pkgviews:
Package Views Implementation

Johnny Lam
jlam@NetBSD.org
Outline

• pkg_install(1) modifications
• bsd.pkg.mk
• PKG_SYSCONFDIR
• bsd.pkg.install.mk
• buildlink3
• Unresolved design issues
A "pkgviews" package is one that is installed using pkgviews.

An "overwrite" package is one that hasn't.

A pkgviews package is installed into /usr/pkg/packages in its "depot" directory, e.g. /usr/pkg/packages/pth-2.0.0nb1

A package "instance" in a view describes a pkgviews package symlinked into a view.
pkg_install(1) modifications

- `-K <pkg_dbdir>` to specify package database directory on the command line. Same as what you'd set in `PKG_DBDIR`.
  - `/var/db/pkg` for null view
  - `/usr/pkg/<view>/pkgdb` otherwise

- `pkg_view(1)` and `linkfarm(1)` manage package instances and views.
  - `linkfarm(1)` was inspired by GNU `stow(1)` and symlinks everything in the depot directory into `${LOCALBASE}`
  - `pkg_view(1)` wraps `linkfarm(1)` and manipulates package metadata files
pkg_install(1) modifications (cont.)

- /usr/pkg/packages/<pkg>/+VIEW
  - Lists all of the views to which <pkg> has been added
    - e.g. /var/db/pkg/<pkg>
    - Note! This is a gotcha when moving the package database directory for the null view – can't just pax(1) it somewhere else, as you'd also need to update all +VIEWS files within ${DEPOTBASE}
      - Updated by pkg_view(1)
      - Used by pkg_delete(1) to figure out if it's safe to delete a depoted package (if it's still in a view, we can't delete it)

- /var/db/pkg/<pkg>/+DEPOT
  - Lists the depot directory of <pkg>
  - Used by pkg_delete(1) to update the +VIEWS in the depot directory when removing a package instance from a view
• **DEPOTBASE, DEPOT_SUBDIR**
  
  - `${DEPOTBASE}` always lives in `${LOCALBASE}` and defaults to `/usr/pkg/packages`. Modified by setting `DEPOT_SUBDIR`, which defaults to "packages"

• **PKG_INSTALLATION_TYPES, PKG_INSTALLATION_PREFS**

• **Dynamic PLISTs**
  
  - Every file in `${PREFIX}` is listed in the PLIST
  - All directories in `${PREFIX}` are added to the PLIST but are removed using `@unexec rmdir ... || true`
    
    • Could also make `@dirrm` fail silently if removing the directory fails
PKG_SYSCONFDIR

- If `PKG_SYSCONFBASE` is `${PREFIX}`, then do nothing special
  - Config files are symlinked from `${PREFIX}/etc` into `/usr/pkg/etc`
  - Real config files still live in `${PREFIX}/etc` *(Important detail for admins!)*
    - e.g. Edit `/usr/pkg/packages/samba-3.0.2/etc/samba/smb.conf`, not `/usr/pkg/etc/samba/smb.conf`
  - When deleting the package, the depot directory won't be removed if the config files were altered and preserved *(Important detail for admins!)*
- If `PKG_SYSCONFBASE` is `/etc`, then config files live in `/etc/packages/<pkg>` and symlinked into `/etc`
  - Just like adding a package instance to a view, but for the config files for that package.
bsd.pkg.install.mk

- **VIEW-INSTALL**
  - Executed when after adding an instance to a view.

- **VIEW-DEINSTALL**
  - Executed when before deleting an instance from a view.

- Contains actions that are view-specific
  - Update `/etc/shells` when adding/deleting a shell package to a view.
  - Symlink config files correctly in the `PKG_SYSCONFBASE=/etc` case
  - Update info file entries in `/usr/pkg/<view>/info/dir` when adding/deleting an instance from a view.

- For overwrite packages, `VIEW-INSTALL` and `VIEW-DEINSTALL` are invoked as part of `POST-INSTALL` and `DEINSTALL`. 
buildlink3

- `BUILDLINK_PREFIX.<pkg>` is the depot directory for `<pkg>`
- `BUILDLINK_IS_DEPOT.<pkg>` is “yes” if `<pkg>` is installed in a depot directory.
- Use `-I<depot_dir>/include`, `-L<depot_dir>/lib`, and `-R<depot_dir>/lib` instead of symlinking files into the buildlink directory.
- Libtool archives
  - Still need to create libtool archives in the buildlink directory that refer only within the buildlink directory or else libtool breaks.
- `/usr/pkg/lib` is in the rpath
  - Allows binary packages with dependencies like `foo>=1.0` to still work if `foo` is updated to 1.1, as long as it's in the null view.
- If building an overwrite package, change references to `${DEPOTBASE}/<dep_pkg>` into `${LOCALBASE}`
  - Overwrite packages think they're just depending on other overwrite packages.
Unresolved issues

- Packages that can be extended with module packages
- Fully mix pkgviews and overwrite packages
- `<whine> I don't want a symlink farm! </whine>`
Extensible packages

- e.g. PAM, PHP, Perl, Apache, Cyrus-SASL, etc.
- Main package looks for modules within its own depot directory
  - /usr/pkg/packages/PAM-0.77/lib/security/pam_*.so
- Module packages install into their own depot directories and are added to the null view.
  - /usr/pkg/packages/pam-ldap-150nb2/lib/security/pam_ldap.so
- **Problem!** Main package doesn't find the module.
Extensible packages: package-specific views

• Teach main package to look for its modules in a particular directory under ${VIEWBASE}
  – VIEWBASE is /usr/pkg/${DEFAULT_VIEW.<pkg>}
  – DEFAULT_VIEW.<pkg> defaults to the null view
• Add module packages to the default view of the main package.
• E.g. DEFAULT_VIEW.PAM = no_bsd_auth
  – PAM-aware applications look for PAM modules in /usr/pkg/no_bsd_auth/lib/security/
  – pam-ldap is added to the not_bsd_auth view.
• This is the currently implemented solution.
• Problems:
  – Hardcoded paths across many different packages (Yuk!)
  – Module packages must be present in DEFAULT_VIEW.< pkg> or they won't be found at all – mandatory views (Yuk!)
Extensible packages: main depot directory as a pseudo-view

- Pretend main depot directory is a view and add an instance of the module to that view.
  - `pkg_view -V /usr/pkg/packages -v PAM-0.77 add pam-ldap-150`
- Main package finds modules without any changes
- Don't need mandatory views or hardcoded shared directories.
- Problems:
  - Breaks idea that depot directories only belong to one package.
    - Who cares?
  - Symlinks to the module instances symlinks in the main package's depot directory will also be created, but aren't listed in the `+CONTENTS` file.
    - Can't use `pkg_delete(1)` to delete the main package instance from the view.
    - Can use `pkg_view(1)` to delete the instance correctly (`pkg_view(1)` doesn't consult the `+CONTENTS` file)
Mixing pkgviews and overwrite packages

- Current situation
  - Overwrite packages can depend on other overwrite packages.
  - Overwrite packages can depend on instances of pkgviews packages that have been added to the null view.
  - Pkgviews packages can depend on other pkgviews packages.
  - Pkgviews packages cannot depend on overwrite packages.
- If the last case can be made to work, then we can fully mix using either type of package.
  - Perfect migration scenario!
  - Should really try to solve this before taking pkgviews to mainstream to avoid a flag day for users
Mixing pkgviews and overwrite packages (cont.)

- When building a pkgviews package against an overwrite package dependency, add the depot directory for a pkgviews version of that dependency to the rpath.
  - e.g. -R/usr/pkg/packages/png-1.25nb4/lib
  - This allows future replacement of that dependency with a pkgviews package
Mixing pkgviews and overwrite packages (cont.)

- Tricky issues with dependency checking
  - **Solution 1**: Teach pkg_install(1) tools and bsd.pkg.mk to check in ${DEPOTBASE} then fall back to /var/db/pkg for packages to satisfy dependencies
    - e.g. Does /usr/pkg/packages/png-1.25nb4 exist? What about /var/db/pkg/png-1.25nb4?
  - **Solution 2**: Create dummy pkgviews package in ${DEPOTBASE} for an overwrite package if used as a dependency.
    - Modify pkgviews packages to install files into <depot_dir>/files
    - <depot_dir>/files is a directory: it's a pkgviews package
    - <depot_dir>/files is a symlink to /usr/pkg: it's a dummy package for an overwrite package
    - This seems hackish
<whine>I don't want a symlink farm!</whine>

- linkfarm(1) creates a symlink for every single file in the depot directory
- Teach linkfarm(1) to do tree-folding, a la GNU stow(1)
  - Uses symlinks more efficiently — only symlink as far down into a directory tree as absolutely needed.
  - SMOP (really!)
  - Can't use pkg_delete(1) to delete an instance.
  - Can use pkg_view(1) to do it.
<whine>I don't want a symlink farm!</whine>

- tv@NetBSD.org: For packages in only a single view, teach pkg_view(1) to move all of the files directly into the view and maintain a small back-link.
  - Store all of the package's files deeper within the depot directory, e.g /usr/pkg/packages/perl-5.8.4/files
  - Move /usr/pkg/packages/perl-5.8.4/files/* into /usr/pkg/<view>
  - Change /usr/packages/perl-5.8.4/files into a symlink pointing to /usr/pkg/<view>
    - Needed since the package thinks it lives in the depot directory
  - Only one symlink!
- Problems:
  - buildlink3 must use symlinks again
    - No big deal...we've been doing that for years
  - Higher probability of shooting self in foot if there are errors.
User's Guide

- /usr/pkgsrc/mk/buildlink3/PKGVIEWS_UG
- Describes how to set up your system to use package views, and walks through installing packages and manipulating views.
- Please read!
- Please test!