HOMEBREW: PART I

Package manager terms explained

INTRO

- In this tutorial I will explain often used Homebrew package manager terms, such as formula, bottle, cellar, keg, cask and tap.

PRESENTATION

- This presentation can be found at:
  https://www.mobilefish.com/download/homebrew/brew_part1.pdf

- All my Homebrew videos and presentations can be found at:
  https://www.mobilefish.com/developer/homebrew/homebrew.html

HOMEBREW

- Homebrew is a free and open-source package management system for Unix-like operating systems.

- Homebrew is popular on macOS.

- To install Homebrew:
  - Goto: https://brew.sh/ and copy the online installation instruction.
  - Paste the instruction in your terminal.

- The package will be installed in the cellar:
  `/usr/local/Cellar`

HOMEBREW

- The installed packages are symlinked into **/usr/local**
  Which actually means symlinked into (if needed):
  /usr/local/bin
  /usr/local/lib
  /usr/local/include

- For example the package httpd:

  - Type: **cd /usr/local/bin**
    Type: **ls -al**
    httpd -> ../Cellar/httpd/2.4.47/bin/httpd

HOMEBREW

- Type: `cd /usr/local/lib`
  Type: `ls -al`
  `httpd -> ../Cellar/httpd/2.4.47/lib/httpd`

- Type: `cd /usr/local/include`
  Type: `ls -al`
  `httpd -> ../Cellar/httpd/2.4.47/include/httpd`

HOMEBREW

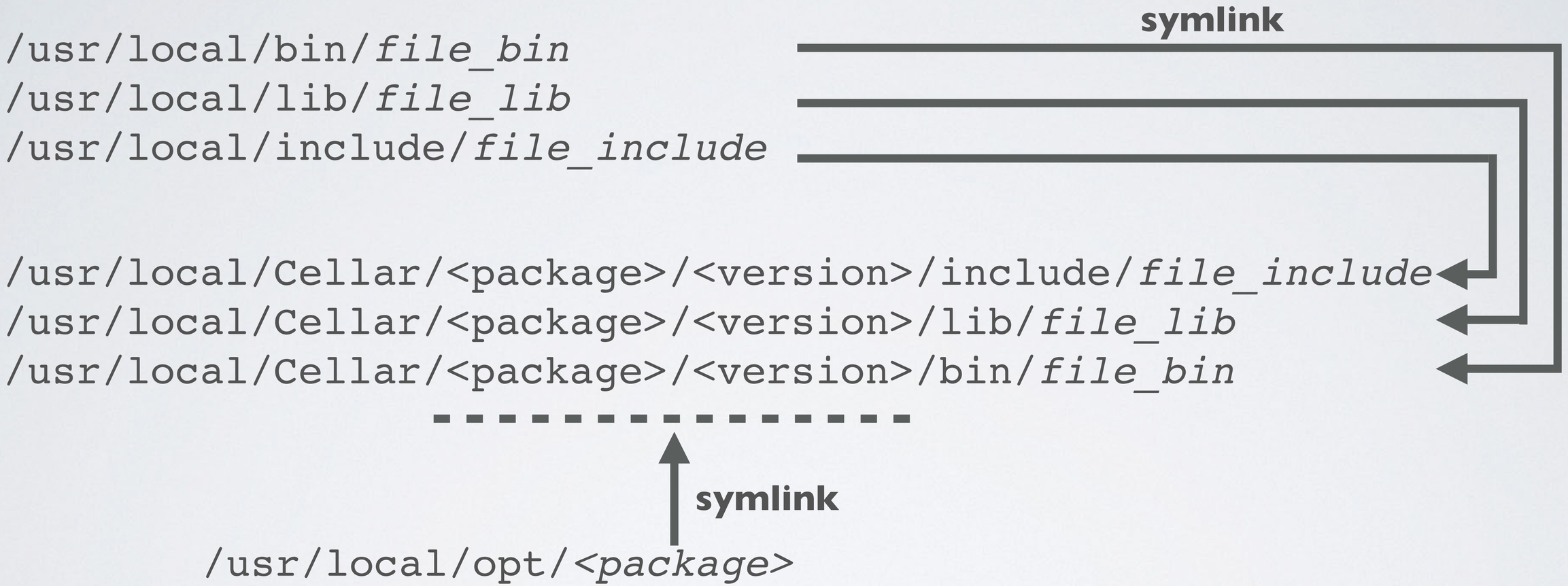
- The packages are also symlinked into:
  `/usr/local/opt`

- For example the package httpd:

  - Type: `cd /usr/local/opt`
    Type: `ls -al`
    `httpd -> ../Cellar/httpd/2.4.48`

- What the purpose is of `/usr/local/opt` will be explained in keg-only.

HOMEBREW


symlink
/usr/local/bin/file_bin
/usr/local/lib/file_lib
/usr/local/include/file_include
/usr/local/Cellar/<package>/<version>/include/file_include
/usr/local/Cellar/<package>/<version>/lib/file_lib
/usr/local/Cellar/<package>/<version>/bin/file_bin
symlink
/usr/local/opt/<package>

HOMEBREW

- Homebrew prefers installing within directory **`/usr/local`**

- Apple has assigned this directory for non-system utilities. This means there are no files in **`/usr/local`** by default, so there is no need to worry about messing up existing tools or system tools.

- The **`/usr/local/bin`** directory must take precedence over **`/bin`**, **`/usr/bin`**, **`/sbin`** and **`/usr/sbin`**, see .bash_profile.
  To check this:
  Type: **`echo $PATH`**
  The $PATH is searched from beginning to end, with the first matching executable being run.

HOMEBREW

Just a reminder, information from: **`man hier`**

Path	Description
/usr/local/bin	Locally installed software that a normal user may run.
/usr/local/sbin	Locally installed programs for system administration.
/bin	User utilities fundamental to both single-user and multi-user environments.
/usr/bin	Common utilities, programming tools, and applications.
/sbin	System programs and administration utilities fundamental to both single-user and multi-user environments.
/usr/sbin	System daemons & system utilities (executed by users).

FORMULA

- Formula is a Ruby script which defines the package (= software).
- The formula contains information such as:
  - Where the package tar ball, containing the package source, can be downloaded from.
  - What package dependencies it has.
  - How to install the package.

FORMULA

Formula example: git-game.rb

```
class GitGame < Formula
  desc "Game for git to guess who made which commit"
  homepage "https://github.com/jsomers/git-game"
  url "https://github.com/jsomers/git-game/archive/1.2.tar.gz"
  sha256 "d893b2c813388754c16d867cf37726cd7e73c9ccd316735aac43bf1cd3ab1412"
  license "MIT"

  bottle :unneeded

  def install
    bin.install "git-game"
  end

  test do
    system "git", "game", "help"
  end
end
```

FORMULA

- The formulae on the macOS can be found at:
  `/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core/Formula/<package-name>.rb`

- The default Homebrew formulae Git location:
  https://github.com/Homebrew/homebrew-core/tree/master/Formula

- To install a formula:

```
brew install <formula>
brew install git-game
```

BOTTLE

- Homebrew provides pre-compiled versions for many formulae.
  These pre-compiled versions are referred to as bottles and are available at:
  https://github.com/Homebrew/homebrew-core/packages

- Homebrew aims to bottle everything and these pre-built binary packages are simple gzipped tarballs of compiled binaries.

- The bottles were hosted at https://homebrew.bintray.com/bottles but as of May 1, 2021 the Bintray hosting provider was shut down.

- The Homebrew's binary packages are migrated to GitHub Packages:
  https://github.com/orgs/Homebrew/packages

- Homebrew 3.1.0+ will have the new default download location.

BOTTLE

- By default if a bottle is available and usable it will be downloaded and installed automatically when you type:
  `brew install <formula>`

- If you wish to disable this you can do so by specifying:
  `brew install --build-from-source <formula>`
  `brew install --enable-bar <formula>`

- Just like formulae and casks, bottles are installed in the cellar:
  `/usr/local/Cellar`

CELLAR

- Homebrew downloads and installs packages in the cellar directory.
- The cellar location: **`/usr/local/Cellar`**
- What happens when the Netwide Assembler (NASM) package is installed:
  - Install nasm:
    `brew install nasm`
  - Execute the nasm Ruby script:
    `/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core/Formula/nasm.rb`

CELLAR

- The nasm package is downloaded and installed in the cellar:
  `/usr/local/Cellar/nasm/2.15.05`

- The nasm binary is installed:
  `/usr/local/Cellar/nasm/2.15.05/bin/nasm`

- The directory **`/usr/local/bin`** contains a symlink to the nasm binary.
  `nasm -> ../Cellar/nasm/2.15.05/bin/nasm`
  Note: The nasm package has no lib or include files.

- The directory **`/usr/local/opt`** contains a symlink to the nasm binary.
  `nasm -> ../Cellar/nasm/2.15.05/bin/nasm`

KEG

- As mentioned earlier a package is installed in the cellar, for example:
  `/usr/local/Cellar/nasm/2.15.05`

- Each package is installed in its own directory followed by the version number.

- The combination package name and version number is referred to as keg.

`/usr/local/Cellar/nasm/2.15.05`

keg

KEG-ONLY

- Keg-only means the formula is only installed in the cellar and the packages are not symlinked into `/usr/local` (meaning `/usr/local/bin`, `/usr/local/lib` or `/usr/local/include`).

- If keg-only, most tools will not find the installed package.
  To check if a formula was installed as keg-only, type: `brew info <formula>`

- You can still create a symbolic link to the formula with:
  `brew link <formula>`
  though this can cause unexpected behaviour if you are shadowing macOS software.

KEG-ONLY

- Homebrew automatically installs a package as keg-only if it detects the new package will cause problems with an already installed package (e.g. version incompatibilities).

- For example the new package shadows a version of a library that ships with macOS, and superseding macOS libraries can cause problems.

- To force a package to be installed as keg-only, type:
  `brew install --force <formula>`

KEG-ONLY

- As mentioned earlier the keg-only packages are not symlinked into **`/usr/local`** but they are symlinked into **`/usr/local/opt`**.

- For example, gettext is keg-only, which means it is not symlinked into **`/usr/local`**

- If you need to have gettext first, modify your PATH:
```
echo ‘export PATH="/usr/local/opt/gettext/bin:$PATH"' >>
~/.bash_profile
```

- For compilers to find gettext you may need to set:
```
export LDFLAGS="-L/usr/local/opt/gettext/lib"
export CPPFLAGS="-I/usr/local/opt/gettext/include"
```

CASK

- Cask is an extension of Homebrew. Just like a formula they are Ruby scripts but they are used to download and install GUI applications.

- The Casks on the macOS can be found at:
  `/usr/local/Homebrew/Library/Taps/homebrew/homebrew-cask/Casks/<package-name>.rb`

- The default casks Git location:
  https://github.com/Homebrew/homebrew-cask/tree/master/Casks

- To install a cask:
  `brew cask install <cask>`
  `brew cask install hex`

CASK

Cask example: hex.rb

```
cask "hex" do
  version "1.0"
  sha256 :no_check

  url "http://dl.hextcg.com/HexInstaller.dmg"
  name "HEX"
  homepage "https://www.hextcg.com/"

  app "Hex.app"
end
```

TAP

- A tap refers to a Git repository and the repository name starts with "**homebrew-**"

- By default homebrew uses the following git repositories:
  https://github.com/Homebrew/homebrew-core
  https://github.com/Homebrew/homebrew-cask
  https://github.com/Homebrew/homebrew-services

TAP

- The command **`brew tap`** without arguments lists the currently tapped repositories:
  - Core formulae for the Homebrew package manager:
    `homebrew/core`
  - Installation and management of GUI macOS applications:
    `homebrew/cask`
  - Manage background services with macOS **`launchctl`** daemon manager.
    `homebrew/services`

- **`<user>/<repository-name>`**
  The actual Git location will be:
  `https://github.com/<user>/homebrew-<repository-name>`

TAP

- More git repositories can be added:
  `brew tap <user/repository-name>`

- When this command is executed the repository
  `https://github.com/<user>/homebrew-<repository-name>`
  is cloned into
  `/usr/local/Homebrew/Library/Taps/<user>/homebrew-<repository-name>`

TAP

- Example add the beeftornado/rmtree git repository:
  `brew tap beeftornado/rmtree`

- The git repository is cloned into:
  `/usr/local/Homebrew/Library/Taps/beeftornado/homebrew-rmtree`

- To remove this tap:
  `brew untap beeftornado/rmtree`
  the tap is removed from the repository list and the cloned git repository is removed from:
  `/usr/local/Homebrew/Library/Taps/`

Useful Brew Commands

BREW HELP

- **`brew --version`**
  Display Homebrew version

- **`brew help`**
  Print help information

- **`brew help <command>`**
  Print help information for a brew command

- **`brew doctor`**
  Check system for potential problems.

BREW HELP

- **`brew home <formula>`**
  Opens a browser and show the formula's home page.

- **`brew deps <formula>`**
  Show all dependencies of a formula.

- **`brew commands`**
  Show all brew commands

- **`brew cat <formula>`**
  Show the Ruby formula source code.

BREW UPDATES

- **`brew update`**
  Updates all the packages and Homebrew itself to the latest version.

- **`brew outdated`**
  List installed casks and formulae that have an updated version available.

- **`brew upgrade`**
  Upgrade outdated casks and outdated, unpinned formulae using the same options they were originally installed with.

- **`brew upgrade <formula>`**
  Upgrade the specified formula.

BREW UPDATES

- **`brew pin <formula>`**
  Pin the specified formula, preventing them from being upgraded when issuing the brew upgrade formula command.

- **`brew unpin <formula>`**
  Undo the pin protection.

- **`brew update-reset`**
  Fetch and reset Homebrew and all tap repositories (or any specified repository) using git to their latest origin/HEAD.

BREW REPOSITORIES

- **`brew tap`**
  Show all current tapped repositories.

- **`brew tap <user/repository-name>`**
  Add new git repository:
  `https://github.com/<user>/homebrew-<repository-name>`

- **`brew untap <user/repository-name>`**
  Remove a specific tap from the repository list.

- **`brew tap <user/repository-name> <url>`**
  Tap a formula repository from the specified URL.

BREW CASK

- **`brew cask list`**
  Show all installed casks.

- **`brew tap caskroom/cask`**
  Add the GitHub cask repository.

- **`brew cask search <text>`**
  Search all casks whose name contains "text".

- **`brew cask install <cask>`**
  Install the cask.

BREW CASK

- **`brew cask uninstall <cask>`**
  Uninstall the cask.

BREW SEARCH, INSTALL, REMOVE

- **`brew list`**
  Show all installed formulae.

- **`brew list --pinned`**
  Show all pinned formulae.

- **`brew search <text>`**
  Search all formulae whose name contains "text".

- **`brew info <formula>`**
  Display information about a formula.

BREW SEARCH, INSTALL, REMOVE

- **`brew install <formula>`**
  Install the formula.

- **`brew reinstall <formula>`**
  Reinstall the formula.

- **`brew uninstall <formula>`**
  Uninstall the formula.
  The package will not be uninstalled if other formulae depends on the formula.
  Use the **`--force`** argument if you want to ignore this check.

BREW CLEANUP

- **`brew cleanup`**
  Delete all old versions for all installed formulae.

- **`brew cleanup <formula>`**
  Delete older versions of the specified formula.

- **`brew cleanup -n`**
  Display all formulae that will be removed (dry run).

BREW INFORMATION

- https://docs.brew.sh/
- Open terminal, type: `man brew`
- Open brew manual in Preview:
  Open terminal, type: `man -t brew | open -fa "Preview"`
- The man brew output in pdf:
  https://www.mobilefish.com/download/homebrew/brew.pdf