

Software installation

Lana Drachova CPSC 4240/6240 Lecture Series

Methods of Installation

1. Package manager from a command line
2. GUI package managers
3. Compiling from source

What is a *package*?

Collection of files organized in directories. Plus a file containing metadata.

Packages include a manifest of *dependencies*, or lists of programs and versions that must exist on your computer for software to run correctly.

Remain most common

New *Snap format* treats programs as self-contained, isolated objects that run in their own protected space, so they aren't "dependent on dependencies." (src. Lifewire.com)

1. Command line Package Manager

A package manager is a collection of tools to automate installing, removing, upgrading, and configuring software on a Linux distribution.

Not cross compatible, depend on your architectural build.

Distribution specific:

dpkg: Base package manager, Debian-based distros (Ubuntu, Mint, Elementary).

apt: Simplified front-end for the DPKG system, Debian-based distros

apt-get: Feature-rich front-end for the DPKG system, Debian-based distros.

rpm: Base package manager, Red Hat-based distributions (RHEL, CentOS, Fedora)

yum: Front-end for the RPM system, Red Hat-based distributions.

pacman: The package manager for Arch Linux-based distributions.

snappy: New, self-contained package format developed by Ubuntu.

Difference between apt (and apt-get) and dpkg

apt and **apt-get** are used to remotely dnld/install software from a repository

dpkg used to install already downloaded (local) file with extension *.deb* or *.udeb*

1. Installing with dpkg

Debian-based distros have installer files with extension *.deb*.

Some software has to be downloaded from vendor websites because they are not available in the repositories. (Be Careful!)

.deb - Unix type tarball that contains three files:

debian binary file (typically a one-liner with package format version number),

control archive - tar archive *control.tar* with scripts and package meta-data

(package name, version, dependencies, md5 checksums, etc).

data archive - tar archive named *data.tar* contains installable files.

.udeb - *micro deb* file that includes some but not all of the same information as a regular DEB file.

Can convert .deb to other formats !

Can install .deb files from GUI by clicking the installer file.

Install your .deb file using command:

dpkg -i /path/file.deb

Install several files from the same directory

dpkg -i -R /path_with_deb_files/

Remove

apt-get remove /path/file.deb

Do **man dpkg** for more options and info.

2. Advanced Package Tool (APT)

is a command line program used to interact with the APT
used to install, update, list, and remove packages from your system.
commands: **apt-get**, **apt-cache**, **apt-config**, and recently added **apt**.
used with **sudo** (on most systems)

*Differences between **apt** and **apt-get***

Both download packages from sw repositories and install them

apt is a new simplified addition from Ubuntu 16.04 and Debian 8

combined functionalities of **apt-get** and **apt-cache**, but not always backwards compatible
gives better and more detailed output

adds a line that tells you how many packages can be upgraded

command **apt list - -upgradable** can be used to show these packages

color-codes the output

shows progress bar

Functionality	apt	apt-get
Update package repository	apt update	apt-get update
Upgrade packages	apt upgrade	apt-get upgrade
Upgrade packages and remove unnecessary dependencies	apt full-upgrade	apt-get dist-upgrade
Search a package	apt search <i>package</i>	apt-get search <i>package</i>
Show package information	apt show <i>package</i>	apt-cache show <i>package</i>
Show installed and available	apt policy <i>package</i>	apt-cache policy <i>package</i>

package versions		
Show package sources	apt policy	apt-cache policy
Install a package	apt install <i>package</i>	apt-get install <i>package</i>
Remove a package	apt remove <i>package</i>	apt-get remove <i>package</i>
Remove a package and all configurations	apt purge <i>package</i>	apt-get purge <i>package</i>
Remove unneeded dependencies	apt autoremove	apt-get autoremove
List packages with certain criteria	apt list <i>criteria</i>	
Edit list of sources	apt edit-sources	

Difference between apt update and apt upgrade

Before you install a package it is a good idea to synchronize the package index files and update the package repository to the latest version. This ensures that the packages you install are up-to-date.

2. Using GUI Package Manager

1. Ubuntu Software manager.

Allows to select or deselect packages
Sponsored ads.

2. Synaptic Package Manager

Replaced Ubuntu Software Center.
No ads for paid software.
Display results from all repositories in */etc/apt/sources.list*

3. Compiling from the course

Can compile and create an installable package.
Compilation produces an optimized image for your machine.