Upgrading FreeBSD Ports

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
	2014-04-17		WB

Contents

Last updated 2014-04-17

Available in HTML or PDF. Links to all my articles here. Created with AsciiDoc.

Keeping ported applications up to date on FreeBSD.

1 Introduction

Keeping ported applications up to date is a bit trickier than just installing them in the first place. The process I use may be useful as an example, so here it is.

2 Updating Ports With Dependencies

If a new version of a program comes out, why can't you just deinstall the old version and install the new one?

For a high-level program like, say, Firefox, you can. It's the lower-level programs that are trickier. For example, consider the jpeg graphics library. New versions of the library may work differently or have different interfaces than the old. Firefox, expecting the old version of jpeg, may not be able to use the new version correctly, or at all.

Simple enough, you say, I'll just install the new version of Firefox also. Yes... except it might not be just Firefox, and there might be a chain of dependencies that have to be updated in a particular order. For example, Firefox depends a bunch of things that depend on the tiff library, which in turn depends on jpeg. Some of these things have to be rebuilt when the jpeg code changes, some do not. Dependency rebuilds have to be done in the correct order; jpeg has to be rebuilt before tiff can be rebuilt, and so on up the chain.

The complexity of figuring out what has to be rebuilt and in what order can be simplified by having a program work it all out for you.

3 Software For Updating Ports

Here, we'll use `portsnap(8)` to update the ports collection and `portmaster`, found in */usr/ports/ports-mgmt/portmaster*, to upgrade ports.

`portsnap` is the faster, simpler, and preferred way to update the ports collection, so that choice is easy.

There are a variety of port upgrade programs. `portmaster(8)` is used here as the new standard, and the most actively supported.

4 Updating The Ports Collection

The Ports Collection is a list of build instructions located in */usr/ports*. Download and initialize it with

```
# portsnap fetch extract
```

That's only required the very first time. Afterward, updates are downloaded and updated with

```
# portsnap fetch update
```

5 Checking For Updates

Now the installed applications can be compared to the versions in the ports collection.

```
# portmaster -L --index-only| egrep '(ew|ort) version|total install'
        ===>>> New version available: bash-4.1.10
===>>> 674 total installed ports
```

This example shows a single port that needs updating.

6 Automating The Update

It's easy to automate the ports collection update with a small shell script:

```
#!/bin/sh
/usr/sbin/portsnap fetch update && \
/usr/local/sbin/portmaster -L --index-only | egrep '(ew|ort) version|total install'
```

7 Checking *UPDATING*

/usr/ports/UPDATING contains notes about updating ports that require special procedures. Always, yes *always*, read this file for special notes before updating ports.

Only entries that have been added since the last time you updated ports are relevant. That date can be determined from the package database:

```
echo -n "Last update: "
date -r `pkg query %t | sort | tail -n1` "+%Y%m%d"
```

This can be added to the end of the update script shown above.

8 Updating Installed Applications

Use portmaster to update the outdated port from the example above:

```
# portmaster bash
```

`portmaster` first checks that everything the port depends on is present. Then it rebuilds that port, and finally rebuilds all ports that depend on it.

If there is more than one outdated port, just list them all on the command line. `portmaster` will work out the dependencies and build order.

9 Odd Situations

Sometimes it's helpful to have `portmaster` figure out what needs to be upgraded and in what order, but not actually do it. This can be useful if something needs to be done manually during a step, or you want to break up an upgrade of a lot of ports into smaller pieces.

```
# portmaster -na
===>>> Starting check of installed ports for available updates
===>>> Launching child to update bash-4.1.9 to bash-4.1.10

===>>> Port directory: /usr/ports/shells/bash

===>>> Gathering dependency list for shells/bash from ports
===>>> Initial dependency check complete for shells/bash
===>>> Returning to update check of installed ports


===>>> Starting build for for ports that need updating <<<===

===>>> Launching child to install shells/bash
===>>> Returning to update check of installed ports

===>>> Update check of installed ports complete
```

In this case, all it would do is rebuild the single bash port.

10 Other Notes

Lots of people try the *-a* (all) option with port upgrade tools. If there are lots of updates, or any with special steps required in */usr/ports/UPDATING*, that approach can fail. The *-n* option can be useful here also, producing a list of ports in the order they should be rebuilt.

```
# portmaster -na > /tmp/portorder.txt
```

`portmaster` shows any configuration screens that require options to be set before it begins the build. That lets you set all the options first rather than at various points during the build.

11 Checking For Missing Libraries

Dominic Fandrey's *sysutils/bsdadminscripts* port has a particularly useful script called `pkg_libchk`. It checks for programs that link to missing libraries, a problem that often happens when people don't read */usr/ports/UPDATING* regularly. After updating ports, run

```
pkg_libchk -qo
```

Any ports listed are trying to use old libraries and need to be rebuilt. If more than one is listed, give the entire list to `portmaster` so it can rebuild them in the correct order.

12 Reinstalling All Ports

Occasionally a system upgrade, like upgrading from FreeBSD 7 to FreeBSD 8, requires a forced reinstall of all ports. The `portmaster` man page shows a procedure for doing that. Search for the keyword "reinstallation" in that page:

```
# man portmaster | less -p reinstallation
```

If the process of reinstalling everything fails, `portmaster` shows a command line that can be used to restart where it failed. It's useful to run `script(1)` before starting the reinstallation process to avoid having to retype that long command.

13 That’s It

Hopefully we’ve upgraded your view of port upgrading, maybe from "Horribly Painful" to "Necessary Evil". Or possibly from merely "Annoying" to "Minor Inconvenience".