How to do (all) install packages binary upgrade FreeBSD 7.2 to FreeBSD 9.0 RELEASE

Author : admin

In last article I've writen how FreeBSD version to version upgrade can be done using, as it is (mostly) mandatory to complete a packages binary upgrade right after the kernel and world update, in this article I will explain *how to do BSD binary package upgrade.*

Once the BSD boots with a new after succesful:

```
# freebsd-update upgrade -r 9.0-RELEASE && freebsd-update install
```

1. Upgrade FreeBSD port tree structure (skele) with latest BSD available,br />
It is necessery either to rebuild all installed packages on System or do a binary upgrade, this is because some of the installed packages might already be linked to older versions of FreeBSD 7.2 [in my case] - (or whatever older version) world you have.

1. Update BSD port tree skele to the latest published on FreeBSD download servers

It is generally wise idea to *upgrade your BSD port structure* immediately after BSD world and kernel upgrade.
For convenience you can use following ports-supfile (configured to fetch latest FreeBSD) ports directory structure:

Download it in */etc* and run **csup**:

```
freebsd# cd /etc/
freebsd# wget -q http://www.pc-freak.net/files/ports-supfile
freebsd# csup -g -L 2 /etc/ports-supfile
Parsing supfile "/etc/ports-supfile"
Connecting to cvsup4.de.freebsd.org
Connected to 212.12.50.227
```

```
Server software version: SNAP_16_1h
MD5 authentication started
MD5 authentication successful
Negotiating file attribute support
Exchanging collection information
Establishing multiplexed-mode data connection
Running
Updating collection ports-all/cvs
Edit ports/UPDATING
....
....
```

The port structure upgrade takes me on my *Pentium 1.8 Ghz with 1GB ram*with a 20 Mbit connectivity to the internet about 15 minutes; It could vary depending on PC hardware and Internet connectivity as well as to when for a last time the port tree was upgraded (how much new data csup has to fetch to hdd).

2. Re-build ports Index file /usr/ports/INDEX*

Onwards the ports index file needs to be rebuild;

```
freebsd# make index
Generating INDEX-9 - please wait..
....
```

Here my *make index* failed with an error:

```
*** Error code 1
1 error
********************************************************************
Before reporting this error, verify that you are running a supported
version of FreeBSD (see http://www.FreeBSD.org/ports/) and that you
have a complete and up-to-date ports collection. (INDEX builds are
not supported with partial or out-of-date ports collections -- in
particular, if you are using cvsup, you must cvsup the "ports-all"
collection, and have no "refuse" files.) If that is the case, then
report the failure to ports@FreeBSD.org together with relevant
details of your ports configuration (including FreeBSD version,
your architecture, your environment, and your /etc/make.conf
settings, especially compiler flags and WITH/WITHOUT settings).
Note: the latest pre-generated version of INDEX may be fetched
automatically with "make fetchindex".
********************************************************************
*** Error code 1
```

Cause this error, I had to update the index with *portsnap* (get already pre-generated one by another person).

portsnap has other advantages too - it doesn't take so long time as well as load CPU less than manually building it; even better it takes shorter time to complete. This is why for (ports index updates) on productive servers it might be (probably) better to use portsnap:

```
freebsd# make fetchindex
/usr/ports/INDEX-9.bz2 100% of 1622 kB 203 kBps
freebsd# ls -al INDEX-*
-rw-r--r-- 1 root wheel 24216397 Mar 14 2011 INDEX-7
-rw-r--r-- 1 root wheel 36006912 Feb 8 2010 INDEX-7.db
-rw-r--r-- 1 root wheel 26526196 Oct 2 14:45 INDEX-9
freebsd# rm -f INDEX-7 INDEX-7.db
```

3. Do a binary package upgrade of every packages present on previous FreeBSD 7.2 install with *portupgrade*

It is generally also a good idea to create a dump of all system installed packages with *pkg_info*, before doing anything else:

```
freebsd# pkg_info -qoa | sort > /root/packagelist
```

Since my system is quite, slow I cannot afford to recompile all from BSD ports, thus I used the "old fashioned" **portupgrade** to complete the binary upgrade:

For "conservative" BSD users it might be better to use *portupgrade*, to do upgrade all installed packages with it run:

```
freebsd# rm /var/db/pkg/pkgdb.db && pkgdb -Ffuv && portupgrade -afpRr -m BATCH=yes
....
```

I got during above portupgrade errors like:

Stale dependency: brasero-2.26.0_1 --> hal-0.5.11_23 -- manually run 'pkgdb -F' to fix, or specify -O to force.

In order to solve this errors, I had to exec:

```
freebsd# pkgdb -Fa
```

Then after each time fixing the pkgdb database run:

```
freebsd# pkgdb -Ffuv && portupgrade -afpRr -m BATCH=yes
```

Besides *portupgrade*, the upgrade can be done using other metdhology;

a) Using a few lines shell of bash scripting loop and *pkg_info* and *pkg_add*

b) Using *pkgng*

4. Binary all package upgrade using pkg_info, pkg_delete, pkg_add with a while shell loop

An alternative **FreeBSD installed packages upgrade to latest** is with standard BSD tools combined with few lines of shell of bash *while* loop.

This whole update approach is [thoroughfully explained in this BSD forum thread].

The update method pointed by the forum can be done by issuing (preferrably in bash shell root console):

```
root # pkg_info -qoa > /root/packagelist
root # pkg_delete -a -f
root # rm -r -f /boot/modules /usr/local /var/db/pkg
root # while read PKG; do pkg_add -r $( basename ${PKG} ); done
```

The problem with this update approach is if some packages names are changed, some packages might not be able to be found by and (respectively) will fail to fetch and install with *pkg_add -r*. Thus I personally don't recommend using this .update method as it is unsafe.

5. Completing packages BSD binary upgrade with pkgng (PKG Next Generation) tool

Install *pkgng* from respective port, e.g.:

```
freebsd# cd cd /usr/ports/ports-mgmt/pkg
....
....
/usr/local/man/man8/pkg-remove.8.gz -> /usr/local/man/man8/pkg-delete.8.gz
/usr/local/man/man8/pkg-static.8.gz -> /usr/local/man/man8/pkg.8.gz
===> pkg-static (install)
install -o root -g wheel -m 555 pkg-static /usr/local/sbin
If you are upgrading from the old package format, first run:
# pkg2ng
===> Running ldconfig
/sbin/ldconfig -m /usr/local/lib
===> Registering installation for pkg-1.0.1
===> Cleaning for pkg-1.0.1
```

```
freebsd# make install clean
....
```

Right after installing you can use it and upgrade packs:

```
freebsd# cp -rpf /usr/local/etc/pkg.conf.sample /usr/local/etc/pkg.conf
freebsd# pkg update
Updating repository catalogue
repo.txz 100% 13MB 771.3KB/s 1.3MB/s 00:17
freebsd# pkg upgrade -fy
...
freebsd# pkg-static update
....
freebsd# pkg-static upgrade -fy
```

,,,
,,,,

P.S. I tried the *pkgng* for my first time, for some weird reason I was unable to upgrade my packages with it on 1 host, though literally followed update instructions on Full binary upgrade with pkgng 1.0-beta7 - Bapt tutorial . If someone knows of a better tutorial how update can be done using *pkgng* drop me a comment or point me to a tutorial.
The overall all packages update experience for me wasn't so nice as it took me too much of time to upgrade. I like BSD stability but when it comes to updates its a time eater ...