Keeping FreeBSD Applications Up-To-Date

Richard Bejtlich (taosecurity at gmail dot com)

25 August 2009

Sections:
---------
Introduction
FreeBSD Handbook
A Common Linux Experience
Simple Package Installation on FreeBSD
Checking for Vulnerable Packages with Portaudit
FreeBSD Package Repositories
Updating Packages by Deletion and Addition
Introducing the FreeBSD Ports Tree
Updatng the FreeBSD Ports Tree
Installing Portupgrade
Updating Packages Using Portupgrade
Removing Packages
Identifying and Removing Leaf Packages
Preparing to Build and Install Packages Using the Ports Tree
Building and Installing Packages Using the Ports Tree: A Simple Example
Building and Installing Packages Using the Ports Tree: A More Complicated Example
Install Packages Built on One System to Another System
Installing Screen Using a Remote FreeBSD Ports Tree
Reading /usr/ports/UPDATING
My Common Package Update Process
Conclusion

Introduction
---------

An important system administration task, and a principle of running a defensible network, is keeping operating systems and applications up-to-date. Running current software is critical when older services are vulnerable to exploitation. Obtaining new features not found in older applications is another reason to run current software. Fortunately, open source software offers a variety of means to give users a secure, capable computing environment.

This article presents multiple ways to keep FreeBSD applications up-to-date. I use a FreeBSD 7.1 system, and subsequent versions, to demonstrate how to install applications not included with the OS and how to keep those applications up-to-date. It is important to realize that this article discusses applications only; it does not discuss the OS. FreeBSD does not have a unified update mechanism for the OS and applications. By applications I mean software outside of the kernel and userland. For example, Debian systems can use the apt tool to keep the distribution and packaged applications up-to-date. FreeBSD does not have a single equivalent tool, so this article only addresses keeping applications up-to-date.

In this article I do not differentiate between an update and an upgrade. I will use the term update to describe any action that changes the version of an installed application.

I chose FreeBSD 7.1, released in January 2009, as my starting point because applications for it offer a security history suitable for describing multiple update cases. At the time of writing FreeBSD 7.2 is the latest STABLE release and 8.0 is in BETA. Readers wondering why someone might want to install an "old" OS version can imagine that there might be an application supported only on FreeBSD 7.1 and not yet officially ready for 7.2 or 8.0, prompting an administrator to run a 7.1 box.

All of the work done in this article was done remotely via OpenSSH. One danger of performing remote upgrades is losing connection during a critical phase of the process. One software-based way to deal with this issue is to conduct all remote upgrades within a screen(1) session. (http://www.freshports.org/misc/screen) Should you lose connectivity during the upgrade while running screen, your session will continue uninterrupted. The screen(1) program has suffered security problems in the past, so balance its features against the possible risks.

My advice on administering this reference platform is based on deploying FreeBSD on servers, workstations, and laptops since 2000. The article represents a mix of my interpretations of official FreeBSD documentation, inputs from mentors, and the result of my own experimentation and deployment strategies. This guide cannot be anywhere near a complete reference on keeping FreeBSD up-to-date or
maintaining a secure system. I strongly recommend reading the excellent FreeBSD Handbook as well as the
multiple helpful published books on FreeBSD.

FreeBSD Handbook and Absolute FreeBSD 2nd Ed
---------------------------------------------

Please note that Chapter 4, Installing Applications: Packages and Ports, is the authoritative source for
information on keeping FreeBSD applications up-to-date (http://www.freebsd.org/doc/en/books/handbook/
ports.html). The reason I wrote this article was to show how these various mechanisms apply in practice,
and which I prefer in production.

I must also recommend Michael W. Lucas' excellent book Absolute FreeBSD, 2nd Ed (No Starch, 2008).
Several other excellent FreeBSD writers have produced books, but Michael's is my favorite. For deeper
coverage on the topics in this article, please see the Handbook or Michael's book.

A Common Linux Experience
-------------------------

FreeBSD's application installation, maintainence, and removal process is sometimes confusing to those
with a Linux background. For purposes of a brief comparison, I will demonstrate how to install the Curl
application on a Debian 5.0 host using the apt-get tool. For authoritative documentation on using Apt,
please see http://www.debian.org/doc/manuals/apt-howto/.

To install Curl, the user simply enters 'apt-get install curl'.

shuttle02:~# uname -a
Linux shuttle02 2.6.26-1-686 #1 SMP Fri Mar 13 18:08:45 UTC 2009 i686 GNU/Linux

shuttle02:~# apt-get install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  ca-certificates libcurl3 libssh2-1 openssl
The following NEW packages will be installed:
  ca-certificates curl libcurl3 libssh2-1 openssl
0 upgraded, 5 newly installed, 0 to remove and 1 not upgraded.
Need to get 1687kB of archives.
After this operation, 4133kB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://http.us.debian.org stable/main openssl 0.9.8g-15+lenny1 [1036kB]
Get:2 http://security.debian.org stable/updates/main libcurl3 7.18.2-8lenny3 [228kB]
Get:4 http://http.us.debian.org stable/main libssh2-1 0.18-1 [64.3kB]
Get:5 http://security.debian.org stable/updates/main curl 7.18.2-8lenny3 [208kB]
Fetched 1687kB in 1s (1290kB/s)
Preconfiguring packages ...
Selecting previously deselected package openssl.
(Reading database ... 51192 files and directories currently installed.)
Unpacking openssl (from .../openssl_0.9.8g-15+lenny1_i386.deb) ...
Selecting previously deselected package ca-certificates.
Unpacking ca-certificates (from .../ca-certificates_20080809_all.deb) ...
Selecting previously deselected package libssh2-1.
Unpacking libssh2-1 (from .../libssh2-1 0.18-1_i386.deb) ...
Selecting previously deselected package libcurl3.
Unpacking libcurl3 (from .../libcurl3 7.18.2-8lenny3_i386.deb) ...
Selecting previously deselected package curl.
Unpacking curl (from .../curl 7.18.2-8lenny3_i386.deb) ...
Processing triggers for man-db ...
Setting up openssl (0.9.8g-15+lenny1) ...
Setting up ca-certificates (20080809) ...
Updating certificates in /etc/ssl/certs...done.
Running hooks in /etc/ca-certificates/update.d...done.
Setting up libssh2-1 (0.18-1) ...
Setting up libcurl3 (7.18.2-8lenny3) ...
Setting up curl (7.18.2-8lenny3) ...
shuttle02:~# which curl
/usr/bin/curl

That is easy enough!

Simple Package Installation on FreeBSD
--------------------------------------

FreeBSD users can install Curl using a similar method.

freebsd7# uname -a
FreeBSD freebsd7.localdomain 7.1-RELEASE FreeBSD 7.1-RELEASE #0: Thu Aug 20 11:24:04 EDT 2009
root@freebsd7.localdomain:/usr/obj/usr/src/sys/FREEBSD7 i386

freebsd7# setenv HTTP_PROXY http://172.16.2.1:3128
freebsd7# pkg_add -vr curl

scheme: [ftp]
user: []
password: []
host: [ftp.freebsd.org]
port: [0]
document: [/pub/FreeBSD/ports/i386/packages-7.1-release/Latest/curl.tbz]
scheme: [http]
user: []
password: []
host: [172.16.2.1]
port: [3128]
document: []

--> 172.16.2.1:3128
looking up 172.16.2.1
connecting to 172.16.2.1:3128
>>> GET ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7.1-release/Latest/curl.tbz HTTP/1.1
>>> Host: ftp.freebsd.org
>>> User-Agent: pkg_add libfetch/2.0
>>> Connection: close
>>>
<<< HTTP/1.0 200 Gatewaying
<<< Server: squid/2.7.STABLE6
<<< Date: Mon, 24 Aug 2009 19:52:19 GMT
<<< Content-Type: text/plain
<<< Content-Length: 1088297
content length: [1088297]
<<< Last-Modified: Mon, 08 Sep 2008 10:45:09 GMT
last modified: [2008-09-08 10:45:09]
<<< X-Cache: MISS from r200a.taosecurity.com
<<< Via: 1.0 r200a.taosecurity.com:3128 (squid/2.7.STABLE6)
<<< Connection: close
<<<
offset 0, length -1, size -1, clength 1088297
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7.1-release/Latest/curl.tbz...
x +CONTENTS
x +COMMENT
x +DESC
x +MTREE_DIRS
x man/man1/curl.1.gz
...edited...
x share/examples/curl/synctime.c
tar command returns 0 status
Done.
extract: Package name is curl-7.18.0
extract: CWD to /usr/local
extract: /usr/local/man/man1/curl.1.gz
...edited...
extract: /usr/local/share/examples/curl/synctime.c
extract: execute '/sbin/ldconfig -m /usr/local/lib'
extract: CWD to .
Running mtree for curl-7.18.0..
mtree -U -f +MTREE_DIRS -d -e -p /usr/local >/dev/null
Attempting to record package into /var/db/pkg/curl-7.18.0..
Package curl-7.18.0 registered in /var/db/pkg/curl-7.18.0

curl-7.18.0         Non-interactive tool to get files from FTP, GOPHER, HTTP(S)
curl-7.18.0

First we set a proxy for our environment.
The 'v' switch permits seeing verbose output. The command to install the Curl package on FreeBSD from a remote package repository requires the '-r' switch. You can see the location from where the package was retrieved in this output:
document: [/pub/FreeBSD/ports/i386/packages-7.1-release/Latest/curl.tbz]
...edited...
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7.1-release/Latest/curl.tbz...

If you visit the FTP server and look at the directory, you'll see that curl.tbz is really a symlink to the following:

The 'packages-7.1-release' directory means that the package curl-7.18.0.tbz is the version of the package built for the release of FreeBSD 7.1, as was "shipped on CD." Newer versions are available remotely, and I will describe how to acquire those later.

The pkg_info command shows the Curl package is now installed. I issue the 'rehash' command to ensure that curl is in the path for the user's shell.

Checking for Vulnerable Packages with Portaudit

FreeBSD's Portaudit tool is the easiest way to determine if any installed packages have security vulnerabilities. Portaudit relies on the FreeBSD VuXML site (http://www.vuxml.org/freebsd/) for knowledge of vulnerable packages. Don't worry about the term "port" vs. package right now; I'll address it soon. To see if the installed packages have any vulnerabilities, install and run Portaudit.

freebsd7# pkg_add -r portaudit
===> To check your installed ports for known vulnerabilities now, do:
/usr/local/sbin/portaudit -Fda
freebsd7# rehash
freebsd7# portaudit -Fdav
Trying to fetch from http://www.FreeBSD.org/ports/...
auditfile.tbz 100% of 57 kB 95 kbps
New database installed.
Database created: Mon Aug 24 15:10:03 EDT 2009
Affected package: curl-7.18.0 (matched by curl>=5.11<7.19.4)
Type of problem: curl -- cURL/Libcurl Location: Redirect URLs Security Bypass.
Reference: <http://www.FreeBSD.org/ports/portaudit/5d433534-f41c-402e-ade5-e0a2259a7cb6.html>
1 problem(s) in your installed packages found.
You are advised to update or deinstall the affected package(s) immediately.

We see that Curl has a security vulnerability that requires a patch. We'll address ways to fix that in the following sections.

FreeBSD Package Repositories
It is important to understand what version of packages are made available through the FreeBSD project. Visiting ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/ shows what's available for the i386 platform. (The FreeBSD team also regularly builds packages for the amd64 platform.)

Oct 23  2006  Symbolic Link    packages -> packages-stable
Aug 24 12:57  Directory        packages-6-stable
Nov 21  2008  Directory        packages-6.4-release
Aug 22 21:56  Directory        packages-7-stable
Dec 22  2008  Directory        packages-7.1-release
May  1 18:42  Directory        packages-7.2-release
Aug 20 21:36  Symbolic Link    packages-8-current -> packages-8-stable
Aug 24 13:05  Directory        packages-8-stable
Aug 18 00:32  Directory        packages-9-current
Feb  9  2008  Symbolic Link    packages-current -> packages-8-current/
Mar  1  2008  Symbolic Link    packages-stable -> packages-7-stable

For the purposes of our system (running a version of FreeBSD 7.x), we care about the packages-7* directories.

Earlier we installed Curl and it was retrieved from the packages-7.1-release directory. The packages-7.2-release directory is likely to contain a newer version of Curl since 7.2 was released months after 7.1. If we check that directory, we find curl-7.19.4.tbz is available, with a build date of Apr 21.

Looking back at our Portaudit output, we see that the 7.19.4 is not vulnerable

Let's remember that for now, but also look at the other packages-7* directory, packages-7-stable. In that directory we find curl-7.19.6_1.tbz available, with a build date of Aug 22. That version is also not vulnerable.

So what does packages-7-stable mean? That directory contains the latest packages built for FreeBSD 7.x. If you're thinking that you might want to install packages from that site on a regular basis, you are right. I'll cover that soon. For now we want to know how to update Curl to a newer version.

Updating Packages by Deletion and Addition

Deleting an installed package and adding a new version is one way to update a package. The easiest way to accomplish this goal is to change to the /var/db/pkg directory and use the pkg_delete command.

```
freebsd7# pkg_info
curl-7.18.0         Non-interactive tool to get files from FTP, GOPHER, HTTP(S)
portaudit-0.5.12    Checks installed ports against a list of security vulnerabi
freebsd7# cd /var/db/pkg/
freebsd7# ls
freebsd7# pkg_delete curl-7.18.0/
freebsd7# pkg_info
curl-7.18.0         Non-interactive tool to get files from FTP, GOPHER, HTTP(S)
portaudit-0.5.12    Checks installed ports against a list of security vulnerabi
```

With Curl deleted, we can add the new version. For demonstration purposes we'll add the version shipped with FreeBSD 7.2 RELEASE. To tell pkg_add how to get that package, we set the PACKAGESITE variable.

```
freebsd7# setenv PACKAGESITE ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7.2-release/Latest/
```

```
freebsd7# pkg_add -vr curl
scheme:   [ftp]
user:     []
password: []
host:     [ftp.freebsd.org]
```
port: [0]
document: [//pub/FreeBSD/ports/i386/packages-7.2-release/Latest/curl.tbz]
scheme: [http]
user: []
password: []
host: [172.16.2.1]
port: [3128]
document: []

--> 172.16.2.1:3128
looking up 172.16.2.1
connecting to 172.16.2.1:3128

>>> GET ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7.2-release/Latest/curl.tbz HTTP/1.1
>>> Host: ftp.freebsd.org
>>> User-Agent: pkg_add libfetch/2.0
>>> Connection: close

<<< HTTP/1.0 200 Gatewaying
<<< Server: squid/2.7.STABLE6
<<< Date: Mon, 24 Aug 2009 20:00:58 GMT
<<< Content-Type: text/plain
<<< Content-Length: 1097934
content length: [1097934]
<<< Last-Modified: Mon, 13 Apr 2009 21:18:46 GMT
last modified: [2009-04-13 21:18:46]
<<< X-Cache: MISS from r200a.taosecurity.com
<<< Via: 1.0 r200a.taosecurity.com:3128 (squid/2.7.STABLE6)
<<< Connection: close

offset 0, length -1, size -1, clength 1097934
Fetching ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7.2-release/Latest/curl.tbz...

Package 'curl-7.19.4' depends on 'ca_root_nss-3.11.9_2' with 'security/ca_root_nss' origin.

scheme: [ftp]
user: []
password: []
host: [ftp.freebsd.org]
port: [0]
document: [ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7.2-release/All/ca_root_nss-3.11.9_2.tbz]
scheme: [http]
user: []
password: []
host: [172.16.2.1]
port: [3128]
document: []

--> 172.16.2.1:3128
looking up 172.16.2.1
connecting to 172.16.2.1:3128
requesting ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7.2-release/All/ca_root_nss-3.11.9_2.tbz

>>> GET ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7.2-release/All/ca_root_nss-3.11.9_2.tbz HTTP/1.1
>>> Host: ftp.freebsd.org
>>> User-Agent: pkg_add libfetch/2.0
>>> Connection: close

<<< HTTP/1.0 200 Gatewaying
<<< Server: squid/2.7.STABLE6
<<< Date: Mon, 24 Aug 2009 20:01:02 GMT
<<< Content-Type: text/plain
Thus far we have worked with FreeBSD packages. They are convenient, but they do not independently support an update mechanism. The reference against which packages are compared to determine their "freshness" is the FreeBSD ports tree. On our reference FreeBSD 7.1 system, we installed the version of...
the ports tree that shipped with FreeBSD 7.1 RELEASE.

The FreeBSD ports tree can be found in the /usr/ports directory.

freebsd7# ls /usr/ports
.cvsignore arabic emulators mbone shells
CHANGES archivers finance misc sysutils
COPYRIGHT astro french multimedia textproc
GIDs audio ftp net ukrainian
INDEX-7 benchmarks games net-im vietnamese
KNOBS biology german net-mgmt www
LEGAL cad graphics net-p2p x11
MOVED chinese hebrew news x11-clocks
Makefile commss hungarian palm x11-drivers
Mk converters irc polish x11-fm
README databases japanese ports-mgmt x11-fonts
Templates deskutils java portuguese x11-servers
Tools devel korean print x11-themes
UIDs distfiles langTY russian x11-toolkits
UPDATING dns mail science x11-wm
accessibility editors math security

For the purposes of this article, it is sufficient to know that FreeBSD ports are a framework upon which application source code is installed on a FreeBSD system.

Updating the FreeBSD Ports Tree
-------------------------------

The easiest way to update the FreeBSD ports tree is to use Colin Percival's Portsnap tool (http://www.daemonology.net/portsnap/), now shipped with FreeBSD. First run 'portsnap fetch' to download a compressed version of the FreeBSD ports tree needed by portsnap.

freebsd7# portsnap fetch
Looking up portsnap.FreeBSD.org mirrors... 3 mirrors found.
Fetching public key from portsnap2.FreeBSD.org... done.
Fetching snapshot tag from portsnap2.FreeBSD.org... done.
Fetching snapshot metadata... done.
Fetching snapshot generated at Sun Aug 23 20:41:07 EDT 2009:
e4a063906c569a6d82cdd053ddad2ced013f53d80723ef4100% of 59 MB 359 kbps 00m00s
Extracting snapshot... done.
Verifying snapshot integrity... done.
Fetching snapshot tag from portsnap2.FreeBSD.org... done.
Fetching snapshot metadata... done.
Fetching 4 metadata patches... done.
Applying metadata patches... done.
Fetching 0 metadata files... done.
Fetching 36 patches...10....20....30... done.
Applying patches... done.
Fetching 3 new ports or files... done.

freebsd7# portsnap extract
/usr/ports/.cvsignore
/usr/ports/CHANGES
/usr/ports/COPYRIGHT
...edited...
/usr/ports/x11/zenity/
Building new INDEX files... done.

In the future, we do not need to run 'portsnap extract'. Instead, we run 'portsnap update'.

With the FreeBSD ports tree installed, we can use the pkg_version tool to check what packages need to be updated. This checks for any update, not just security updates as we saw with Portaudit.

freebsd7# pkg_version -v
ca_root_nss-3.11.9_2 = up-to-date with port
curl-7.19.4 < needs updating (port has 7.19.6_1)
As we can see, two of our packages (Curl and Portaudit) have newer versions available.

Installing Portupgrade
------------------------

The FreeBSD Portupgrade tool (http://wiki.freebsd.org/portupgrade) is a powerful tool that offers the ability to update packages using only packages. Portupgrade is a bit "heavy" in the sense that it requires installing Ruby as a dependency, whereas other options do not require such dependencies. However, other options do not seem to have the ability to dictate installing packages instead of building from ports.

We'll install Portupgrade from the 7-stable package collection by setting the appropriate environment variable and then invoking pkg_add. When we start we have only 3 packages installed.

    FreeBSD7# pkg_info
    ca_root_nss-3.11.9_2 The root certificate bundle from the Mozilla Project
    curl-7.19.4         Non-interactive tool to get files from FTP, GOPHER, HTTP(S)
    portaudit-0.5.12    Checks installed ports against a list of security vulnerabi

    FreeBSD7# setenv PACKAGESITE ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7-stable/Latest/
    FreeBSD7# pkg_add -vr portupgrade

    scheme: [ftp]
    user: []
    password: []
    host: [ftp.freebsd.org]
    port: [0]
    document: [/pub/FreeBSD/ports/i386/packages-7-stable/Latest/portupgrade.tbz]
    scheme: [http]
    user: []
    password: []
    host: [172.16.2.1]
    port: [3128]
    document: []

    ---> 172.16.2.1:3128
    looking up 172.16.2.1
    connecting to 172.16.2.1:3128
    >>> GET ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7-stable/Latest/portupgrade.tbz HTTP/1.1
    >>> Host: ftp.freebsd.org
    >>> User-Agent: pkg_add libfetch/2.0
    >>> Connection: close
    ...edited...
    Running mtree for portupgrade-2.4.6_3,2..
    mtree -U -f +MTREE_DIRS -d -e -p /usr/local >/dev/null
    Attempting to record package into /var/db/pkg/portupgrade-2.4.6_3,2..
    Trying to record dependency on package 'ruby-1.8.7.160_4,1' with 'lang/ruby18' origin.
    Trying to record dependency on package 'db41-4.1.25_4' with 'databases/db41' origin.
    Trying to record dependency on package 'ruby18-bdb-0.6.5_1' with 'databases/ruby-bdb' origin.
    Package portupgrade-2.4.6_3,2 registered in /var/db/pkg/portupgrade-2.4.6_3,2

Fill ALT_PKGDEP section in pkgtools.conf file for portupgrade
be aware of alternative dependencies you use.
E.g.
ALT_PKGDEP = {
   'www/apache13' => 'www/apache13-modssl',
   'print/ghostscript-gnu' => 'print/ghostscript-gpl',
}

Note also, portupgrade knows nothing how to handle ports with different suffixes (E.g. -nox11). So you should explicitly define variables (E.g. WITHOUT_X11=yes) for the ports in /etc/make.conf or pkgtools.conf
(MAKE_ARGS section) files.
After adding Portupgrade, we have 7 packages installed. You can see the Ruby and Berkeley DB dependencies installed by Portupgrade.

Updating Packages Using Portupgrade
-----------------------------------

With Portupgrade installed, we can use the portversion tool to determine what packages need updating.

freebsd7# portversion -v
[Rebuilding the pkgdb <format:bdb_btree> in /var/db/pkg ... - 7 packages found (-0 +7) .... done]
[Updating the portsdb <format:bdb_btree> in /usr/ports ... - 20616 port entries
 found ..........1000............2000............3000............4000............5000............6000
 ..........7000............8000............9000............10000............11000............12000
 ..........13000............14000............15000............16000............17000............18000
 ..........19000............20000...... ..... done]

c_root_nss-3.11.9_2        = up-to-date with port
curl-7.19.4                 < needs updating (port has 7.19.6_1)
db41-4.1.25_4               = up-to-date with port
portaudit-0.5.12            < needs updating (port has 0.5.13)
portupgrade-2.4.6_4         = up-to-date with port
ruby-1.8.7.160_4,1          = up-to-date with port
ruby18-bdb-0.6.5_1          = up-to-date with port

If we just want to see packages that need updating, we run 'portversion -v -l "<".'.

When we run portversion, we see it builds a package database ("pkgdb"), then a ports database ("portsdb") for its own use. We could have used pkg_version to produce the same output:

freebsd7# pkg_version -v
c_root_nss-3.11.9_2        = up-to-date with port
curl-7.19.4                 < needs updating (port has 7.19.6_1)
db41-4.1.25_4               = up-to-date with port
portaudit-0.5.12            < needs updating (port has 0.5.13)
portupgrade-2.4.6_4         = up-to-date with port
ruby-1.8.7.160_4,1          = up-to-date with port
ruby18-bdb-0.6.5_1          = up-to-date with port

So, with this information, how can we update packages that need updating?

The following advice is based on my personal preferences, but when updating packages I prefer to use packages when possible. (I'll discuss alternatives later.) The following example will update the packages for which newer version are available.

First we set proxy and PACKAGESITE variables, and then we invoke Portupgrade.

freebsd7# setenv HTTP_PROXY http://172.16.2.1:3128
freebsd7# setenv PACKAGESITE ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7-stable/Latest/
freebsd7# portupgrade -vaPP
--- Session started at: Tue, 25 Aug 2009 09:28:59 -0400
--- Checking for the latest package of 'ftp/curl'
** No such file or directory - /usr/ports/packages/All
--- Fetching the package(s) for 'curl-7.19.6_1' (ftp/curl)
--- Fetching curl-7.19.6_1
++ Will try the following sites in the order named:
--- Invoking a command: /usr/bin/fetch -o '/var/tmp/portupgradeRvOuj4wl/curl-7.19.6_1.tbz' 'ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7-stable/All/curl-7.19.6_1.tbz'
/var/tmp/portupgradeRvOuj4wl/curl-7.19.6_1.tbz100% of 1103 kB 2824 kBps
--- Downloaded as curl-7.19.6_1.tbz
--- Identifying the package /var/tmp/portupgradeRvOuj4wl/curl-7.19.6_1.tbz
--- Saved as /usr/ports/packages/All/curl-7.19.6_1.tbz
--- Listing the results (+:done / -:ignored / *:skipped / !:failed)
   + curl-7.19.6_1
--- Packages processed: 1 done, 0 ignored, 0 skipped and 0 failed
--- Found a package of 'ftp/curl': /usr/ports/packages/All/curl-7.19.6_1.tbz (curl-7.19.6_1)
--- Located a package version 7.19.6_1 (/usr/ports/packages/All/curl-7.19.6_1.tbz)
--- Upgrading 'curl-7.19.4' to 'curl-7.19.6_1' (ftp/curl) using a package
--- Updating dependency info
--- Fixing up dependencies before creating a package
--- Backing up the old version
--- Uninstalling the old version
--- Deinstalling 'curl-7.19.4'
--- Preserving /usr/local/lib/libcurl.so.5 as /usr/local/lib/compat/pkg/libcurl.so.5
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 6 packages found (-1 +0) (...) done]
--- Uninstallation of curl-7.19.6_1 ended at: Tue, 25 Aug 2009 09:29:34 -0400 (consumed 00:00:15)
--- Installing the new version via the package
--- Removing temporary files and directories
--- Removing old package'
--- Installation of curl-7.19.6_1 ended at: Tue, 25 Aug 2009 09:29:38 -0400 (consumed 00:00:04)
--- Cleaning out obsolete shared libraries
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 7 packages found (-0 +1) . done]
--- Upgrade of ftp/curl ended at: Tue, 25 Aug 2009 09:29:49 -0400 (consumed 00:00:30)
--- ** Upgrade tasks 2: 1 done, 0 ignored, 0 skipped and 0 failed
--- Checking for the latest package of 'ports-mgmt/portaudit'
--- Fetching the package(s) for 'portaudit-0.5.13' (ports-mgmt/portaudit)
--- Fetching portaudit-0.5.13
++ Will try the following sites in the order named:
--- Invoking a command: /usr/bin/fetch -o '/var/tmp/portupgradeY8v1o54H/portaudit-0.5.13.tbz' 'ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7-stable/All/portaudit-0.5.13.tbz'
/var/tmp/portupgradeY8v1o54H/portaudit-0.5.13.100% of 10 kB 1842 kBps
--- Downloaded as portaudit-0.5.13.tbz
--- Identifying the package /var/tmp/portupgradeY8v1o54H/portaudit-0.5.13.tbz
--- Saved as /usr/ports/packages/All/portaudit-0.5.13.tbz
--- Listing the results (+:done / -:ignored / *:skipped / !:failed)
   + portaudit-0.5.13
--- Packages processed: 1 done, 0 ignored, 0 skipped and 0 failed
--- Found a package of 'ports-mgmt/portaudit': /usr/ports/packages/All/portaudit-0.5.13.tbz (portaudit-0.5.13)
--- Located a package version 0.5.13 (/usr/ports/packages/All/portaudit-0.5.13.tbz)
--- Upgrading 'portaudit-0.5.12' to 'portaudit-0.5.13' (ports-mgmt/portaudit) using a package
--- Updating dependency info
--- Uninstallation of portaudit-0.5.12 started at: Tue, 25 Aug 2009 09:30:00 -0400
--- Fixing up dependencies before creating a package
--- Backing up the old version
--- Uninstalling the old version
--- Deinstalling 'portaudit-0.5.12'

The portaudit package has been deleted.
If you're *not* upgrading and won't be using it any longer, you may want to remove the portaudit database:

```
rn -Rf /var/db/portaudit
```

[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 6 packages found (-1 +0) (...) done]
Uninstallation of portaudit-0.5.12 ended at: Tue, 25 Aug 2009 09:30:20 -0400 (consumed 00:00:20)

Installing the new version via the package

Removing temporary files and directories

Removing old package'

Installation of portaudit-0.5.13 ended at: Tue, 25 Aug 2009 09:30:22 -0400 (consumed 00:00:01)

Cleaning out obsolete shared libraries

[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 7 packages found (-0 +1) . done]

Upgrade of ports-mgmt/portaudit ended at: Tue, 25 Aug 2009 09:30:31 -0400 (consumed 00:00:32)

** Upgrade tasks 2: 2 done, 0 ignored, 0 skipped and 0 failed

Listing the results (+:done / -:ignored / *:skipped / !:failed)
  + ftp/curl (curl-7.19.4)
  + ports-mgmt/portaudit (portaudit-0.5.12)

Packages processed: 2 done, 0 ignored, 0 skipped and 0 failed

Session ended at: Tue, 25 Aug 2009 09:30:40 -0400 (consumed 00:01:40)

When done, we can see the packages have been updated.

```bash
freebsd7# portversion -v
c_ca_root_nss-3.11.9_2   =  up-to-date with port
curl-7.19.6_1           =  up-to-date with port
db41-4.1.25.4           =  up-to-date with port
portaudit-0.5.13         =  up-to-date with port
portupgrade-2.4.6_3,2    =  up-to-date with port
ruby-1.8.7.160_4,1       =  up-to-date with port
ruby18-bdb-0.6.5_1       =  up-to-date with port
```

So what just happened? Portupgrade found that Curl and Portaudit were out-of-date. It downloaded the newest packages from the packages-7-stable directory on a remote FreeBSD FTP server, uninstalled the out-of-date package, and installed the up-to-date package.

If you noticed in the Portupgrade output, the program stores copies of the packages it downloads in the /usr/ports/packages/All directory.

```bash
freebsd7# ls /usr/ports/packages/All
curl-7.19.6_1.tbz portaudit-0.5.13.tbz
```

By specifying the `-a` switch we told Portupgrade to update all packages. The `-v` switch enabled verbose mode. The `-PP` switch told Portupgrade to only use packages, and it retrieved those packages from the public FreeBSD package repository.

There are other ways to invoke Portupgrade, such as telling it to only update individual packages, and then update their dependencies, and so on. I prefer this simpler approach of updating everything that requires it.

FreeBSD Package Dependencies
----------------------------

Dependencies are packages which are required in order to run other packages. We can use the `pkg_info` command to learn what packages a specified package depends on.

```bash
freebsd7# pkg_info -rx curl
Information for curl-7.19.6_1:
Depends on:
  Dependency: ca_root_nss-3.11.9_2
```

Here we see that curl depends on the ca_root nss package. The `-r` command tells `pkg_info` to display the packages on which curl depends. The `-x` switch tells `pkg_info` to do a regular expression match, so we don't have to list the whole package name.

Does anything depend on curl?

```bash
freebsd7# pkg_info -Rx curl
Information for curl-7.19.6_1:
```

The `-R` switch shows that nothing depends on curl.
If we ran this command for ca_root_nss, however, we would see that curl requires it.

freebsd7# pkg_info -Rx ca_root_nss
Information for ca_root_nss-3.11.9_2:

Required by:
curl-7.19.6_1

Another way to understand these relationships is to install the pkg_tree package.

freebsd7# pkg_add -r pkg_tree
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7-stable/Latest/pkg_tree.tbz... Done.
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7-stable/All/perl-5.8.9_3.tbz... Done.
Removing stale symlinks from /usr/bin...
  Skipping /usr/bin/perl
  Skipping /usr/bin/perl5
Done.
Creating various symlinks in /usr/bin...
  Symlinking /usr/local/bin/perl5.8.9 to /usr/bin/perl
  Symlinking /usr/local/bin/perl5.8.9 to /usr/bin/perl5
Done.
Cleaning up /etc/make.conf... Done.
Spamming /etc/make.conf... Done.
Cleaning up /etc/manpath.config... Done.
Spamming /etc/manpath.config... Done.

From now on, when adding new packages, it helps to update the package database maintained by Portupgrade, using the pkgdb command.

freebsd7# pkgdb -vu
---> Updating the pkgdb
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 9 packages found (-0 +2) .. done]

If you forget to run pkgdb after installing a package, it’s not a big problem. Any time a tool in the Portupgrade suite is invoked (such as portupgrade itself, or other tools), the pkgdb will be updated.

During the pkg_tree installation process we saw Perl installed as a dependency of pkg_tree. Once installed, run pkg_tree and tell it to show what packages curl depends on.

freebsd7# pkg_tree curl
curl-7.19.6_1
  ca_root_nss-3.11.9_2

Portupgrade presents a more complicated example.

freebsd7# pkg_tree portupgrade
portupgrade-2.4.6_3,2
  ruby-1.8.7.160_4,1
  db41-4.1.25_4
  ruby18-bdb-0.6.5_1

We can go one step farther to follow the dependency chain using the -v switch.

freebsd7# pkg_tree -v portupgrade
portupgrade-2.4.6_3,2
  ruby-1.8.7.160_4,1
  db41-4.1.25_4
  ruby18-bdb-0.6.5_1
    ruby-1.8.7.160_4,1
    db41-4.1.25_4

Now we see that Portupgrade depends on ruby, db41, and ruby18-bdb. However, ruby18-bdb depends on ruby and db41 as well.

Running pkg_tree with no options shows all package dependencies.
Understanding dependencies is important, because FreeBSD won't let you delete a package when another package depends on it. We'll look at that next.

Removing Packages
-----------------
For the following examples we add the open source text email client Mutt to our system.

freebsd7# pkg_add -r mutt
Fetching ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7-stable/All/urlview-0.9_2.tbz... Done.
Fetching ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7-stable/All/ispell-3.3.02_4.tbz... Done.
Fetching ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7-stable/All/mime-support-3.46.1.tbz... Done.
Fetching ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7-stable/All/png-1.2.38.tbz... Done.
Fetching ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7-stable/All/libiconv-1.13.1.tbz... Done.
Fetching ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7-stable/All/libslang2-2.1.4_1.tbz... Done.
Fetching ftp://ftp.freebsd.org//pub/FreeBSD/ports/i386/packages-7-stable/All/gettext-0.17_1.tbz... Done.

freebsd7# pkgdb -vu
-->  Updating the pkgdb
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 18 packages found (-0 +9) ........ done]

When you check Mutt's dependencies, you find several:

freebsd7# pkg_tree mutt
mutt-1.4.2.3_3
  \ urlview-0.9_2
  \ ispell-3.3.02_4
  \ mime-support-3.46.1
  \ png-1.2.38
  \ pcre-7.9
  \ libiconv-1.13.1
  \ libslang2-2.1.4_1
  \ gettext-0.17_1

If you try to delete, say, the pcre package, the attempt will fail.

freebsd7# pkg_delete pcre-7.9/
pkg_delete: package 'pcre-7.9' is required by these other packages
and may not be deinstalled:
libslang2-2.1.4_1
mutt-1.4.2.3_3

If you try using the pkg_deinstall tool shipped with Portupgrade, it will also fail.
freebsd7# pkg_deinstall pcre-7.9/
----> Deinstalling 'pcre-7.9'
pkg_delete: package 'pcre-7.9' is required by these other packages
and may not be deinstalled:
libslang2-2.1.4_1
mutt-1.4.2.3_3
** Listing the failed packages (-:ignored / *:skipped / !:failed)

! pcre-7.9 (pkg_delete failed)

This is a strength of using the packages system, not a weakness. We don't want to break the system by removing a package on which others depend.

What if we decided to remove Mutt? We could check what depends on it using pkg_info again.

freebsd7# pkg_info -Rx mutt
Information for mutt-1.4.2.3_3:

Nothing depends on Mutt. So, if we wanted to, we could simply delete it using pkg_deinstall or pkg_delete. However, when we installed Mutt, it brought 8 dependencies along with it. Wouldn't it be good to remove those as well? We can use the pkg_deinstall command with the -R switch for that purpose.

freebsd7# pkg_deinstall -R mutt-1.4.2.3_3/
----> Deinstalling 'mutt-1.4.2.3_3'
----> Deinstalling 'urlview-0.9.2'
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 17 packages found (-1 +0) (... done]
----> Deinstalling 'libslang2-2.1.4_1'
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 16 packages found (-1 +0) (... done]
----> Deinstalling 'pcre-7.9'
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 15 packages found (-1 +0) (... done]
----> Deinstalling 'mime-support-3.46.1'
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 14 packages found (-1 +0) (... done]
----> Deinstalling 'png-1.2.38'
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 13 packages found (-1 +0) (... done]
----> Deinstalling 'gettext-0.17_1'
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 12 packages found (-1 +0) (... done]
----> Deinstalling 'libiconv-1.13.1'
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 11 packages found (-1 +0) (... done]
----> Deinstalling 'ispell-3.3.02_4'
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 10 packages found (-1 +0) (... done]
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 9 packages found (-1 +0) (... done]

We've now completely removed Mutt and the packages on which Mutt depended.

Identifying and Removing Unwanted Packages
---------------------------------------------

For the purposes of the next example, I install Nmap and Tcpflow.

freebsd7# pkg_add -r nmap
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7-stable/All/pkg-config-0.23_1.tbz... Done.
freebsd7# pkg_add -r tcpflow
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-7-stable/Latest/tcpflow.tbz... Done.

freebsd7# pkgdb -vu
----> Updating the pkgdb
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 14 packages found (-0 +5) ..... done]

Let's imagine that a while passes, and later we'd like to perform some housecleaning on our installed packages. I periodically install packages for a single task, and then leave them behind. To perform housecleaning, I prefer using the pkg_cutleaves tool.

freebsd7# pkg_add -r pkg_cutleaves
freebsd7# pkgdb -vu
--->  Updating the pkgdb
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 15 packages found (-0 +1) . done]

freebsd7# rehash

Next I invoke pkg_cutleaves. I'm looking for packages I'd like to remove. Nmap and Tcpflow catch my eye. When I want to keep a package, I hit [return] to keep it. When I want to delete a package, I hit d. When asked if I want to go on with new leaf packages, I enter y and continue the process.

going...

freebsd7# pkg_cutleaves
Package 1 of 7:
curl-7.19.6_1 - Non-interactive tool to get files from FTP, GOPHER, HTTP(S) servers
curl-7.19.6_1 - [keep]/(d)elete/(f)lush marked pkgs/(a)bort?
  ** Keeping curl-7.19.6_1.

Package 2 of 7:
nmap-5.00 - Port scanning utility for large networks
nmap-5.00 - [keep]/(d)elete/(f)lush marked pkgs/(a)bort? d
  ** Marking nmap-5.00 for removal.

Package 3 of 7:
pkg_cutleaves-20090810 - Interactive script for deinstalling 'leaf' packages
pkg_cutleaves-20090810 - [keep]/(d)elete/(f)lush marked pkgs/(a)bort?
  ** Keeping pkg_cutleaves-20090810.

Package 4 of 7:
pkg_tree-1.1_1 - Get a 'graphical' tree-overview of installed packages
pkg_tree-1.1_1 - [keep]/(d)elete/(f)lush marked pkgs/(a)bort?
  ** Keeping pkg_tree-1.1_1.

Package 5 of 7:
portaudit-0.5.13 - Checks installed ports against a list of security vulnerabilities
portaudit-0.5.13 - [keep]/(d)elete/(f)lush marked pkgs/(a)bort?
  ** Keeping portaudit-0.5.13.

Package 6 of 7:
portupgrade-2.4.6_3,2 - FreeBSD ports/packages administration and management tool suite
portupgrade-2.4.6_3,2 - [keep]/(d)elete/(f)lush marked pkgs/(a)bort?
  ** Keeping portupgrade-2.4.6_3,2.

Package 7 of 7:
tcpflow-0.21_1 - A tool for capturing data transmitted as part of TCP connections
tcpflow-0.21_1 - [keep]/(d)elete/(f)lush marked pkgs/(a)bort? d
  ** Marking tcpflow-0.21_1 for removal.

Deleting nmap-5.00 (package 1 of 2).
Deleting tcpflow-0.21_1 (package 2 of 2).
Go on with new leaf packages ([yes]/no)? y

Package 1 of 2:
lua-5.1.4 - Small, compilable scripting language providing easy access to C code
lua-5.1.4 - [keep]/(d)elete/(f)lush marked pkgs/(a)bort? d
  ** Marking lua-5.1.4 for removal.

Package 2 of 2:
pcre-7.9 - Perl Compatible Regular Expressions library
pcre-7.9 - [keep]/(d)elete/(f)lush marked pkgs/(a)bort? d
  ** Marking pcre-7.9 for removal.

Deleting lua-5.1.4 (package 1 of 2).
Deleting pcre-7.9 (package 2 of 2).
Go on with new leaf packages ([yes]/no)? y

Package 1 of 1:
pkg-config-0.23_1 - A utility to retrieve information about installed libraries
pkg-config-0.23_1 - [keep]/(d)elete/(f)lush marked pkgs/(a)bort? d
** Marking pkg-config-0.23_1 for removal.

Deleting pkg-config-0.23_1 (package 1 of 1).
** Didn't find any new leaves to work with, exiting.
** Deinstalled packages:
  lua-5.1.4
  nmap-5.00
  pcire-7.9
  pkg-config-0.23_1
  tcpflow-0.21_1
** Number of deinstalled packages: 5

freebsd7# pkgdb -vu
 ---> Updating the pkgdb

[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 10 packages found (-5 +0) (...) done]

As we can see, the result of this process was removing Nmap, Tcpflow, and all of their dependencies. If we knew from the outset what we wanted to delete, we could have run pkg_deinstall as shown earlier. Here I like to use the "browsing" nature of pkg_cutleaves to identify packages which I don't necessarily realize I want to delete from the beginning.

Preparing to Build and Install Packages Using the Ports Tree
------------------------------------------------------------

Throughout this article we have installed packages installed by the FreeBSD project. However, because we have the ports tree installed on our system, we can build and install our own packages.

Earlier we updated our ports tree using Portsnap. Here we will update it again.

freebsd7# portsnap fetch
Looking up portsnap.FreeBSD.org mirrors... 3 mirrors found.
Fetching snapshot tag from portsnap2.FreeBSD.org... done.
Fetching snapshot metadata... done.
Fetching 4 metadata patches... done.
Applying metadata patches... done.
Fetching 0 metadata files... done.
Fetching 32 patches.....10....20....30. done.
Applying patches... done.
Fetching 2 new ports or files... done.

freebsd7# portsnap update
Removing old files and directories... done.
Extracting new files:
/usr/ports/audio/gtkpod/
/usr/ports/databases/pgadmin3/
/usr/ports/devel/cvsnt/
/usr/ports/devel/git/
/usr/ports/devel/jude-community/
/usr/ports/devel/p5-local-lib/
/usr/ports/games/wesnoth/
/usr/ports/graphics/Makefile
/usr/ports/graphics/mmrecover/
/usr/ports/graphics/rubygem-scruffy/
/usr/ports/mail/metal/
/usr/ports/net-mgmt/nagios-plugins/
/usr/ports/net/nss_ldapd/
/usr/ports/ports-mgmt/portmaster/
/usr/ports/security/fiked/
/usr/ports/security/swatch/
/usr/ports/security/vuxml/
/usr/ports/sysutils/e2fsprogs/
/usr/ports/sysutils/libchk/
/usr/ports/sysutils/virtualmin/
/usr/ports/textproc/ansifilter/
/usr/ports/textproc/yodl/
/usr/ports/www/Makefile
After running 'portsnap fetch' and 'portsnap update', we update the INDEX-7.db used by Portupgrade.

```
freebsd7# portsdb -u
[Updating the portsdb <format:bdb_btree> in /usr/ports ... - 20618 port entries
found ........1000........2000........3000........4000........5000.......
6000........7000........8000........9000........10000........11000.......
12000........13000........14000........15000........16000........17000.......
18000........19000........20000...... ..... done]
```

To keep it clear, Portsnap updates /usr/ports/INDEX-7 and portsdb updates /usr/ports/INDEX-7.db.

For the following examples we will deinstall Curl and its dependencies, and then reinstall them later.

```
freebsd7# pkg_deinstall -R curl
--->  Deinstalling 'curl-7.19.6_1'
--->  Deinstalling 'ca_root_nss-3.11.9_2'
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 9 packages found (-1 +0) (...) done]
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 8 packages found (-1 +0) (...) done]
```

For this example we will install the Screen application using the ports tree.

```
We'll start by using the port as an example of how to install a package. First we have to locate the port. We can use the 'make search name=' command in the /usr/ports directory.

freebsd7# cd /usr/ports
freebsd7# make search name=screen
...edited...
Port: screen-4.0.3_6
Path: /usr/ports/sysutils/screen
Info: A multi-screen window manager
Maint: cy@FreeBSD.org
B-deps:
R-deps: gettext-0.17_1 libiconv-1.13.1 texinfo-4.11
WWW: http://www.gnu.org/software/screen/
...truncated...
```

Here we see that sysutils/screen is the port we want.

```
freebsd7# cd /usr/ports/sysutils/screen
freebsd7# ls -al
```

```
total 32
```

```
drwxr-xr-x  3 root wheel   512 Aug 24 16:46 .
drwxr-xr-x  917 root wheel  17920 Aug 25 10:37 ...
```

```
-rw-r--r--  1 root wheel  2366 Feb 23 2009 Makefile
-rw-r--r--  1 root wheel  193 Oct 26 2006 distinfo
drwxr-xr-x  2 root wheel  512 Aug 24 16:46 files
-rw-r--r--  1 root wheel  554 Dec 27 2002 pkg-descr
-rw-r--r--  1 root wheel  853 Aug 30 2004 pkg-plist
```

These are the files we will need to build a package using this port.

To determine if there are any dependencies required to build a package from this port, we can use the following command.

```
```
freebsd7# make pretty-print-build-depends-list
There are no dependencies to build the package.
We can also see if any packages are required to run the package once installed.

freebsd7# make pretty-print-run-depends-list
There are no dependencies to run the package.

The next command I like to run when encountering a new port is 'make showconfig'. This command will show the options that will be set by default when building the package from the ports tree. The default settings are used to build the package provided by the FreeBSD project.

freebsd7# make showconfig

===> The following configuration options are available for screen-4.0.3_6:
CJK=OFF (default) "Treat CJK ambiguous characters as full width"
INFO=ON (default) "Build and install info documentation"
MAN=ON (default) "Build and install man pages"
NETHACK=ON (default) "Enable nethack-style messages"
XTERM_256=OFF (default) "Enable support for 256 colour xterm"
HOSTINLOCKED=OFF (default) "Print user@host in locked message"
SHOWENC=OFF (default) "Show encoding on the status line"

===> Use 'make config' to modify these settings

We can run 'make config' to change or just view these settings. This starts a Curses window.

```
Options for screen 4.0.3_6

[ ] CJK  Treat CJK ambiguous characters as full width
[ ] INFO Build and install info documentation
[ ] MAN  Build and install man pages
[ ] NETHACK Enable nethack-style messages
[ ] XTERM_256 Enable support for 256 colour xterm
[ ] HOSTINLOCKED Print user@host in locked message
[ ] SHOWENC Show encoding on the status line

[ OK ] Cancel
```

We leave the configuration as-is but hit OK to exit.

Running 'make config' has created the following entries in the /var/db/ports directory.

freebsd7# ls /var/db/ports
screen

freebsd7# ls /var/db/ports/screen/
options

freebsd7# cat /var/db/ports/screen/options
# This file is auto-generated by 'make config'.
# No user-servicable parts inside!
# Options for screen-4.0.3_6
_OPTIONS_READ=screen-4.0.3_6
WITHOUT_CJK=true
WITH_INFO=true
The ports tree will use these options when building the package.

Building and Installing Packages Using the Ports Tree: A Simple Example

At this point we are ready to proceed. In the /usr/ports/sysutils/screen directory, run 'make'.

```bash
freebsd7# make
===> Found saved configuration for screen-4.0.3_6
=> screen-4.0.3.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
    screen-4.0.3.tar.gz 100% of 820 kB 562 kbps
===> Extracting for screen-4.0.3_6
=> MD5 Checksum OK for screen-4.0.3.tar.gz.
=> SHA256 Checksum OK for screen-4.0.3.tar.gz.
===> Patching for screen-4.0.3_6
====> Applying FreeBSD patches for screen-4.0.3_6
===> Configuring for screen-4.0.3_6
this is screen version 4.0.3
checking for gcc... cc
checking for C compiler default output... a.out
checking whether the C compiler works... yes ...
edited...
cc -c -I. -I. -O2 -fno-strict-aliasing -pipe encoding.c
cc -o screen screen.o ansi.o fileio.o mark.o misc.o resize.o socket.o search.o tty.o term.o window.o utmp.o loadav.o putenv.o help.o termcap.o input.o attacher.o pty.o process.o display.o comm.o kmapdef.o acls.o braille.o braille_tsi.o logfile.o layer.o sched.o teln.o nethack.o encoding.o -ltermcap -lutil -lcrypt
To install we run 'make install'.

freebsd7# make install
====> Installing for screen-4.0.3_6
====> Generating temporary packing list
====> Checking if sysutils/screen already installed
...edited...
====> Registering installation for screen-4.0.3_6
====> SECURITY REPORT:
This port has installed the following binaries which execute with increased privileges.
/usr/local/bin/screen

If there are vulnerabilities in these programs there may be a security risk to the system. FreeBSD makes no guarantee about the security of ports included in the Ports Collection. Please type 'make deinstall' to deinstall the port if this is a concern.

For more information, and contact details about the security status of this software, see the following webpage:
http://www.gnu.org/software/screen/

freebsd7# pkgdb -vu
---> Updating the pkgdb
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 9 packages found (-0 +1) . done]

freebsd7# rehash
freebsd7# which screen
/usr/local/bin/screen
Screen is now installed.
For a more complicated example, let's install Curl using the ports tree. To install Curl via the ports tree, we need to know where it lives. We might remember it from the ftp/curl directory at the beginning of the article, but if we aren't sure we can again use the 'make search name=' command in the /usr/ports directory.

freebsd7# cd /usr/ports
freebsd7# make search name=curl
Port: curl-7.19.6_1
Path: /usr/ports/ftp/curl
Info: Non-interactive tool to get files from FTP, GOPHER, HTTP(S) servers
Maint: roam@FreeBSD.org
B-deps: perl-5.8.9_3
R-deps: ca_root_nss-3.11.9_2
WWW: http://curl.haxx.se/

Port: curlpp-0.7.0_1
Path: /usr/ports/ftp/curlpp
Info: A C++ wrapper for libcurl
Maint: roam@FreeBSD.org
B-deps: ca_root_nss-3.11.9_2 curl-7.19.6_1
R-deps: ca_root_nss-3.11.9_2 curl-7.19.6_1
WWW: ...

The first option is what we want, but many other programs with "curl" in their name are listed. In addition to running 'make search name=' we could also use 'make search key=' to specify a keyword for searching.

We see ftp/curl has what we want, we change there.

Again we run 'make showconfig' to see available options.

freebsd7# make showconfig
=> The following configuration options are available for curl-7.19.6_1:
  CARES=off (default) "Asynchronous DNS resolution via c-ares"
  CURL_DEBUG=off (default) "Enable curl diagnostic output"
  GNUTLS=off (default) "Use GNU TLS if OPENSSL is OFF"
  IPV6=on (default) "IPv6 support"
  KERBEROS4=off (default) "Kerberos 4 authentication"
  LDAP=off (default) "LDAP support (requires LDAP and SSL)"
  LIBIDN=off (default) "Internationalized Domain Names via libidn"
  LIBSSH2=off (default) "SCP/SFTP support via libssh2"
  NTLM=off (default) "NTLM authentication"
  OPENSSL=on (default) "OpenSSL support"
  PROXY=on (default) "Proxy support"
  TRACKMEMORY=off (default) "Enable curl memory diagnostic output"
=> Use 'make config' to modify these settings

If we run 'make config', we'll see a Curses interface like the following. Here I have enabled the LIBSSH2 option, which was off by default.
The ability to modify a package to meet local requirements, but then manage that package using standard tools, is one of the great strengths of the FreeBSD ports tree.

After selecting OK, I run 'make showconfig' again and notice the change.

```
freebsd7# make showconfig
```
curl-7.19.6.tar.bz2 doesn't seem to exist in /usr/ports/distfiles/.

fetch: transfer timed out

Attempting to fetch from http://curl.haxx.se/download/.

fetch: transfer timed out


fetch: transfer timed out

Attempting to fetch from http://curl.haxx.se/download/.

curl-7.19.6.tar.bz2 100% of 2292 kB 2336 kBps

Extracting for curl-7.19.6 1

MD5 Checksum OK for curl-7.19.6.tar.bz2.

SHA256 Checksum OK for curl-7.19.6.tar.bz2.

curl-7.19.6 1 depends on file: /usr/local/bin/perl5.8.9 - found

Patching for curl-7.19.6 1

curl-7.19.6 1 depends on file: /usr/local/bin/perl5.8.9 - found

Applying FreeBSD patches for curl-7.19.6 1

curl-7.19.6 1 depends on file: /usr/local/bin/perl5.8.9 - found

curl-7.19.6 1 depends on shared library: ssh2.1 - not found

Verifying install for ssh2.1 in /usr/ports/security/libssh2

libssh2-1.2.tar.gz doesn't seem to exist in /usr/ports/distfiles/.

fetch: transfer timed out

Attempting to fetch from http://www.libssh2.org/download/.
success: fetching libssh2-1.2.tar.gz

fetch: transfer timed out


libssh2-1.2.tar.gz 100% of 519 kB 2150 kBps

Extracting for libssh2-1.2,2

MD5 Checksum OK for libssh2-1.2.tar.gz.

SHA256 Checksum OK for libssh2-1.2.tar.gz.

Patching for libssh2-1.2,2

Configuring for libssh2-1.2,2

checking whether to enable maintainer-specific portions of Makefiles... no

checking for sed... /usr/bin/sed

checking for a BSD-compatible install... /usr/bin/install -c -o root -g wheel

checking whether build environment is sane... yes

...edited...

Registering installation for libssh2-1.2,2

Returning to build of curl-7.19.6 1

Configuring for curl-7.19.6 1

checking whether to enable maintainer-specific portions of Makefiles... no

checking whether to enable compiler optimizer... not specified (assuming yes)

checking whether to enable strict compiler warnings... no

checking whether to enable curl debug memory tracking... no

checking for sed... /usr/bin/sed

checking for grep... /usr/bin/grep

checking for egrep... /usr/bin/grep -E

checking for ar... /usr/bin/ar

checking for a BSD-compatible install... /usr/bin/install -c -o root -g wheel

checking whether build environment is sane... yes

...edited...

Making all in examples

Making all in libcurl

Now we run 'make install'.

freebsd7# make install

Installing for curl-7.19.6 1

curl-7.19.6 1 depends on file: /usr/local/share/certs/ca-root-nss.crt - found

curl-7.19.6 1 depends on shared library: ssh2.1 - found

Generating temporary packing list

Checking if ftp/curl already installed

Making install in lib

...edited...

Registering installation for curl-7.19.6 1

SECURITY REPORT:
This port has installed the following files which may act as network servers and may therefore pose a remote security risk to the system.
/usr/local/lib/libcurl.so.5

If there are vulnerabilities in these programs there may be a security
risk to the system. FreeBSD makes no guarantee about the security of
ports included in the Ports Collection. Please type 'make deinstall'
to deinstall the port if this is a concern.

For more information, and contact details about the security
status of this software, see the following webpage:
http://curl.haxx.se/

freebsd7# pkgdb -vu
--> Updating the pkgdb
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 12 packages found (-0 +2) .. done]

During installation, libssh2 was found to be a dependency, based on the customization we made. We can see the dependency using pkg_tree.

freebsd7# pkg_tree curl
curl-7.19.6_1
|__ libssh2-1.2,2
 __ ca_root_nss-3.11.9_2

If we want to create a package for Curl, we can use the 'make package' command.

freebsd7# make package
====> Building package for curl-7.19.6_1
Creating package /usr/ports/packages/All/curl-7.19.6_1.tbz
Registering depends: ca_root_nss-3.11.9_2 libssh2-1.2,2.
Creating bzip'd tar ball in '/usr/ports/packages/All/curl-7.19.6_1.tbz'

If we want to make the package and its dependencies, we use 'make package-recursive'.

freebsd7# make package-recursive
====> Generating temporary packing list
Creating package /usr/ports/packages/All/perl-5.8.9_3.tbz
Registering depends:
Registering conflicts: perl-5.6.* perl-5.10.* perl-threaded-5.10.*.
Creating bzip'd tar ball in '/usr/ports/packages/All/perl-5.8.9_3.tbz'
Creating package /usr/ports/packages/All/libssh2-1.2,2.tbz
Registering depends:
Creating bzip'd tar ball in '/usr/ports/packages/All/libssh2-1.2,2.tbz'
rmdir: /usr/ports/security/libssh2/work: Directory not empty
*** Error code 1 (ignored)
====> Generating temporary packing list
Creating package /usr/ports/packages/All/ca_root_nss-3.11.9_2.tbz
Registering depends:
Creating bzip'd tar ball in '/usr/ports/packages/All/ca_root_nss-3.11.9_2.tbz'

Note that using the 'make package-recursive' command means you don't have to run 'make install'. With FreeBSD, there is not a way to make a package but avoid installation.

freebsd7# ls /usr/ports/packages/All
ca_root_nss-3.11.9_2.tbz perl-5.8.9_3.tbz
curl-7.19.6_1.tbz portaudit-0.5.13.tbz
libssh2-1.2,2.tbz

Install Packages Built on One System to Another System

------------------------------------------------------

Once packages are built using the ports tree, you can install them on similar systems elsewhere. For example, we can copy packages from freebsd7 to another system and install them locally. In the following example we begin on host freebsd7S and will install packages built earlier in this article.

freebsd7S# uname -a
FreeBSD freebsd7S.taosecurity.com 7.2-STABLE FreeBSD 7.2-STABLE #2: Sat Aug 22 17:12:42 EDT 2009
root@freebsd7.localdomain:/usr/obj/usr/src/sys/FREEBSD7  i386

freebsd7S# mkdir -p /usr/ports/packages/All

freebsd7S# ls /usr/ports/packages/All
ca_root_nss-3.11.9_2.tbz perl-5.8.9_3.tbz
curl-7.19.6_1.tbz portaudit-0.5.13.tbz
libssh2-1.2,2.tbz
freebsd7S# sftp analyst@172.16.134.128
Connecting to 172.16.134.128...
Password:
sftp> cd /usr/ports/packages/All
sftp> ls
ca_root_nss-3.11.9_2.tbz  curl-7.19.6_1.tbz  libssh2-1.2,2.tbz
perl-5.8.9_3.tbz  portaudit-0.5.13.tbz
sftp> get ca_root_nss-3.11.9_2.tbz
Fetching /usr/ports/packages/All/ca_root_nss-3.11.9_2.tbz to ca_root_nss-3.11.9_2.tbz
100% 169KB  168.7KB/s  00:00
sftp> quit
freebsd7S# pkg_add ca_root_nss-3.11.9_2.tbz
freebsd7S# pkg_info | grep ca_root
ca_root_nss-3.11.9_2 The root certificate bundle from the Mozilla Project

So, the new package is installed, but what if we wanted to add a package with dependencies? For examples like this, we could mount the remote system's /usr/ports/packages directory using NFS, and add from there. The remote system here is freebsd7, or 172.16.134.128. I recommend making a read-only mount (via -o ro) so that the NFS client does not accidentally alter the server.

freebsd7S# mount -t nfs -o ro 172.16.134.128:/usr/ports/packages /usr/ports/packages
freebsd7S# cd /usr/ports/packages/All
freebsd7S# ls
ca_root_nss-3.11.9_2.tbz  perl-5.8.9_3.tbz
curl-7.19.6_1.tbz  portaudit-0.5.13.tbz
libssh2-1.2,2.tbz
freebsd7S# pkg_add -v curl-7.19.6_1.tbz
Requested space: 4525K bytes, free space: 455M bytes in /var/tmp/insttmp.0YHgDV
Package 'curl-7.19.6_1' depends on 'libssh2-1.2,2' with 'security/libssh2' origin.
Loading it from /usr/ports/packages/All/libssh2-1.2,2.tbz.
Requested space: 711K bytes, free space: 452M bytes in /var/tmp/insttmp.6EzFeD
extract: Package name is libssh2-1.2,2
extract: CWD to /usr/local
extract: /usr/local/include/libssh2.h
...edited...
extract: /usr/local/man/man3/libssh2_version.3.gz
extract: execute '/sbin/ldconfig -m /usr/local/lib'
extract: CWD to .
Running mtree for libssh2-1.2,2..
mtree -f +MTREE_DIRS -d -e -p /usr/local >/dev/null
Attempting to record package into /var/db/pkg/libssh2-1.2,2..
Package libssh2-1.2,2 registered in /var/db/pkg/libssh2-1.2,2
Package 'curl-7.19.6_1' depends on 'ca_root_nss-3.11.9_2' with 'security/ca_root_nss' origin.
- already installed.
extract: Package name is curl-7.19.6_1
extract: CWD to /usr/local
extract: /usr/local/share/examples/curl/threaded-ssl.c
extract: execute '/sbin/ldconfig -m /usr/local/lib'
extract: CWD to .
Running mtree for curl-7.19.6_1..
mtree -f +MTREE_DIRS -d -e -p /usr/local >/dev/null
Attempting to record package into /var/db/pkg/curl-7.19.6_1.
Trying to record dependency on package 'libssh2-1.2.2' with 'security/libssh2' origin.
Trying to record dependency on package 'ca_root_nss-3.11.9_2' with 'security/ca_root_nss' origin.
Package curl-7.19.6_1 registered in /var/db/pkg/curl-7.19.6_1

defs7S# cd

defs7S# umount /usr/ports/packages

Curl and its dependencies have now been installed over NFS from the freebsd7 system.

This example demonstrates how a centralized system (in this case, freebsd7) could serve as a local site ports tree and package repository, and client systems (like freebsd7S) could install packages from the local repository. In fact, the clients would not have to maintain their own ports trees.

Let's show how mounting /usr/ports from the package repository freebsd7 helps the client freebsdS learn what packages need updating.

First we know that our clients will need the sysutils/cmdwatch utility, so we make a package of it on our package builder freebsd7.

defs7S# cd /usr/ports/sysutils/cmdwatch
defs7S# make package
  => cmdwatch-0.2.0.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
  => Attempting to fetch from http://www.chruetertee.ch/files/download/.
  cmdwatch-0.2.0.tar.gz 100% of 11 kB 66 kBps
  ===> Extracting for cmdwatch-0.2.0_2
  => MD5 Checksum OK for cmdwatch-0.2.0.tar.gz.
  => SHA256 Checksum OK for cmdwatch-0.2.0.tar.gz.
  ===> Patching for cmdwatch-0.2.0_2
  ===> Applying FreeBSD patches for cmdwatch-0.2.0_2
  ===> Configuring for cmdwatch-0.2.0_2
  ===> Building for cmdwatch-0.2.0_2
  Making cmdwatch...getopt.c: In function '_getopt_internal':
    ...edited...
    done.
  ===> Installing for cmdwatch-0.2.0_2
  ===> Generating temporary packing list
  ===> Checking if sysutils/cmdwatch already installed
  Making cmdwatch... done.
  Installing cmdwatch
  ===> Compressing manual pages for cmdwatch-0.2.0_2
  ===> Registering installation for cmdwatch-0.2.0_2
  ===> Building package for cmdwatch-0.2.0_2
  Creating package /usr/ports/packages/All/cmdwatch-0.2.0_2.tbz
  Registering depends:
  Creating bzip'd tar ball in '/usr/ports/packages/All/cmdwatch-0.2.0_2.tbz'
defs7S# pkgdb -vu
    ---> Updating the pkgdb
    [Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 13 packages found (-0 +1) . done]

Now we turn to the package client, freebsdS, and mount the package builder's /usr/ports directory.

defs7S# mount -t nfs -o ro 172.16.134.128:/usr/ports /usr/ports

defs7S# mount
  /dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
  /dev/ad0s1f on /home (ufs, local, soft-updates)
  /dev/ad0s1g on /tmp (ufs, local, soft-updates)
  /dev/ad0s1d on /usr (ufs, local, soft-updates)
  /dev/ad0s1e on /var (ufs, local, soft-updates)
  172.16.134.128:/usr/ports on /usr/ports (nfs, read-only)

We can run 'pkgdb -vu' on freebsd7S because it stores the package database used by Portupgrade in /var/db/pkg on the local system.
freebsd7S# pkgdb -vu
----> Updating the pkgdb
[Rebuilding the pkgdb <format:bdb_btree> in /var/db/pkg ... - 12 packages found (-0 +12) ............ done]

Now we run `portversion -v` to see which packages need updating on the client. The client uses NFS to compare to the versions on the package builder.

freebsd7S# portversion -v
ca_root_nss-3.11.9_2        =  up-to-date with port
cmdwatch-0.2.0_1            <  needs updating (port has 0.2.0_2)
curl-7.19.6_1               =  up-to-date with port
cvsup-without-gui-16.1h_4   =  up-to-date with port
db41-4.1.25_4               =  up-to-date with port
libssh2-1.2,2               =  up-to-date with port
perl-5.8.9.3                =  up-to-date with port
pkg_cutleaves-20090810      =  up-to-date with port
portupgrade-2.4.6_3,2       =  up-to-date with port
ruby-1.8.7.160_4,1          =  up-to-date with port
ruby18-bdb-0.6.5_1          =  up-to-date with port
screen-4.0.3_5              <  needs updating (port has 4.0.3_6)

We see that cmdwatch and screen need updating. On the client we invoke Portupgrade using the `-vaPP` switches.

freebsd7S# portupgrade -vaPP
----> Session started at: Tue, 25 Aug 2009 14:56:56 -0400
----> Checking for the latest package of 'sysutils/screen'
----> Fetching the package(s) for 'screen-4.0.3_6' (sysutils/screen)
----> Fetching screen-4.0.3_6
++ Will try the following sites in the order named:
----> Invoking a command: /usr/bin/fetch -o '/var/tmp/portupgrade69eLJ2VS/screen-4.0.3_6.tbz' 'ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen-4.0.3_6.tbz'
fetch: ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen-4.0.3_6.tbz: File unavailable (e.g., file not found, no access)
** The command returned a non-zero exit status: 1
** Failed to fetch ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen-4.0.3_6.tbz
----> Invoking a command: /usr/bin/fetch -o '/var/tmp/portupgrade69eLJ2VS/screen-4.0.3_6.tgz' 'ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen-4.0.3_6.tgz'
fetch: ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen-4.0.3_6.tgz: File unavailable (e.g., file not found, no access)
** The command returned a non-zero exit status: 1
** Failed to fetch ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen-4.0.3_6.tgz
----> Listing the results (+:done / -:ignored / *:skipped / !:failed)
  ! screen-4.0.3_6 (fetch error)
----> Packages processed: 0 done, 0 ignored, 0 skipped and 1 failed
----> Fetching the latest package(s) for 'screen' (sysutils/screen)
----> Fetching screen
++ Will try the following sites in the order named:
----> Invoking a command: /usr/bin/fetch -o '/var/tmp/portupgradeKmGTSv48/screen.tbz' 'ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen.tbz'
fetch: ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen.tbz: File unavailable (e.g., file not found, no access)
** The command returned a non-zero exit status: 1
** Failed to fetch ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen.tbz
----> Invoking a command: /usr/bin/fetch -o '/var/tmp/portupgradeKmGTSv48/screen.tgz' 'ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen.tgz'
fetch: ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen.tgz: File unavailable (e.g., file not found, no access)
** The command returned a non-zero exit status: 1
** Failed to fetch ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-7-stable/All/screen.tgz
----> Listing the results (+:done / -:ignored / *:skipped / !:failed)
We see that Portupgrade did not find a screen package on the package builder (in /usr/ports/packages/All), and when it failed it tried to find a package on a remote FreeBSD server. That failed too, because the FreeBSD project does not build screen packages. The project recommends users build their own packages for screen. However, the newest version of cmdwatch was installed, using the package /usr/ports/packages/All/cmdwatch-0.2.0_2.tbz.

Installing Screen Using a Remote FreeBSD Ports Tree

What do we do about screen? It turns out that we can work around this problem.

First, remove the old package using any of the methods demonstrated in this article.

Next, mount the package builder's ports directory.

freebsd7S# mount -t nfs -o ro 172.16.134.128:/usr/ports /usr/ports

If /usr/ports doesn't exist on the client, create it.
Now cd to /usr/ports/sysutils/screen and run the following.

```bash
freebsd7S# cd /usr/ports/sysutils/screen
freebsd7S# make WRKDIRPREFIX=/tmp
freebsd7S# make install WRKDIRPREFIX=/tmp
```

When done, the new version of screen will be built, using the remote package builder's ports tree but by installing source code on the local system.

Reading /usr/ports/UPDATING

---------------------------

It is important to read /usr/ports/UPDATING before invoking Portupgrade. We have not done so yet because these examples have been fairly simple. However, there may be information in /usr/ports/UPDATING that could recommend different actions depending on the ports of interest. From now on, consult /usr/ports/UPDATING after you upgrade your ports tree and before you invoke Portupgrade.

My Common Package Update Process

--------------------------------

So what is the end result of this process? For individual systems, I recommend the following process. This assumes Portupgrade is installed, and that I rely on packages produced by the FreeBSD project. I also assume that Portaudit is running automatically every day already.

1. setenv HTTP_PROXY [insert your proxy]
2. setenv PACKAGESITE ftp://ftp[X].freebsd.org//pub/FreeBSD/ports/i386/packages-7-stable/Latest/ where [X] is the number of a FreeBSD FTP server near you.
3. portsnap fetch
4. portsnap update
5. portsdb -u
6. pkgdb -vu
7. portversion -v -l "<"
8. Read /usr/ports/UPDATING to see if any special instructions apply to packages of interest.
9. portupgrade -vaPP
10. portversion -v -l "<"

Conclusion

--------

I hope this article has helped you understand the different ways to keep FreeBSD applications up-to-date. It is by no means comprehensive, but by following it you hopefully can judge the different ways to keep your applications current.

Revision History

25 August 2009: Draft FreeBSD 7.x version posted
24 December 2004: First publication
21 April 2005: Minor typo corrections

Copyright 2009 Richard Bejtlich and TaoSecurity.