

# APT Package Management

---

## Table of Contents

Introduction.....	1
Install a Package.....	2
Remove a Package.....	2
Finding a Package .....	2
Software Updates.....	3
Upgrade vs. Update.....	3
List Installed Packages.....	3
Other Packages to Install.....	3
Printing .....	3
APT Directories and Files .....	3
Repositories.....	3
Package Database .....	4
Package Cache .....	4

## Introduction

This tutorial assumes you are using a Ubuntu Linux system.

APT is an acronym for Advanced Package Tool.

A package is a collection of scripts, programs, files, and directories needed to install and run a program or application. If the package is dependent on another package, that information is also included in the package. A package may contain an application in the form of source code or an executable file.

Packages primarily come in two different formats: RPM and dpkg. RPM was originally an acronym for Red Hat Package Manager, but it is now the recursive acronym for RPM Package Manager. RPM packages are used by Fedora and SUSE. The Debian package manager, dpkg, is used by Debian and Ubuntu.

The **apt-get**, **apt-cache**, and **apt-config** commands are low level commands for package management. They are built on the even lower level **dpkg** commands. The **apt** command provides a more structured, high-level interface for the most commonly used options from these commands. Being a higher level interface, its usage is preferable when it will do the job.

Use the **sudo** command to run any commands that install or update the packages on the system.

## Install a Package

If you want to install a package that is not supported by Ubuntu, you need to add the repository where it is located to the **source.list** file (see the section “Repositories” later in this document).

Before installing a package, you should update the local package database. Many system administrators use a **cron** script to update this database nightly, in which case, this step is not necessary.

```
$ sudo apt update
```

Install the package:

```
$ sudo apt install packageName
```

## Remove a Package

Remove the package:

```
$ sudo apt remove packageName
```

Removing a package does not remove configuration files for the package, which is useful if the package is reinstalled.

If the removed package was dependent upon another package that is no longer used, the output of the command will list that package as no longer needed, but it will not remove it.

If you want to remove a package, including its configuration files, use the **apt purge** command. You can also use this command after the **apt remove** command to remove the configuration files at a later time.

```
$ sudo apt purge packageName
```

You can use the **apt autoremove** command to remove packages that were automatically installed to satisfy a dependency but are no longer needed.

```
$ sudo apt autoremove
```

## Finding a Package

You can search for package names that match a pattern using the **apt search** command.

```
$ apt search pattern
```

Or, use the **packages.ubuntu.com** web page, which allows you to search packages using several criteria. For example, this utility enables you to find the package that contains a file or command.

You can find the package that contains a particular file using the **apt-file** command.

```
$ apt-file search filename
```

# Software Updates

## Upgrade vs. Update

The **apt update** command updates the list of available packages and their versions, but it does not install or upgrade any packages. After running this command, APT has the information locally to determine whether the installed packages are the most recent version.

The **apt upgrade** command actually installs newer versions of the packages you already have.

To keep Ubuntu up to date, periodically execute the following commands. The **-y** option instructs the **apt** command to assume a **yes** response to most prompts.

```
$ sudo apt update
$ sudo apt upgrade -y
```

## List Installed Packages

Run the **apt list** command to list Ubuntu Linux packages that are available.

```
$ apt list --all_versions
```

To list the packages that are currently installed:

```
$ apt list --installed
```

To list the packages that are currently installed and the package name matches a pattern:

```
$ apt list --installed pattern
```

## Other Packages to Install

**apt-file**                      Contains the **apt-file** command.

## Printing

```
$ sudo apt install cups lpr
$ sudo service cups start
$ sudo service lpd start

lpadmin -p printer -E -v device -m ppd Lpadmin
```

## APT Directories and Files

### Repositories

Repositories are collections of packages available for download. They are usually on the installation CDROM or a server accessible via the Internet. Ubuntu maintains repositories for each of its supported releases.

The file **/etc/apt/sources.list** and the files in **/etc/apt/sources.list.d/** contain the location and description of the repositories that are configured for use by the current system. That is, this is the list of repositories that the APT commands will search when you want to find or install a package.

In the **sources.list** file(s), each repository is listed on a separate line in the following format:

*Type URI Repository Category ...*

Type		
	deb	Packages contain executable files.
	deb-src	Packages contain source files.
URI		Location of the repository ( <b>cdrom</b> or an internet address).
Repository		Name of the repository at that location.
Category		Identifies the type of packages in the repository.
	main	Ubuntu supported open source software.
	universe	Community maintained open source software.
	multiverse	Software restricted by copyright.
	restricted	Proprietary device drivers.
	extras	Third party developed software.
	partner	Offered by vendors as service to Ubuntu users.
	backports	Packages from later releases of Ubuntu.

When making changes to the repositories, you should comment or uncomment entries in the **sources.list** file, or add new entries in files that you create in the **sources.list.d** directory. You can search for “Ubuntu repositories” on the Internet to find additional repositories.

## Package Database

The Package Database is a local database that contains the list of available packages and their versions. This information is created by reading each of the repositories configured for use on the system and recording information about each package that is available in each repository. The files that make up the package database are stored in the directory **/var/lib/apt/lists/**.

## Package Cache

The APT Package Cache is located in the directory **/var/cache/apt/archives/**. It contains recently downloaded package files.