MocaccinoOS

Desktop Edition — Documentation

Source: https://www.mocaccino.org/docs/

Generated: February 2026

Mocaccino Desktop is a Gentoo-based Linux distribution oriented towards Desktop systems. It uses Luet as its package manager and delivers software as self-contained layers — making installation and upgrades as simple as possible.

Table of Contents

1. Documentation Overview

https://www.mocaccino.org/docs/

Welcome to the MocaccinoOS Desktop documentation. MocaccinoOS Desktop is a Gentoo-based Linux distribution using **Luet** as a package manager, which is static and uses containers to build packages.

Main Features

- Simple to install and use — get a full desktop up and running in minutes
- Applications ship as self-contained layers — install and uninstall without dependency headaches
- Rolling release with LTS and mainline kernel options — always up to date
- Wide desktop environment support: KDE Plasma, GNOME, XFCE, MATE, Hyprland, and more
- Gaming-ready out of the box: Steam, gamemode, VKBasalt, and Proton support
- Luet package manager with both a command-line interface and a graphical tool (Vajo)
- Flatpak, Snap, AppImage, and Wine supported for maximum software compatibility

2. Mocaccino Desktop

https://www.mocaccino.org/docs/desktop/

Mocaccino Desktop is a Gentoo-based distribution (derived from Sabayon) oriented towards Desktop systems. It contains installable "apps" referred to internally as "layers" to install common suite of packages needed to bootstrap a pure and simple OS.

A user should be able to install KDE Plasma by running `luet install layers/plasma` and nothing else. All micropackages are abstracted away and included in a layer installed as a single package — think Android apps: Install and uninstall should be that simple.

Why MOS Desktop is Different?

For the user, MOS Desktop is a pure and simple OS. Applications bundle all required dependencies in order to run, or share common layers used between them (e.g. MATE, GNOME, GTK-based apps). This allows OTA-alike updates without struggling with package dependencies.

From a developer standpoint, MOS takes a unique approach on package building, allowing developers to iterate locally changes to packages easily, using Docker to build packages locally or Kubernetes in a cluster.

2.1 Download and Install

https://www.mocaccino.org/docs/desktop/install/

Download

https://www.mocaccino.org/docs/desktop/install/download/

Official releases can be downloaded from the **Github repository**:
https://github.com/mocaccinoOS/mocaccino/releases

Login

The live images should log you into the graphical environment automatically. If you require a password for the default user:

- Default username and password is **mocaccino**

Installation

https://www.mocaccino.org/docs/desktop/install/installation/

This guide walks you through installing MocaccinoOS to your hard drive using the **Calamares graphical installer**. It applies to all editions including KDE Plasma, GNOME, XFCE, and MATE.

Step 1: Boot into the Live Environment
After downloading the MocaccinoOS ISO and writing it to a USB stick, boot your computer from the USB. You'll be greeted with the live session desktop. From here you can explore the live environment or go straight to installation.

Step 2: Launch the Calamares Installer
Look for the installer icon labeled "Install MocaccinoOS" on the desktop or in the application menu. Click to start the installation process.

Step 3: Choose Your Language
Select your preferred language for the installation process. This will also set the system language after installation. Click Next to continue.

Step 4: Select Your Location and Time Zone
Choose your region and time zone. This sets your system clock and affects time/date display. Click Next to proceed.

Step 5: Keyboard Layout
Choose the keyboard layout that matches your hardware or preferences. You can test your keyboard in the input field provided. Click Next when ready.

Step 6: Partition Your Disk
Choose how to partition your hard drive. Options: Erase disk (wipes entire disk), Replace a partition, or Manual partitioning. Make sure to back up important data before continuing. Click Next once you've made your choice.

Step 7: Create a User Account
Enter your full name, desired username, and password. Optionally choose to log in automatically or use the same password for the administrator account. Click Next to continue.

Step 8: Summary
Review the installation summary. Double-check partition choices and user information. If correct, click Install and confirm when prompted.

Step 9: Installation Progress
Calamares will now install MocaccinoOS to your disk. This can take a few minutes.

Step 10: Installation Complete
Once finished, you'll see a confirmation screen. You can choose to restart immediately or stay in the live session. After rebooting, log into your new MocaccinoOS system using the username and password you created. Remove the installation media when prompted and reboot.

2.2 Package Management (Luet)

https://www.mocaccino.org/docs/desktop/luet/

MocaccinoOS uses **Luet** as a package manager. Learn how to work with Luet in the command shell or by using Vajo, the GUI/TUI tool.

Using the Command Shell

https://www.mocaccino.org/docs/desktop/luet/luet_cli_commands/

Common luet commands for package management:

`luet install <package>` — Install a package
`luet uninstall <package>` — Remove a package
`luet upgrade` — Upgrade all installed packages
`luet search <query>` — Search for packages in enabled repositories
`luet search --installed <query>` — Search among installed packages
`luet repo list` — List installed repositories
`luet repo update` — Update repository metadata

Vajo — GUI/TUI for Luet

https://www.mocaccino.org/docs/desktop/luet/vajo/

Vajo is a GUI and TUI tool built to make managing the system more user-friendly. It provides a graphical and terminal-based interface for Luet operations, allowing users to browse, install, and manage packages without using the command line directly.

Repositories

By default MocaccinoOS ships with the **mocaccino-desktop-stable** repository installed. This repository has all the core layers and desktop environments.

Community Repository

Any additional software compiled and packaged by the MocaccinoOS project is available in the community repository. For example, neovim, alacritty, or gimp require installing the community repository first:

```
$ sudo luet install repository/mocaccino-community-stable
```

Identify Installed Repositories

```
$ sudo luet repo list
```

Switch Between Stable and Development Repositories

■ **Warning:** Do not mix development and stable repositories! Be sure to only have installed stable or the development repositories.

To remove development repositories:

```
$ luet uninstall -y repository/mocaccino-extra repository/mocaccino-desktop
repository/mocaccino-os-commons
```

To enable stable repositories:

```
$ luet install -y --nodeps repository/mocaccino-extra-stable
repository/mocaccino-desktop-stable repository/mocaccino-os-commons-stable
```

2.3 Software Installation

https://www.mocaccino.org/docs/desktop/software/

Different ways to install and configure software on MocaccinoOS Desktop.

Docker

https://www.mocaccino.org/docs/desktop/software/docker/

Install Docker on MocaccinoOS Desktop:

```
luet install container/docker systemd-service/dockerd systemctl enable docker; systemctl start
docker
```

Apache Webserver with Docker

https://www.mocaccino.org/docs/desktop/software/docker/apache_with_docker/

Install Apache webserver using Docker on MocaccinoOS.

Flatpak

https://www.mocaccino.org/docs/desktop/software/flatpak/

MocaccinoOS Desktop supports Flatpak, and its usage is encouraged for packages like Spotify, Telegram, and others not found in the default desktop repository or community repository.

To install Flatpak, run as root:

```
luet install container/flatpak
```

Flatpak runs applications in a sandbox that prevents them from accessing host content. You can add paths that are accessible to a specific Flatpak app:

```
flatpak override --filesystem=/mnt org.app.Id
```

Flatseal is a GUI permissions manager which offers simple point-and-click permissions operations for Flatpak apps.

Snap

https://www.mocaccino.org/docs/desktop/software/snap/

MocaccinoOS supports Snap package installation for additional software.

AppImage

https://www.mocaccino.org/docs/desktop/software/appimage/

AppImages are self-contained applications that run on any Linux distribution. Download an AppImage, make it executable, and run it directly. No installation required.

```
chmod +x application.AppImage ./application.AppImage
```

Whalebrew

https://www.mocaccino.org/docs/desktop/software/whalebrew/

Whalebrew allows you to install Docker images as if they were native applications.

Wine

https://www.mocaccino.org/docs/desktop/software/wine/

Wine allows running Windows applications on Linux. It avoids the overhead of full virtualization, so resource usage stays low.

Important notes about Wine on MocaccinoOS:

- Wine does not guarantee perfect compatibility: some applications run flawlessly, others only partially or not at all
- For 32-bit Windows applications on a 64-bit system, multi-architecture support (32-bit compatibility libraries) is needed
- Some Windows features (drivers, low-level kernel integrations, copy-protected software) often don't work under Wine
- MocaccinoOS uses `apps/wine-staging` which bundles Wine (Staging) with necessary 32-bit/multilib support

Install Wine:

```
luet install apps/wine-staging
```

Waydroid

https://www.mocaccino.org/docs/desktop/software/waydroid/

Waydroid allows running Android applications on Linux through a container-based approach. It requires a Wayland session to work.

2.4 Desktop Environments

https://www.mocaccino.org/docs/desktop/desktop_environments/

MocaccinoOS supports a wide range of desktop environments, all installable as single layer packages using Luet. After installing a desktop environment, restart your system and select it from the login screen.

KDE Plasma

```
sudo luet install layers/plasma layers/kde-apps-minimal apps/discover
```

The `layers/plasma` package ships with the SDDM login manager. To enable it:

```
sudo systemctl enable sddm --force
```

GNOME

```
sudo luet install layers/gnome apps/gnome-software
```

The `layers/gnome` package ships with the GDM login manager. To enable it:

```
sudo systemctl enable gdm --force
```

MATE

```
sudo luet install layers/mate themes/mate
```

XFCE

```
sudo luet install layers/xfce
```

LXQT

```
sudo luet install layers/lxqt
```

Cinnamon

```
sudo luet install layers/cinnamon
```

Enlightenment

```
sudo luet install layers/enlightenment
```

Trinity

```
sudo luet install layers/trinity
```

Hyprland

Hyprland is available in the community repository (install it first if you haven't). It includes the Kitty terminal emulator:

```
sudo luet install layers/hyprland apps/kitty
```

Fluxbox

```
sudo luet install layers/fluxbox
```

COSMIC

```
sudo luet install layers/cosmic apps/cosmic-store
```

Login Managers

If you prefer **LightDM** as your login manager:

```
sudo luet install apps/lightdm sudo systemctl enable lightdm --force
```

If you prefer **Slim**:

```
sudo luet install apps/slim sudo systemctl enable slim --force
```

2.5 Gaming

https://www.mocaccino.org/docs/desktop/gaming/

Steam

We recommend installing Steam directly from the desktop repository. You can also use the Flatpak version, but that may give problems because it runs in a sandbox.

```
luet install apps/steam
```

Gamemode

Gamemode allows you to temporarily apply some optimizations when you launch a game. This functionality is bundled in the `apps/steam` package. To make Steam start a game with gamemode, right click the game in the Library, select Properties, then in the Launch Options text box enter:

```
gamemoderun %command%
```

VKBasalt

VKBasalt is a post processing layer for Vulkan which enables you to enhance graphics further. It only works with Vulkan, including all Proton games.

Nvidia Drivers

MocaccinoOS does not come with Nvidia-drivers pre-installed. If you require proprietary drivers:

```
luet install drivers/nvidia
```

MangoHud

MangoHud is available in the community repository. If you have enabled this repository:

```
luet install utils/mangohud
```

Proton GE

Proton GE is a fork of Valve's Proton by GloriousEggroll (Thomas Crider) with additional patches and bleeding edge technology that may improve performance and compatibility.

NTFS and Steam Libraries

If you share a partition between Windows and MocaccinoOS that has a Steam library, this can cause issues when trying to launch games. Due to NTFS, creating files with names invalid on Windows causes disk errors. The most common issue is a colon (:) character in filenames that Proton creates. Fix by creating the /compatdata folder on the mounted NTFS disk as a symlink pointing to a folder on a Linux partition.

> ■ **Warning:** If the /compatdata folder already exists on the mounted disk BEFORE the symlink, DELETE IT!

2.6 File Sharing

https://www.mocaccino.org/docs/desktop/file-sharing/

MocaccinoOS Desktop provides several protocols for sharing files over a network: Samba (for Windows/macOS compatibility), NFS (for Linux-to-Linux sharing), and SFTP (secure transfer over SSH).

Samba Configuration

https://www.mocaccino.org/docs/desktop/file-sharing/samba/

Samba allows MocaccinoOS to share files and printers with Windows, macOS, and other Linux systems using the SMB protocol. The core packages are already installed — you only need to configure it.

Configure Global Settings

MocaccinoOS provides a default config file. Copy it before making any changes:

```
sudo cp /etc/samba/smb.conf.default /etc/samba/smb.conf
```

The config file is now at `/etc/samba/smb.conf`. The default `[global]` section works for most simple setups.

Define a Share

To create a shared folder, add a section to the end of `/etc/samba/smb.conf`:

```
[Shared_Docs] comment = Collaborative Document Sharing path = /mnt/storage/shared_data
browsable = yes guest ok = no read only = no create mask = 0775 directory mask = 0775 valid
users = @staff, username
```

Key options: `path` — the directory to share. `guest ok = no` requires authentication. `valid users` restricts access to specific users or groups (prefix groups with @).

Add Samba Users

Samba maintains its own password database separate from Linux. Any user who needs access must be added:

```
sudo smbpasswd -a username
```

Start and Enable Samba

```
sudo systemctl enable smb sudo systemctl start smb
```

NFS Configuration

https://www.mocaccino.org/docs/desktop/file-sharing/nfs/

NFS (Network File System) is ideal for sharing directories between Linux systems on a local network. Install the required utilities:

```
luet install net-fs/nfs-utils
```

To share a directory, add it to `/etc/exports`:

```
/srv/share 192.168.1.0/24(rw,sync,no_subtree_check)
```

Apply the export and enable the service:

```
sudo exportfs -ra sudo systemctl enable nfs-server sudo systemctl start nfs-server
```

On the client, mount the share:

```
sudo mount server-ip:/srv/share /mnt/nfs
```

To mount automatically on boot, add to `/etc/fstab`:

```
server-ip:/srv/share /mnt/nfs nfs defaults 0 0
```

SFTP Configuration

https://www.mocaccino.org/docs/desktop/file-sharing/sftp/

SFTP (Secure File Transfer Protocol) transfers files over an encrypted SSH connection. MocaccinoOS includes OpenSSH, so no additional installation is needed.
Enable and start the SSH service:

```
sudo systemctl enable sshd sudo systemctl start sshd
```

Connect from a remote machine:

```
sftp username@your-ip-address
```

For graphical SFTP access, use a file manager that supports SFTP (e.g. Dolphin, Nautilus, Thunar) or a dedicated client like FileZilla.

2.7 Virtualization

https://www.mocaccino.org/docs/desktop/virtualization/

MocaccinoOS supports virtualization using KVM/QEMU and libvirt.

Virt-manager

https://www.mocaccino.org/docs/desktop/virtualization/virt-manager/

Virt-manager is a graphical interface for managing virtual machines via libvirt. Install it:

```
luet install apps/virt-manager
```

Virt-manager provides a desktop user interface for managing virtual machines through libvirt, allowing you to create, configure, and control virtual machines from a GUI.

2.8 Performance Optimizations

https://www.mocaccino.org/docs/desktop/performance/

MocaccinoOS includes optimized system and network settings by default. These tweaks enhance networking, memory management, and filesystem efficiency. Some settings are enabled by default; additional examples are provided for users who want further customization.

■ **Warning:** Do NOT edit /usr/lib/sysctl.d/70-mocaccino.conf — it will be overwritten on system updates. To make custom changes permanent, create /etc/sysctl.d/99-custom.conf instead.

Default Optimizations (Already Enabled)

The following settings are active out of the box:

```
net.core.default_qdisc=fq # FQ queuing: reduces bufferbloat net.ipv4.tcp_congestion_control=bbr
# BBR: low-latency congestion control vm.swappiness=10 # Reduce swap usage for responsiveness
vm.dirty_ratio=20 # Prevent stutters via disk writeback vm.dirty_background_ratio=5
vm.vfs_cache_pressure=50 # Better I/O performance fs.inotify.max_user_watches=524288 # For
IDEs, Docker, game engines fs.inotify.max_user_instances=1024
```

Customizing TCP Buffer Sizes

■ **Warning:** Only use ONE of the following profiles at a time — do not mix them.

Low Latency (Gaming, Video Calls):

```
net.core.rmem_max=4194304 net.core.wmem_max=4194304 net.ipv4.tcp_rmem=4096 87380 4194304
net.ipv4.tcp_wmem=4096 16384 4194304
```

High Bandwidth (File Transfers, Streaming):

```
net.core.rmem_max=16777216 net.core.wmem_max=16777216 net.ipv4.tcp_rmem=4096 87380 16777216
net.ipv4.tcp_wmem=4096 65536 16777216
```

Copy your chosen settings into `/etc/sysctl.d/99-custom.conf` to make them permanent.

Adjusting vm.max_map_count

The default `vm.max_map_count=65530` works for most users. Some applications (Elasticsearch, game engines, AI tools) may need a higher limit:

```
echo "vm.max_map_count=262144" >> /etc/sysctl.d/99-custom.conf sudo sysctl --system
```

Recommended values: 65530 (default), 262144 (databases/Elasticsearch), 1048576+ (high-performance workloads). Note: the game DayZ requires 1048576.

Applying Changes

After modifying `/etc/sysctl.d/99-custom.conf`, apply immediately without rebooting:

```
sudo sysctl --system
```

To verify a setting:

```
sysctl net.ipv4.tcp_congestion_control
```

To restore defaults, delete your custom file and re-apply:

```
sudo rm /etc/sysctl.d/99-custom.conf sudo sysctl --system
```

2.9 Internationalization

https://www.mocaccino.org/docs/desktop/internationalization/

MocaccinoOS supports internationalization features which are not installed by default but can be added at any time. The following features are available:

- Input methods (Chinese, Japanese, Korean, etc.) — see Fcitx or IBus sections below
- Fonts for additional scripts (CJK, emoji, etc.)
- Locale and language data

Input Method — Fcitx

https://www.mocaccino.org/docs/desktop/internationalization/input_method/fcitx/

Fcitx is an input method framework with support for Chinese, Japanese, Korean, and other languages. Install the Fcitx layer:

```
luet install layers/fcitx
```

The `layers/fcitx` package includes fcitx-based input methods including Chinese table and phonetic engines.

Input Method — IBus

https://www.mocaccino.org/docs/desktop/internationalization/input_method/ibus/

IBus (Intelligent Input Bus) is an alternative input method framework. It supports a wide variety of input methods for different languages.

Spell Checking Dictionaries

https://www.mocaccino.org/docs/desktop/internationalization/spellcheck/

Install spell checking dictionaries for your language to enable spell checking in supported applications. Dictionaries are available through the Hunspell package system.

3. General Section

https://www.mocaccino.org/docs/general/

■ **Warning:** Note: Mocaccino and Luet are under active development. Some features may change between updates. It is always a good idea to back up important data before making significant system changes.

This section covers system configuration topics that apply to all MocaccinoOS installations.

3.1 Network Configuration

https://www.mocaccino.org/docs/general/network-config/

By default your system is configured with DHCP using **Network Manager**, which is installed as part of the desktop layer and handles Wi-Fi and wired connections automatically.

Switching to systemd-networkd

If you want to use **systemd-networkd** instead of Network Manager, follow these steps.
First disable Network Manager and enable systemd-networkd:

```
systemctl stop NetworkManager systemctl disable NetworkManager systemctl enable
systemd-networkd
```

Enable the systemd-resolved service:

```
systemctl enable systemd-resolved systemctl start systemd-resolved
```

List available network connections:

```
networkctl
```

This will display output like:

```
IDX LINK TYPE OPERATIONAL SETUP 1 lo loopback carrier unmanaged 2 enp3s0 ether routable
configured 3 enp2s0 ether off unmanaged
```

Static IP Configuration

Create a network config file for your interface (replace `enp3s0` with your interface name):

```
nano /etc/systemd/network/00-enp3s0.network
```

Add the following for a static address:

```
[Match] Name=enp3s0 [Network] Address=192.168.1.50/24 Gateway=192.168.1.1 DNS=192.168.1.1
```

DHCP Configuration

For DHCP, use the same file path but with this content:

```
[Match] Name=enp3s0 [Network] DHCP=yes
```

3.2 Switching Kernels

https://www.mocaccino.org/docs/general/kernels/

MocaccinoOS images ship with the latest LTS kernel by default. It is possible to switch kernels by running `mos kernel-switcher`.

■ **Warning:** WARNING: Due to a known bug, after switching kernels you may end up without a working initramfs, causing the system to fail to boot. Make sure you have the proper kernel/mocaccino-lts-initramfs or kernel/mocaccino-initramfs installed BEFORE rebooting! See: https://github.com/mocaccinoOS/mocaccino/issues/109

List Available Kernels

```
mos kernel-switcher list
```

Switch to Main Kernel

To switch to the main kernel (updated more frequently):

```
mos kernel-switcher switch kernel/mocaccino-full
```

Switch to LTS Kernel

To switch to the LTS kernel:

```
mos kernel-switcher switch kernel/mocaccino-lts-full
```

3.3 System Config Files

https://www.mocaccino.org/docs/general/system-config/

MocaccinoOS Desktop automatically protects `/etc` from overwriting user-override changes. However, upgrades might require overwriting and upgrading specific default configuration. The `mos` CLI embeds a `config-update` tool compatible with `etc-update` and `dispatch-conf`.

Check for Pending Config Changes

To show a summary of config changes that require merging:

```
$ mos config-update check Files with unmerged config files: 1 - /etc/test/foo (2 unmerged
config files)
```

Merging Configuration

To merge configurations interactively:

```
mos config-update update
```

To check and merge all files one by one:

```
mos config-update update -a
```

Auto-merging

To auto-merge all configurations without interactive prompts:

```
mos config-update update --interactive=false
```

4. Contribution Guidelines

https://www.mocaccino.org/docs/contribution-guidelines/

Contributing to MocaccinoOS. The project is open-source and community-driven. Contributions are welcome in all forms:

- **Bug Reports:** Open an issue at https://github.com/mocaccinoOS/mocaccino/issues
- **Pull Requests:** Fork the relevant repository and submit a PR
- **Documentation:** Help improve the docs by editing pages on GitHub
- **Package Requests:** Request new packages for the community repository
- **Package Contributions:** Submit PRs to the community repository

When contributing to MocaccinoOS Desktop packages, make sure to:

- Read the Luet documentation on package building
- Follow the existing package structure and conventions
- Test changes on a test system before submitting
- Revbump reverse dependencies when making breaking changes

Key repositories:

- Main repository: https://github.com/mocaccinoOS/mocaccino
- Desktop packages: https://github.com/mocaccinoOS/desktop
- Community repository: https://github.com/mocaccinoOS/community-repository
- Website/Docs: https://github.com/MocaccinoOS/website

5. Support / Donate

https://www.mocaccino.org/docs/donate/

MocaccinoOS is currently supported by community donations. While development continues, funding helps sustain long-term work.

MocaccinoOS builds on the incredible work of the Gentoo Linux community. All packages are compiled on the project's own infrastructure powered by Kairos.

If you find MocaccinoOS useful and would like to help keep the project healthy, consider supporting via donations.

Donate page: https://www.mocaccino.org/docs/donate/

Appendix: Quick Reference

Common Luet Commands

`luet install <package>`
Install a package

`luet uninstall <package>`
Uninstall a package

`luet upgrade`
Upgrade all packages

`luet search <query>`
Search packages in repositories

`luet search --installed <query>`
Search installed packages

`luet repo list`
List installed repositories

`luet repo update`
Update repository metadata

`luet cleanup`
Clean up cache and downloaded packages

`luet install repository/<repo-name>`
Add a repository

`luet uninstall repository/<repo-name>`
Remove a repository

Key Repositories

- **mocaccino-desktop-stable** — Core layers and desktop environments (installed by default)
- **mocaccino-community-stable** — Community packages (neovim, alacritty, gimp, etc.)
- **mocaccino-extra-stable** — Extra MocaccinoOS packages
- **luet** — Official Luet repository

Useful Links

- **Documentation**: https://www.mocaccino.org/docs/
- **Issue Tracker**: https://github.com/mocaccinoOS/mocaccino/issues
- **Packages**: https://packages.mocaccino.org
- **Community**: https://www.mocaccino.org/community/
- **GitHub Organization**: https://github.com/mocaccinoOS
- **Luet Documentation**: https://luet-lab.github.io/docs/