

KDE Plasma

KDE Plasma is a modern, highly customizable desktop environment that strikes a rare balance in the Linux ecosystem: it delivers a gorgeous, feature-rich visual experience without hogging your system resources. While it looks premium out of the box, its true power lies in its absolute modularity, allowing you to tweak everything from window rules to panel layouts.

The base KDE Plasma desktop can be installed by either the `plasma` **group** or the `plasma-meta` **meta package**. The group installs all packages in it once, while the meta package pulls in the packages via dependencies and installs new packages when the meta package gets updated to include new packages as dependencies. Which one you choose is personal preference how you'd like to manage your installation.

Install either `plasma` **or** `plasma-meta`:

```
# Plasma group
pacman -S plasma

# Plasma meta package
pacman -S plasma-meta
```

This will install only the desktop without barely any applications. Instead of installing the massive `kde-applications` bundle (which includes over 190 packages, many of which you may never use), Arch Linux splits the ecosystem further into modular, category-specific **groups** and **meta-packages**.

Use the table below to selectively install only the categories that you need:

NOTE: The same semantics as for the base `plasma` group/meta package apply. Append ``-meta`` to any group, if you prefer the meta package variants.

Group Name	Category Focus	Key Included Applications
<code>kde-accessibility</code>	Accessibility	<code>kmouth</code> (speech synthesizer), <code>kmousetool</code>
<code>kde-education</code>	Education & Science	<code>marble</code> (virtual globe), <code>kalzium</code> (periodic table), <code>cantor</code>
<code>kde-games</code>	Gaming & Entertainment	<code>kpat</code> (solitaire), <code>kshisen</code> , <code>granatier</code> (bomberman clone)

Group Name	Category Focus	Key Included Applications
<code>kde-graphics</code>	Graphics & Creativity	<code>gwenview</code> (image viewer), <code>okular</code> (document viewer), <code>kolourpaint</code>
<code>kde-multimedia</code>	Audio & Video	<code>kdenlive</code> (video editor), <code>dragon</code> (media player), <code>jukebox</code> (music player)
<code>kde-network</code>	Networking & Remote	<code>kdeconnect</code> (phone sync), <code>kget</code> (download manager), <code>krdc</code> (remote desktop)
<code>kde-office</code>	Productivity	<code>calligra</code> (office suite), <code>krosspython</code>
<code>kde-pim</code>	Personal Information Management	<code>kmail</code> (email), <code>korganizer</code> (calendar), <code>kaddressbook</code>
<code>kde-sdk</code>	Software Development	<code>katedevel</code> , <code>cervisia</code> , <code>kompare</code> (diff tool)
<code>kde-system</code>	System Administration	<code>kssystemlog</code> (log viewer), <code>kdiskfree</code> (storage space manager)
<code>kde-utilities</code>	Desktop Utilities	<code>kate</code> (advanced text editor), <code>kcalc</code> (calculator), <code>filelight</code> (disk analyzer)

If you want the **complete** suite of KDE applications (over 190 packages), install `kde-applications` (group) or `kde-applications-meta` (meta package).

Setting up the display manager

By default, both the group and meta package install both SDDM and the new Plasma Login Manager. The default since KDE Plasma 6.6 is Plasma Login Manager, as it's specifically designed to better integrate with KDE Plasma.

Enable the display manager to start on boot and present a graphical login interface:

```
systemctl enable plasmalogin
```

The default keymap is set to US English. If your keyboard layout differs, change the default keymap with `localectl`:

NOTE: Executing this command while `chroot` ed into the installation environment will produce an error that the locale could not be found. To set it, finish the installation, reboot into the installed system, press `CTRL + ALT + F3` when Plasma Login Manager shows up (or any F-key between 2 and 7) to switch to a different console, log in via the command line and execute the command as `root`.

```
localectl set-x11-keymap de
```

KDE Wallet

KDE Wallet is the integrated password manager and secret store of KDE Plasma. It stores passwords to websites, WiFi networks, network shares, SSH keys and more.

Unlock Wallet automatically on login

To automatically unlock your wallet on login, the `kwallet-pam` package provides the necessary PAM modules (already part of the `plasma` package group).

There are several caveats to consider:

- Only `blowfish` encryption is supported
- Wallet can only be unlocked if the autologin method saves the password, e.g. when using `pam_autologin`
- Wallet cannot be unlocked when logging in with a fingerprint
- Wallet must be named `kdewallet` (default name)
- Disabling automatic closing of Wallet may be desired to keep it from asking for the password after every use
- When choosing to secure Wallet with a password it must match the user account password

Automatic unlocking can also be achieved by setting no password. Do keep in mind, however, that this could lead to potentially undesired read/write access to your secrets. Enabling *Prompt when an application accesses a wallet* under *Access Control* is highly recommended.

Using Plasma Login Manager (or SDDM) requires no further configuration steps.

Storing SSH key passphrases in Wallet

KDE Wallet can be used to store passphrases for SSH keys and have a KDE prompt appear asking for the password.

To also automatically unlock the SSH keys a SSH agent needs to be set up and running.

The `openssh` package (since version 9.4p1-3) comes with a systemd **user unit** to start the SSH agent on login regardless of a graphical session running:

NOTE: This needs to be run as the user you set up earlier, without `sudo`.

```
systemd --user enable ssh-agent
```

The user unit creates a Unix socket for other applications to communicate with the agent. For these applications to know this socket, the `SSH_AUTH_SOCK` environment variable needs to be set. This can be achieved via user-specific `systemd` environment variables.

On login, `systemd` parses `*.conf` files in `~/.config/environment.d/` and sets environment variables from these. Environment variables are set in a `KEY=VALUE` fashion.

Create a new file `~/.config/environment.d/ssh_agent.conf`:

```
SSH_AUTH_SOCK=$XDG_RUNTIME_DIR/ssh-agent.socket
```

Additionally, to have a KDE dialog box appear in case the passphrase is not stored in your Wallet, point the `SSH_ASKPASS` environment variable to the `ksshaskpass` application (part of the `plasma` package group):

```
SSH_ASKPASS=/usr/bin/ksshaskpass
SSH_ASKPASS_REQUIRE=prefer
```

Chromium-based browsers

To make Chromium-based browsers (Google Chrome, Microsoft Edge, Brave, Opera, etc.) use Wallet as a password store launch it with `--password-store=kwallet6` or `--password-store=detect` (preferred).

To make this launch argument persistent, add it to the "flags" file for the Chromium-based browser you want to use:

Browser	Path
Chromium	<code>~/.config/chromium-flags.conf</code>
Google Chrome	<code>~/.config/chrome-flags.conf</code>
Google Chrome DEV	<code>~/.config/chrome-dev-flags.conf</code>
Vivaldi	<code>~/.config/vivaldi-stable.conf</code>

See also: [Making flags persistent on Arch Wiki](#).

Updated 2026-06-28 14:26:54 UTC by Sebin