

Software Installation and Package Management on Linux (Debian/Ubuntu)

Managing software packages is a core part of any Linux system. On Linux-based distributions (e.g., **Ubuntu**, **Debian**, **Linux Mint**, **Proxmox VE**), the **APT (Advanced Package Tool)** is used for this. In addition, there are utility programs like `wget` to download files from the internet.

Unlike with Windows, Linux has **no automatic update service** that updates the system without user interaction. Updates must therefore be **triggered manually**, for example with:

```
sudo apt update      # Update package lists  
sudo apt upgrade    # Apply installable updates
```

◊ APT Fundamentals

APT works with a **package repository**, which contains a collection of software packages. Each package consists of programs, libraries, and metadata.

The **repositories** are maintained by the respective Linux distributions and ensure that the packages are up-to-date and verified. For the system to know which packages are available, `apt` relies on a **package list** that should be updated regularly.

When installing or updating packages, the corresponding files are queried and downloaded via the **URLs stored in the repository**.

Frequently Used APT Commands

1. Update Package Lists

```
sudo apt update
```

- Retrieves the latest package information from the configured repositories.
- **Does not yet perform an upgrade**, but only updates the database.
- Should always be run **before installations**.

2. Update System Packages

```
sudo apt upgrade
```

- Installs the latest versions of all already installed packages.

- Normally leaves existing configurations untouched.

☞ Tip: To also automatically update dependencies and orphaned packages, you can use:

```
sudo apt full-upgrade
```

3. Install Software

```
sudo apt install <package name>
```

Examples:

```
sudo apt install apache2  
sudo apt install htop
```

- Downloads and installs the package and all its dependencies.
- For graphical programs, additional libraries are automatically installed.

4. Remove Software

```
sudo apt remove <package name>
```

Example:

```
sudo apt remove apache2
```

- Removes the package but **leaves configuration files behind**.
- Useful if you want to install the software again later.

☞ To also delete the configuration:

```
sudo apt purge <package name>
```

5. Remove Orphaned Packages

```
sudo apt autoremove
```

- Automatically removes no longer needed dependencies.
 - Practical after uninstalling software.
-

6. Search for a Package

```
apt search <search term>
```

Example:

```
apt search nginx
```

- Displays available packages that match the search term.
-

7. Show Package Information

```
apt show <package name>
```

Example:

```
apt show curl
```

- Provides details such as version, description, and dependencies.
-

🔗 Upgrade to a New Ubuntu Version

When a **new Ubuntu version** is released (e.g., from Ubuntu 22.04 LTS → 24.04 LTS), `apt update && apt upgrade` is not enough. For this, the **do-release-upgrade** tool is used.

Preparation

1. Apply all current updates:

```
sudo apt update && sudo apt upgrade && sudo apt full-upgrade
```

2. Reboot the system (if a kernel update was installed):

```
sudo reboot
```

Start Upgrade

```
sudo do-release-upgrade
```

- Starts the Ubuntu upgrade assistant.
- Guides you through the complete upgrade process.
- May ask how to handle existing configuration files.

☞ If `do-release-upgrade` is not available, it can be installed:

```
sudo apt install update-manager-core
```

Manual Upgrade (Alternative Method)

If you change the release directly via `/etc/apt/sources.list`:

1. Change all entries from, for example, `jammy` → `noble` (depending on the version).
2. Afterwards:

```
sudo apt update && sudo apt full-upgrade
```

⚠ Warning: This manual method is **more prone to errors** and is only recommended for experienced admins.

🌐 Downloading Files with `wget`

`wget` is a command-line tool to download files from the internet.

Note: Software should **only be installed** via `wget` if the desired application is **not available through the package repository**. By default, it is always safer and easier to install packages via **APT**.

```
sudo apt install wget
```

Download a File

```
wget <URL>
```

Example:

```
wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.43/bin/apache-tomcat-10.1.43.tar.gz
```

- Downloads the file to the current directory.

Note: You must either be in the desired download directory before running the command, or explicitly specify the destination path during the download (see example below):

```
wget -P /path/to/destination/directory https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.43/bin/apache-tomcat-10.1.43.tar.gz
```

Download Multiple Files

```
wget -i urls.txt
```

- Useful if `urls.txt` contains a list of URLs.

Download in the Background

```
wget -b <URL>
```

Summary

Command	Function
<code>sudo apt update</code>	Updates package lists
<code>sudo apt upgrade</code>	Installs new package versions
<code>sudo apt full-upgrade</code>	Also updates dependencies
<code>sudo apt install</code>	Installs a package
<code>sudo apt remove</code>	Removes package, keeps config files
<code>sudo apt purge</code>	Removes package including configuration
<code>sudo apt autoremove</code>	Removes orphaned dependencies
<code>apt search</code>	Searches for packages
<code>apt show</code>	Shows package details
<code>sudo do-release-upgrade</code>	Upgrades to a new Ubuntu version
<code>wget <URL></code>	Downloads a file from the internet

Sources

- "apt > apt > Wiki > ubuntuusers.de". Accessed: August 21, 2025. [Online]. Available at: [apt reference](#)
 - "Upgrade > Wiki > ubuntuusers.de". Accessed: August 21, 2025. [Online]. Available at: [Upgrade reference](#)
 - "wget > Wiki > ubuntuusers.de". Accessed: August 21, 2025. [Online]. Available at: [wget reference](#)
-

License

This work is licensed under the **Creative Commons Attribution - ShareAlike 4.0 International License**.

[To the license text on the Creative Commons website](#)