Package Managers

What is a Package Manager?

A package manager or package management system is a collection of software tools that automates the process of installing, upgrading, configuring, and removing computer programs for a computer's operating system in a consistent manner.

Wikipedia

The Dark Days Before Package Managers

- “Dependency hell”
  - One application needs another...and that one needs a few...and those....and oh god why do I install things
  - Lots of reading (and writing) README files
  - Large applications (in particular on Windows) because of static linking
  - So hated and well known, it has a Wikipedia page: https://en.wikipedia.org/wiki/Dependency_hell
- Manual updates
  - Of everything you installed
  - Also, dependency hell
- Manual installs
  - Not everything was good at installing itself and sometimes your distro didn’t match what the developers expected

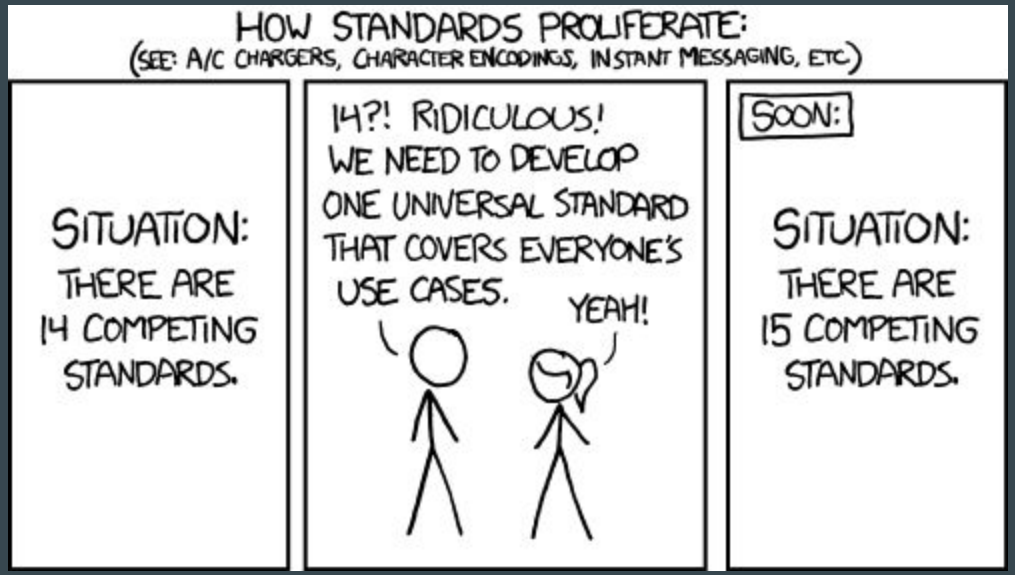
But now, package managers do the hard work for you

Where can I use it?

- All Linux distributions have a package manager (though they have different ones)
  - You may not use it directly; the Ubuntu Store, GNOME Software Center, and other graphical apps call them behind the scenes

- Programming languages (Python, Node, Ruby, etc) for installing requirements

- Windows 10 and Mac OS X (yes, really!)


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

xkcd.com

Terminology

- Package
  - A library or piece of software and information about how to install it. Different formats depending on the package manager.
- Repository
  - A location where you can find packages
- Package List
  - A cached list of packages in a repository
- Mirror
  - A copy of a repository. Various organizations provide mirrors (including RIT!) that you can pick from
- Sources List
  - A list of where to get software, usually a list of mirrors

apt (and aptitude and dpkg)

- Your friendly (Debian|Ubuntu|Linux Mint) package manager!

- Package format: .deb files
  - Fairly commonly available for download when companies distribute their software

Basic apt commands

- `apt-get install      # Install a package`
- `apt-get remove       # Uninstall a package`
- `apt-cache search     # Search for a package`
- `apt-get update       # Update package lists`
- `apt-get upgrade      # Upgrade the system`
- `apt-get dist-upgrade # Run a distribution upgrade`

Relation to aptitude

- Aptitude is a more friendly frontend to apt.
  - Use it as a menu-based console application by running `aptitude`
  - Use it on the command line similar to apt

- Aptitude is often better at helping you figure out dependency problems

Relation to dpkg

- dpkg is the software utility that actually installs and removes packages (.deb files)
  - Use `dpkg -i` to install a .deb file

- dpkg is a lower level tool that you will rarely use directly, and often shouldn't

The most important apt command

`$ apt-get moo` (but only if your apt has super cow powers)

pacman

- Your friendly neighborhood (ArchLinux|Antergos|Chakra|...) package manager

- Package format: .pkg.xz

Basic pacman commands

- `pacman -Ss  # Search for a package`
- `pacman -S   # Install a package`
- `pacman -Rs  # Remove a package`
- `pacman -Syy # Update the package lists`
- `pacman -Syu # Upgrade the system`
- `pacman -U   # Install a .pkg.xz file`

The most important pacman option

Add “ILoveCandy” to /etc/pacman.conf

to make your progress bars go from this:

```
[#######################################################-------]
```

to this:

```
[-----------C  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o ]
```

dnf

- The up and coming replacement to Yum (Fedora, CentOS, RHEL)

- Package format: .rpm files
  - Like .deb, often provided for download

Basic dnf commands

- `dnf install     # install a package (including a package file)`
- `dnf remove      # uninstall a package`
- `dnf search      # search for a package`
- `dnf upgrade     # upgrade packages`

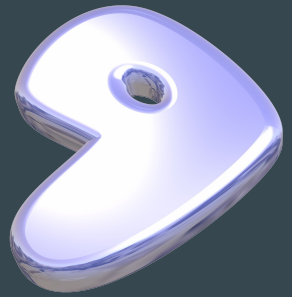
dnf and yum

- Yum is the previous package manager used by RHEL, Fedora, etc. Dnf is its successor

Portage (emerge)

- Used by Gentoo, Sabayon, Funtoo and others

- Package format: ebuild (source) and .tbz2 (binary)
  - Gentoo does not use binary packages by default. ebuild is a shell script with information about how to get, prepare, compile, and install a package





Basic portage commands

- `emerge --sync` # Update package lists
- `emerge -C` # Uninstall a package
- `emerge -uDU --with-bdeps=y @world` # Upgrade your whole system
- `emerge -s` # Search for packages
- `emerge` # Install a package

And of course, the important setting

1. Open /etc/make.conf

2. Add "candy" to FEATURES

The “working” spinner will display a random sentence a few characters at a time.

What a world. Or according to emerge now: “Inaccuracy saves a world of explanation.”

A smattering of other Linux package managers

- ipkg
- opkg
- slackpkg
- nix package manager
- petget
- ...

Homebrew

- Third-party package manager for Mac OS X
  - “Homebrew installs the stuff you need that Apple didn’t” *brew.sh*

- Package format: git + formulae
  - formulae is a Ruby script describing a package

Basic Homebrew commands

- `brew install   # Install a package`
- `brew uninstall # Uninstall a package`
- `brew update    # Update package lists`
- `brew upgrade   # Upgrade installed packages`
- `brew search    # Search packages`

More Mac OS X Package Managers

- Joyent
- Mac App Store
- Fink
- MacPorts
- Nix package manager

OneGet

- Your friendly neighborhood...wait for it...WINDOWS 10 package manager
  - Official, too!
  - Not quite a package manager, it's more of a package manager manager

- Package format: none, it downloads installers that do the work

Basic OneGet Commands

- `> install-package   # Install a package`
- `> find-package      # Search for a package`
- `> uninstall-package # Uninstall a package`

The issue with OneGet...

- It looks easy, but it's not always easy to get it to work
  - I've never once successfully installed anything with it, although it tells me I have

- OneGet downloads and runs the installer (.exe, .msi, what-have-you) and doesn't handle things itself. This means it can't clean anything up on uninstall.

Other Windows Package Managers

- The Windows Store
- chocolatey (available for older Windows, and the inspiration for OneGet)
- cygwin
- wpkg

Programming Language Package Managers

pip

- Handles Python packages and dependencies

- Package format: setup.py script that describes the application (source code)

Basic pip commands

- `pip install   # Install a package`
- `pip uninstall # Uninstall a package`
- `pip search    # Search for a package`
- `pip upgrade   # Upgrade a package`
- `pip freeze    # List installed package versions (not intuitive)`

pip vs easy_install

- Both are used, pip is generally more widely used

- Package formats are the same

Setuptools and distutils with pip

- Python frameworks for handling installation in the setup.py script

- Largely interchangeable, with the exception of a few options

npm

- NodeJS package manager

- Package format: package.json + source
  - package.json describes requirements and the package

Basic npm commands

- `npm install   # Install a package`
- `npm uninstall # Uninstall a package`
- `npm update    # Update packages (not the recommended way, just the easy way)`
- `npm search    # Search for a package`

RubyGems

- Ruby's package manager!

- Package format: similar to Gentoo ebuild. Source, usually with a gemspec file though sometimes built by Rake (Ruby's Make)

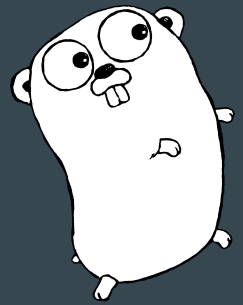
Basic Gem Commands

- `gem install   # Install a package`
- `gem uninstall # Uninstall a package`
- `gem search    # Search for a package`

Other Programming Language Package Managers

- NuGet (C#)
- Go, Gopm (Go)
- Composer (PHP)
- CocoaPods (Objective-C, but built in Ruby)
- Maven (Java)

maven



COCOAPODS

Final Thoughts

- The package manager and package format should factor into your distro choice
  - Changing the package manager on your distro is not an option (they are not compatible)
  - Creating packages (if you write software) is way easier for some than it is others

- Extra security features (like package signing) are nice

- Changing your mirrors can improve your experience substantially
  - mirrors.rit.edu is super fast
  - There's often a mirrors.<your ISP, your school, or your company>.whatever