

# BSD HACKS

*100 Industrial-Strength  
Tips & Tools*



O'REILLY®

*Dru Lavigne*

HACK  
#82

## Build a Port Without the Ports Tree

While the ports tree is one of the most useful FreeBSD directory structures, you may have systems where it's not appropriate to maintain the entire ports structure.

On some of your systems, disk space may be an issue. The ports tree tarball itself is a 21 MB download. Once untarred, it will occupy around 500 MB of disk space. That space will continue to grow as you install ports since, by default, source files download into */usr/ports/distfiles*.

Does this mean that installing packages is your only alternative? Packages are convenient, but since they are precompiled, you don't have the option of providing your own make arguments to optimize the install for your environment.

One alternative is the anonymous CVS system. Even a minimal install of FreeBSD includes the `cvs` command. This allows you to check out only the particular port skeleton you need. You'll still have the convenience of the ports collection without actually having to install it.

### Connecting to Anonymous CVS

The first time you use `cvs`, create an empty CVS password file, as CVS will complain if this file is missing:

```
# touch ~root/.cvspass
```

Then, ensure your present working directory is */usr*:

```
# cd /usr
```



When using `cvs` to maintain your ports, be sure you are in */usr*. `cvs` downloads the requested files to your current working directory and will overwrite any files of the same name.

Then, use the `cvs login` command to connect to a CVS server. There are five FreeBSD anonymous CVS servers; see the Handbook reference at the end of this hack for their names and passwords. Use the `setenv` command to specify the server to log into:

```
# setenv CVSROOT :pserver:anoncvs@anoncvs.at.FreeBSD.org/home/ncvs
# cvs login
Logging in to :pserver:anoncvs@anoncvs.at.freebsd.org:2401/home/ncvs
CVS password: anoncvs
#
```

Once you've successfully logged in, you'll receive your normal prompt back. You'll remain connected to the CVS server until you explicitly log off. In the

meantime, you now have the ability to issue commands either on the CVS server or on your own system.

## Checking Out Port Skeletons

Let's assume you have a minimum install and don't have an existing `/usr/ports` directory structure. To install a port, you need the *Mk* and *Templates* directories as well as the port's *Makefile*.

Use the `cvs checkout` command to retrieve the necessary files from the CVS server:

```
# cvs checkout -A -P -l ports/Mk
cvs server: Updating ports/Mk
U ports/Mk/bsd.emacs.mk
U ports/Mk/bsd.gnome.mk
U ports/Mk/bsd.gnustep.mk
U ports/Mk/bsd.java.mk
U ports/Mk/bsd.kde.mk
U ports/Mk/bsd.openssl.mk
U ports/Mk/bsd.port.mk
U ports/Mk/bsd.port.post.mk
U ports/Mk/bsd.port.pre.mk
U ports/Mk/bsd.port.subdir.mk
U ports/Mk/bsd.python.mk
U ports/Mk/bsd.ruby.mk
U ports/Mk/bsd.sites.mk

# cvs checkout -A -P -l ports/Templates
cvs server: Updating ports/Templates
U ports/Templates/README.category
U ports/Templates/README.port
U ports/Templates/README.top
U ports/Templates/config.guess
U ports/Templates/config.sub
#
```

Since you're in the `/usr` directory, `cvs` will create `/usr/ports` for you and will populate the *Mk* and *Templates* subdirectories with their sets of files. It's interesting to note how little disk space this bare-minimum ports tree requires:

```
# du -h /usr/ports | tail -n1
418K  ports
```

That's a pretty big difference from 500 MB!

## Finding a Port and Its Dependencies

Next, decide which port you'd like to install. The only disadvantage to not having the entire ports structure is that you need an alternate method of dis-

covering the name of the port you'd like to install. For example, in order to install lynx, I need to know that it is in the *www* subdirectory and that there are three different versions of lynx to choose from. The easiest way to discover this information is to use the search utility at <http://www.freshports.org>.

Once you find the port you're looking for, it will indicate the name of its directory. In my example, *lynx-2.8.5d17* lives in *www/lynx-current*.

Now it's a simple matter of checking out that port's skeleton:

```
# cvs checkout -A -P -l ports/www/lynx-current
cvs server: Updating ports/www/lynx-current
U ports/www/lynx-current/Makefile
U ports/www/lynx-current/distinfo
U ports/www/lynx-current/pkg-descr
U ports/www/lynx-current/pkg-plist
```

Next, check the port's *Makefile* to see if there are any dependencies:

```
# grep DEPENDS /usr/ports/www/lynx-current/Makefile
LIB_DEPENDS=    intl.5:${PORTSDIR}/devel/gettext
```

As it stands right now, this port will not install, as I don't have the ports skeleton for the dependency *devel/gettext*. So, I'll download that port skeleton and double-check that *that* port doesn't have any dependencies:

```
# cvs checkout -A -P -l ports/devel/gettext
<snip output>
# grep DEPENDS /usr/ports/devel/gettext/Makefile
#
```

Okay, it looks like all dependencies are there. I'm ready to build the port:

```
# cd /usr/ports/www/lynx-current
# make install clean
```



If disk space is an issue, instead use `make install distclean`, which will delete the source from */usr/ports/distfiles* once the build successfully completes.

That's it. As long as you remember to look for dependencies before you issue your `make install` command, your minimal ports structure should work as flawlessly as the full ports collection.

Don't forget to use `cvs logout` when you're finished retrieving the files you need from the CVS server.

## See Also

- `man cvs`

- The AnonCVS section of the FreeBSD Handbook, which includes the names of the BSD CVS servers ([http://www.freebsd.org/doc/en\\_US.ISO8859-1/books/handbook/anoncv.html](http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/anoncv.html))