etc/nginx/mime.types

```
...
text/plain                              txt log;
...
```

Display build logs in browser window, rather than downloading them as a file

Configure pkg(8) clients

/usr/local/etc/pkg/repos/poudriere.conf

```
poudriere: {
  # url: "file:///usr/local/poudriere/data/packages/${ABI}",
  url: "pkg+https://pkg.yourdomain.com/packages/${ABI}",
  mirror_type: "srv",
  signature_type: "pubkey",
  pubkey: "/usr/local/etc/ssl/certs/poudriere.cert",
  enabled: yes
}
```

Disable the official FreeBSD repo and update package catalogue

```
# echo "FreeBSD: { enabled: no }" > /usr/local/etc/pkg/repos/FreeBSD.conf
# pkg update
```

So what are you waiting for?

I am here all week if you want some immediate mentoring!

The End

Questions?

https://svnweb.freebsd.org/ports/head/

https://github.com/freebsd/freebsd-ports