SOFTWARE INSTALLATION

Gotta have it

DELIVERY!

- Software is delivered in one of two manners:

    - Source form - requires compiling

    - Binary form - generally wrapped up in a package

WHICH IS BEST?

- Both formats have their advantages and disadvantages..

  - Compiling from source can provide higher performing machine code, plus it gives the option of selecting features and configurations only available at compile time.

  - Pre-compiled software is easier - it alleviates the [possible] headaches of compiling, and if distributed in a package format, provides built-in management functionality.

COMPILING

- Compiling from source can be tricky.

- First of all, the development tools and packages must be installed, most importantly: `gcc` and `make`.

- `gcc`: The GNU C Compiler. The de facto compiler for open source software.

- `make`: GNU Make. A development tool which uses a rules-based configuration syntax to determine and run all of the necessary commands needed to build a software project.

COMPILING BASICS

- The <u>basic</u> steps for compiling a software package:

  - Download the source tarball

  - `cd` into the extracted directory

  - Read the `INSTALL` and/or `README` file, follow directions!

  - `./configure`

  - `make`

  - `make install`

PACKAGES

- Installing a software package is pretty straight forward.

- There are a few different package formats out there. The two most popular are:

  - `rpm`: Redhat Package Manager

  - `deb`: Debian package

- In this course, we'll only be focusing on rpm's. Deb's have similar functionality and capability, so learning the command syntax is about all that is needed for proficiency.

RPM

- RPM's provide full software packaging features: pre-install scripts, post-install scripts, dependencies, meta information, and an installed software database to name a few.

- The RPM system maintains a database of all installed software on a machine - this is useful for tracking and updating reasons, as well as dependency verification and software management.

RPM

- `rpm`: The Redhat Package Manager tool. Provides interface to RPM system, performing queries, installs, upgrades, uninstalls and general database maintenance operations.

  - `-i` option: <u>i</u>nstall the given package

  - `-q` option: <u>q</u>uery the database

  - `-e` option: <u>e</u>rase the given package from the system

YUM

- Not yum as in "This is yummy!"

- yum: Yellowdog Updater Modified

  - Supports package installation over the network through <u>repositories</u>.

  - RPM backend

  - Simple interface

slideshow.end();