

Things I Think Are Broken And That I'd Like To Fix

Johnny Lam
jlam@NetBSD.org

Some problems

- Directory-handling
- `BUILDLINK_RECOMMENDED`
- Dependency checking
- `LOCALBASE = /usr`
- [bsd.pkg.mk](#) is hard to read
- Bootstrap

Directory handling

- `*.{pkg,x11}.dist` seem unwieldy
 - Adding a directory to every single `*.dist` file when a new locale directory pops up seems wrong to me
 - Move to `NO_MTREE=yes` and setting `INSTALLATION_DIRS` appropriately
 - Need to do this anyway for `pkgviews`
- `*-dirs` pkgs shouldn't be needed
 - These packages exist solely to provide a directory hierarchy
 - Maintenance of these packages seems like more busywork
- Teach `pkg_delete(1)` to automatically remove empty directories
 - Some packages share directories and want empty directories to exist. Need to account for this somehow.

BUILDLINK_RECOMMENDED

- Core problem is user not wanting to update a dependency.
 - Don't want to rebuild huge web of packages.
 - User wants to override `BUILDLINK_DEPENDS.<pkg>` with the installed version.
- Create a variable `IGNORE_DEPENDS` that contains a list of packages for which we ignore the dependency check and just use the installed package
 - e.g. `IGNORE_DEPENDS = tiff png`
- For each package listed in `IGNORE_DEPENDS`, override `BUILDLINK_DEPENDS.<pkg>` to accept the installed package
 - Take advantage of C-shell-style globs for package patterns
 - e.g. If `tiff-3.5.4` is installed on the system, then set `BUILDLINK_DEPENDS.tiff = {tiff>=3.6.1,tiff>=3.5.4}`
- This deprecates need for `BUILDLINK_RECOMMENDED`
- Easy to implement!

Dependency checking

- X depends on Y, Z. Y depends on Z. X requires a newer version of Z than Y.
 - Currently builds X but links in two different versions of Y's libraries, which is bad.
- We should note this as a error before building
- Most tricky when dependency on Z may be satisfied by a built-in pkg
 - We need some way to note the dependency on the builtin pkg.

LOCALBASE = /usr

- Hi, jreed@netbsd.org!
- How to hide /usr/include and /usr/lib from configure scripts?
 - Use `-nostdinc` and `-nostdlib` for GCC
 - Possible problems with C++ headers on older GCCs?
 - Similar options for other compilers?
 - Create “sys-links” pkg (similar to x11-links) that contains just the headers and libs in /usr/include and /usr/lib.
 - Needs to be specific to each OS + version
 - Not onerous to maintain since files per OS + version is pretty stable

`bsd.pkg.mk` is hard to read

- Gargantuan `bsd.pkg.mk` is hard to read, understand and maintain
 - Split out “utility” targets & definitions into separate files
 - Split out targets & definitions specific to each build phase into own files: `bsd.fetch.mk`, `bsd.extract.mk`, `bsd.patch.mk`, etc.

Bootstrap

- Still can't really bootstrap very well from scratch
- Bootstrap packages must not use pkgsrc infrastructure to build
 - Use shell scripts in `/usr/pkgsrc/bootstrap`
- Put bootstrap versions of packages into `${LOCALBASE}/bootstrap`
 - Won't conflict with same packages built using “normal” pkgsrc
 - Teach infrastructure to fall back to using bootstrap versions of `pkg_add(1)`, etc.
- Should be done before `pkgsrc-2004Q2` is branched