

# General recommendations

This document is an annotated index of popular articles and important information for improving and adding functionalities to the installed Arch system. Readers are assumed to have read and followed the [Installation guide](#) to obtain a basic Arch Linux installation. Having read and understood the concepts explained in [#System administration](#) and [#Package management](#) is *required* for following the other sections of this page and the other articles in the wiki.

## Related articles

---

[FAQ](#)

[Installation guide](#)

[List of applications](#)

## Contents

- [1 System administration](#)
  - [1.1 Users and groups](#)
  - [1.2 Privilege escalation](#)
  - [1.3 Service management](#)
  - [1.4 System maintenance](#)
- [2 Package management](#)
  - [2.1 pacman](#)
  - [2.2 Repositories](#)
  - [2.3 Mirrors](#)
  - [2.4 Arch Build System](#)
  - [2.5 Arch User Repository](#)
- [3 Booting](#)
  - [3.1 Hardware auto-recognition](#)
  - [3.2 Microcode](#)
  - [3.3 Retaining boot messages](#)
  - [3.4 Num Lock activation](#)
- [4 Graphical user interface](#)
  - [4.1 Display server](#)
  - [4.2 Display drivers](#)
  - [4.3 Desktop environments](#)
  - [4.4 Window managers](#)
  - [4.5 Display manager](#)
- [5 Power management](#)
  - [5.1 ACPI events](#)
  - [5.2 CPU frequency scaling](#)
  - [5.3 Laptops](#)
  - [5.4 Suspend and Hibernate](#)
- [6 Multimedia](#)
  - [6.1 Sound](#)
  - [6.2 Browser plugins](#)
  - [6.3 Codecs](#)
- [7 Networking](#)
  - [7.1 Clock synchronization](#)
  - [7.2 DNS security](#)

- 7.3 Setting up a firewall
- 7.4 Resource sharing
- 8 Input devices
  - 8.1 Keyboard layouts
  - 8.2 Mouse buttons
  - 8.3 Laptop touchpads
  - 8.4 TrackPoints
- 9 Optimization
  - 9.1 Benchmarking
  - 9.2 Improving performance
  - 9.3 Solid state drives
- 10 System service
  - 10.1 File index and search
  - 10.2 Local mail delivery
  - 10.3 Printing
- 11 Appearance
  - 11.1 Fonts
  - 11.2 GTK+ and Qt themes
- 12 Console improvements
  - 12.1 Aliases
  - 12.2 Alternative shells
  - 12.3 Bash additions
  - 12.4 Colored output
  - 12.5 Compressed files
  - 12.6 Console prompt
  - 12.7 Emacs shell
  - 12.8 Mouse support
  - 12.9 Scrollback buffer
  - 12.10 Session management

## System administration

This section deals with administrative tasks and system management. For more, please see **Core utilities** and **Category:System administration**.

### Users and groups

A new installation leaves you with only the superuser account, better known as "root". Logging in as root for prolonged periods of time, possibly even exposing it via **SSH** on a server, **is insecure** (<https://apple.stackexchange.com/questions/192365/is-it-ok-to-use-the-root-user-as-a-normal-user/192422#192422>). Instead, you should create and use unprivileged user account(s) for most tasks, only using the root account for system administration. See **Users and groups#User management** for details.

Users and groups are a mechanism for *access control*; administrators may fine-tune group membership and ownership to grant or deny users and services access to system resources. Read the **Users and groups** article for details and potential security risks.

### Privilege escalation

Both the **su** and **sudo** commands allow you to run commands as another user. By default **su** drops you to a login shell as the root user, and **sudo** by default temporarily grants you root privileges for a single command. See their respective articles for differences.

## Service management

Arch Linux uses **systemd** as the **init** process, which is a system and service manager for Linux. For maintaining your Arch Linux installation, it is a good idea to learn the basics about it. Interaction with *systemd* is done through the *systemctl* command. Read **systemd#Basic systemctl usage** for more information.

## System maintenance

Arch is a rolling release system and has rapid package turnover, so users have to take some time to do **system maintenance**. Read **Security** for recommendations and best practices on hardening the system.

## Package management

This section contains helpful information related to package management. For more, please see **FAQ#Package management** and **Category:Package management**.

**Note:** It is imperative to keep up to date with changes in Arch Linux that require manual intervention **before** upgrading your system. Subscribe to the **arch-announce mailing list** (<https://mailman.archlinux.org/mailman/listinfo/arch-announce/>) or check the front page **Arch news** (<https://www.archlinux.org/>) every time before you update. Alternatively, you may find it useful to subscribe to **this RSS feed** (<https://www.archlinux.org/feeds/news/>) or follow **@archlinux** (<https://twitter.com/archlinux>) on Twitter.

## pacman

**pacman** is the Arch Linux *package manager*: all users are required to become familiar with it before reading any other articles.

See **pacman/Tips and tricks** for suggestions on how to improve your interaction with *pacman* and package management in general.

## Repositories

See **Official repositories** for details about the purpose of each officially maintained repository.

If you plan on using 32-bit applications, you will want to enable the **multilib** repository.

**Unofficial user repositories** lists several other unsupported repositories.

You may consider installing the **pkgstats** service.

## Mirrors

Visit **Mirrors** for steps on taking full advantage of using the fastest and most up to date mirrors of the official repositories. As explained in the article, a particularly good advice is to routinely check the **Mirror Status** (<https://www.archlinux.org/mirrors/status/>) page for a list of mirrors that have been recently synced.

## Arch Build System

*Ports* is a system initially used by BSD distributions consisting of build scripts that reside in a directory tree on the local system. Simply put, each port contains a script within a directory intuitively named after the installable third-party application.

The **Arch Build System** offers the same functionality by providing build scripts called **PKGBUILDs**, which are populated with information for a given piece of software; integrity hashes, project URL, version, license and build instructions. These PKGBUILDs are later parsed by **makepkg**, the actual program that generates packages cleanly manageable by *pacman*.

Every package in the repositories along with those present in the AUR are subject to recompilation with *makepkg*.

## Arch User Repository

While the Arch Build System allows the ability of building software available in the official repositories, the **Arch User Repository** (AUR) is the equivalent for user submitted packages. It is an unsupported repository of build scripts accessible through the **web interface** (<https://aur.archlinux.org/>) or through **Aurjson**.

## Bootting

This section contains information pertaining to the boot process. An overview of the Arch boot process can be found at **Arch boot process**. For more, please see **Category:Boot process**.

## Hardware auto-recognition

Hardware should be auto-detected by **udev** during the boot process by default. A potential improvement in boot time can be achieved by disabling module auto-loading and specifying required modules manually, as described in **Kernel modules**. Additionally, **Xorg** should be able to auto-detect required drivers using `udev`, but users have the option to configure the X server manually too.

## Microcode

Processors may have **faulty behaviour** (<http://www.anandtech.com/show/8376/intel-disables-tsx-instructions-erratum-found-in-haswell-haswelleep-broadwelly>), which the kernel can correct by updating the *microcode* on startup. Intel processors require a separate package to this effect. See **Microcode** for details.

## Retaining boot messages

Once it concludes, the screen is cleared and the login prompt appears, leaving users unable to gather feedback from the boot process. [Disable clearing of boot messages](#) to overcome this limitation.

## Num Lock activation

Num Lock is a toggle key found in most keyboards. For activating Num Lock's number key-assignment during startup, see [Activating Numlock on Bootup](#).

## Graphical user interface

This section provides orientation for users wishing to run graphical applications on their system. See [Category:X server](#) for additional resources.

### Display server

**Xorg** is the public, open-source implementation of the **X Window System** (commonly X11, or X). It is required for running applications with graphical user interfaces (GUIs), and the majority of users will want to install it.

**Wayland** is a newer, alternative display server protocol and the Weston reference implementation is available.

### Display drivers

The default vesa display driver will work with most video cards, but performance can be significantly improved and additional features harnessed by installing the appropriate driver for **ATI**, **Intel**, or **NVIDIA** products.

### Desktop environments

Although Xorg provides the basic framework for building a graphical environment, additional components may be considered necessary for a complete user experience. **Desktop environments** such as **GNOME**, **KDE**, **LXDE**, and **Xfce** bundle together a wide range of *X clients*, such as a window manager, panel, file manager, terminal emulator, text editor, icons, and other utilities. Users with less experience may wish to install a desktop environment for a more familiar environment. See [Category:Desktop environments](#) for additional resources.

### Window managers

A full-fledged desktop environment provides a complete and consistent graphical user interface, but tends to consume a considerable amount of system resources. Users seeking to maximize performance or otherwise simplify their environment may opt to install a **window manager** alone and hand-pick desired extras. Most desktop environments allow use of an alternative window manager as well. **Dynamic**, **stacking**, and **tiling** window managers differ in their handling of window placement.

### Display manager

Most desktop environments include a **display manager** for automatically starting the graphical environment and managing user logins. Users without a desktop environment can install one separately. Alternatively you may **start X at login** as a simple alternative to a display manager.

## Power management

This section may be of use to laptop owners or users otherwise seeking power management controls. For more, please see **Category:Power management**.

See **Power management** for more general overview.

### ACPI events

Users can configure how the system reacts to ACPI events such as pressing the power button or closing a laptop's lid. For the new (recommended) method using **systemd**, see **Power management with systemd**. For the old method, see **acpid**.

### CPU frequency scaling

Modern processors can decrease their frequency and voltage to reduce heat and power consumption. Less heat leads to more quiet system and prolongs the life of hardware. See **CPU frequency scaling** for details.

### Laptops

For articles related to portable computing along with model-specific installation guides, please see **Category:Laptops**. For a general overview of laptop-related articles and recommendations, see **Laptop**.

### Suspend and Hibernate

See main article: **Suspend and hibernate**.

## Multimedia

**Category:Multimedia** includes additional resources.

### Sound

**Sound** is provided by kernel sound drivers:

- **ALSA** is included with the kernel and is recommended because usually it works out of the box (it just needs to be **unmuted**).
- **OSS** is a viable alternative in case ALSA does not work.

Users may additionally wish to install and configure a **sound server** such as **PulseAudio**. For advanced audio requirements, see **professional audio**.

## Browser plugins

For access to certain web content, **browser plugins** such as Adobe Acrobat Reader, Adobe Flash Player, and Java can be installed.

## Codecs

**Codecs** are utilized by multimedia applications to encode or decode audio or video streams. In order to play encoded streams, users must ensure an appropriate codec is installed.

## Networking

This section is confined to small networking procedures. Head over to **Network configuration** for a full guide. For more, please see **Category:Networking**.

## Clock synchronization

The **Network Time Protocol** (NTP) is a protocol for synchronizing the clocks of computer systems over packet-switched, variable-latency data networks. See **Time#Time synchronization** for implementations of such protocol.

## DNS security

For better security while browsing web, paying online, connecting to **SSH** services and similar tasks consider using **DNSSEC**-enabled client software which can validate signed **DNS** records, and **DNSCrypt** to encrypt DNS traffic.

## Setting up a firewall

A firewall can provide an extra layer of protection on top of the Linux networking stack. While the stock Arch kernel is capable of using **Netfilter**'s **iptables** and **nftables**, neither are enabled by default. It is highly recommended to set up some form of firewall. See **Category:Firewalls** for available guides.

## Resource sharing

To share files among the machines in a network, follow the **NFS** or the **SSHFS** article.

Use **Samba** to join a Windows network. To configure the machine to use Active Directory for authentication, read **Active Directory Integration**.

See also **Category:Network sharing**.

## Input devices

This section contains popular input device configuration tips. For more, please see **Category:Input devices**.

## Keyboard layouts

Non-English or otherwise non-standard keyboards may not function as expected by default. The necessary steps to configure the keymap are different for virtual console and **Xorg**, they are described in **Keyboard configuration in console** and **Keyboard configuration in Xorg** respectively.

## Mouse buttons

Owners of advanced or unusual mice may find that not all mouse buttons are recognized by default, or may wish to assign different actions for extra buttons. Instructions can be found in **Mouse buttons**.

## Laptop touchpads

Many laptops use **Synaptics** (<https://www.synaptics.com/>) or **ALPS** (<http://www.alps.com/>) "touchpad" pointing devices. For these, and several other touchpad models, you can use either the Synaptics input driver or libinput; see **Touchpad Synaptics** and **libinput** for installation and configuration details.

## TrackPoints

See the **TrackPoint** article to configure your TrackPoint device.

# Optimization

This section aims to summarize tweaks, tools and available options useful to improve system and application performance.

## Benchmarking

**Benchmarking** is the act of measuring performance and comparing the results to another system's results or a widely accepted standard through a unified procedure.

## Improving performance

The **Improving performance** article gathers information and is a basic rundown about gaining performance in Arch Linux.

## Solid state drives

The **Solid State Drives** article covers many aspects of solid state drives, including configuring them to maximize their lifetimes.

# System service

This section relates to **daemons**. For more, please see **Category:Daemons and system services**.

## File index and search



Most distributions have a *locate* command available to be able to quickly search for files. To get this functionality in Arch Linux, **mlocate** (<https://www.archlinux.org/packages/?name=mlocate>) is the recommended install. After the install you should run *updatedb* to index the filesystems.

**Desktop search engines** provide a similar service, while better integrated into **desktop environments**.

## Local mail delivery

A default setup does not provide a way to sync mail. To configure *Postfix* for simple local mailbox delivery, see **Postfix**. Other options are **SSMTP**, **msmtp** and **fdm**.

## Printing

**CUPS** is a standards-based, open source printing system developed by Apple. See **Category:Printers** for printer-specific articles.

## Appearance

This section contains frequently-sought "eye candy" tweaks for an aesthetically pleasing Arch experience. For more, please see **Category:Eye candy**.

## Fonts

You may wish to install a set of TrueType fonts, as only unscalable bitmap fonts are included in a basic Arch system. There are several general-purpose **font families** providing large **Unicode** coverage and even **metric compatibility** with fonts from other operating systems.

A plethora of information on the subject can be found in the **Fonts** and **Font configuration** articles.

If spending a significant amount of time working from the virtual console (i.e. outside an X server), users may wish to change the console font to improve readability; see **Fonts#Console fonts**.

## GTK+ and Qt themes

A big part of the applications with a graphical interface for Linux systems are based on the **GTK+** or the **Qt** toolkits. See those articles and **Uniform look for Qt and GTK applications** for ideas to improve the appearance of your installed programs and adapt it to your liking.

## Console improvements

This section applies to small modifications that better console programs' practicality. For more, please see **Category:Command shells**.

## Aliases

Aliasing a command, or a group thereof, is a way of saving time when using the console. This is specially helpful for repetitive tasks that do not need significant alteration to their parameters between executions. Common time-saving aliases can be found in [Bash#Aliases](#), which are easily portable to [zsh](#) as well.

## Alternative shells

[Bash](#) is the shell that is installed by default in an Arch system. The live installation media, however, uses [zsh](#) with the [grml-zsh-config](https://www.archlinux.org/packages/?name=grml-zsh-config) (<https://www.archlinux.org/packages/?name=grml-zsh-config>) addon package. See [Command-line shell#List of shells](#) for more alternatives.

## Bash additions

A list of miscellaneous Bash settings, including completion enhancements, history search and [Readline](#) macros is available in [Bash#Tips and tricks](#).

## Colored output

This section is covered in [Color output in console](#).

## Compressed files

Compressed files, or archives, are frequently encountered on a GNU/Linux system. [Tar](#) is one of the most commonly used archiving tools, and users should be familiar with its syntax (Arch Linux packages, for example, are simply xzipped tarballs). See [Bash/Functions](#) for other helpful commands.

## Console prompt

The console prompt (PS1) can be customized to a great extent. See [Bash/Prompt customization](#) or [Zsh#Prompts](#) if using Bash or Zsh, respectively.

## Emacs shell

Emacs is known for featuring options beyond the duties of regular text editing, one of these being a full shell replacement. Consult [Emacs#Colored output issues](#) for a fix regarding garbled characters that may result from enabling colored output.

## Mouse support

Using a mouse with the console for copy-paste operations can be preferred over [GNU Screen](#)'s traditional copy mode. Refer to [Console mouse support](#) for comprehensive directions. Note that you can already do this in [terminal emulators](#) with the [clipboard](#).

## Scrollbar buffer

To be able to save and view text which has scrolled off the screen, refer to [General troubleshooting#Scrollbar](#).

## Session management

Using terminal multiplexers like **tmux** or **GNU Screen**, programs may be run under sessions composed of tabs and panes that can be detached at will, so when the user either kills the terminal emulator, terminates **X**, or logs off, the programs associated with the session will continue to run in the background as long as the terminal multiplexer server is active. Interacting with the programs requires reattaching to the session.

Retrieved from "[https://wiki.archlinux.org/index.php?title=General\\_recommendations&oldid=512844](https://wiki.archlinux.org/index.php?title=General_recommendations&oldid=512844)"

---

- This page was last edited on 5 March 2018, at 23:03.
- Content is available under [GNU Free Documentation License 1.3 or later](#) unless otherwise noted.