

Troubleshooting

Contents

Help! There was an update and my system doesn't boot!

Scrambled early boot messages with OpenRC

LiveISO doesn't boot, showing black screen after GRUB

Updating icu/ncurses/whatever breaks a ton of packages

Invalid or corrupted packages (PGP signature)

Can't play games, run Steam etc!

Warning: this GPT partition label contains no BIOS Boot Partition; embedding won't be possible.

KDE/Plasma won't start, saying something bad about dbus!

I'm having dbus-related problems and I keep seeing messages about /etc/machine-id

The display is locked and I can't unlock it / requires unlocking twice (MATE / XFCE / LXDE / Cinnamon)

Btrfs can't boot from RAID

Service file not present

My file manager / application cannot perform elevated privileges tasks

When I switch theme in Plasma/MATE/XFCE/AnyDEorWM the colours are broken!

What about Qt6 applications?

Installing libelogind breaks dependency systemd-libs / systemd

Installing lib32-elogind breaks dependency 'lib32-systemd'

Steam does not remap non-Xbox controllers to the Xbox layout

mDNS service(s) cannot be found; a service cannot advertise itself via mDNS

KDE Plasma does not automatically refresh the application list

Applications won't open while wireless connection is active

I can't run KMail properly, it nags about Akonadi, which nags about

MySQL/MariaDB/database

MariaDB won't start on a fresh installation

Plasma/Wayland immediately crashes back to SDDM

It is the year of the Linux desktop and you still don't provide a GUI package manager!

Help! There was an update and my system doesn't boot!

Artix is a "rolling release" distribution. Although unlikely, sometimes an updated package will not work. If it is a critical system package, for example glibc, it can cause the system to become unbootable on next boot. Most often, however, failure to boot will be caused by user misconfiguration.

It is a VERY good idea to have **pacman-static** installed in your system. That way, you can repair almost anything without having to resort to external help, i.e. a live ISO. You can get **pacman-static** from the [AUR](#).

First thing to do in such a case is to have a bootable medium, or obtain an Artix Live ISO image and create the bootable medium. You can download fresh ISO images from the [Download](#) page. Boot from the ISO, open the terminal (or just login into a live system if using a base ISO), and login as user *artix* with the password *artix*. You can then inspect what media are recognized by the kernel by using the command

```
lsblk -f
```

For example

```
NAME FSTYPE FSVER LABEL      UUID          FSAVAIL FSUSE%
MOUNTPOINTS
fd0
loop0 squashfs 4.0          0 100% /run/artix/sfs/livefs
loop1 squashfs 4.0          0 100% /run/artix/sfs/rootfs
sda
├─sda1 vfat     FAT32 NO_LABEL  74EC-8A05
├─sda2 ext4     1.0          5fd4dc34-1f6f-49aa-ab2f-80669d374369
sr0  iso9660  ARTIX_202107 2021-07-25-19-06-42-00 0 100% /run/artix/bootmnt
```

Here, `/dev/sda2` is the ext4 partition holding the root filesystem. From there, you can try to mount your root system by using the commands

```
su # If asked, use the password "artix"
mount /dev/sda2 /mnt
```

and modify any needed files by accessing them under `/mnt`, for example

```
vi /mnt/etc/pacman.conf
```

If **pacman** isn't working in the installed system, here you can also download any needed Artix packages by using programs preinstalled in LiveISO, such as `curl` or `scp`. You can also temporarily install any other programs you need to recover, say, `wget` in the live environment with `pacman -Sy wget`. Afterwards, you can chroot to a mounted system by using the command

```
artix-chroot /mnt
```

Any subsequent commands in that terminal will be carried out as though issued on the root filesystem from the hard disk. For example, you can type

```
pacman -U /path/to/package_name.tar.zst
```

to install the manually downloaded file, or just

```
pacman -Syyu
```

to update your system.

Scrambled early boot messages with OpenRC

A bug in *kbd-2.6.0* combined with one in *ncurses* (and some liberties taken by OpenRC regarding *printf*) are to blame. It only occurs when the *consolefont* service is enabled at *boot* runlevel. Quick fix:

```
rc-update del consolefont boot
rc-update add consolefont default
```

LivISO doesn't boot, showing black screen after GRUB

Things to try:

- Activate UEFI mode in BIOS setup.
- Disable "Fast/Secure Boot" in BIOS setup.
- Add `nomodeset` to the kernel command line.
- My computer boots to a black screen, what options do I have to fix it?

Updating icu/ncurses/whatever breaks a ton of packages

Make sure you have the correct repositories activated in `/etc/pacman.conf`. Testing ([*-gremlins]) repositories should be disabled by default. These **must** be above all other repositories.

The official Artix ISOs do not enable Arch repositories; only exception are the community editions, which come with Arch repositories enabled and may contain some packages

from there.

Note: [gremlins] is Artix equivalent of Arch [testing]

```
#[system-gremlins]
#Include = /etc/pacman.d/mirrorlist
#[world-gremlins]
#Include = /etc/pacman.d/mirrorlist
#[galaxy-gremlins]
#Include = /etc/pacman.d/mirrorlist
#[lib32-gremlins]
#Include = /etc/pacman.d/mirrorlist

[system]
Include = /etc/pacman.d/mirrorlist
[world]
Include = /etc/pacman.d/mirrorlist
[galaxy]
Include = /etc/pacman.d/mirrorlist
# If you want to run 32 bit applications on your x86_64 system,
# enable the multilib repositories as required here.
[lib32]
Include = /etc/pacman.d/mirrorlist
```

Some version mismatches during updates may occur when the Artix team builds newer versions; usually, a few hours of patience shall reward users with the correct versions, once the mirrors have picked up the updates. If it takes longer than a day, the best course of action is to report the problem at the official Artix IRC channel or the forum.

The list below might not be updated, always check the [Gitea repository](#).

```
##
## Artix Linux repository mirrorlist
## Generated on 2023-05-28
##

# Artix mirrors
# Use rankmirrors(1) to get a list of the fastest mirrors for your location,
# e.g.: rankmirrors -v -n 5 /etc/pacman.d/mirrorlist
# Then put the resulting list on top of this file.

# Default mirrors
Server = https://mirrors.dotsrc.org/artix-linux/repos/$repo/os/$arch
Server = https://mirror.clarkson.edu/artix-linux/repos/$repo/os/$arch
Server = http://ftp.ntua.gr/pub/linux/artix-linux/$repo/os/$arch
```

Europe

Czech Republic

Server = [https://ftp.sh.cvut.cz/artix-linux/\\$repo/os/\\$arch](https://ftp.sh.cvut.cz/artix-linux/$repo/os/$arch)

Denmark

Server = [https://mirrors.dotsrc.org/artix-linux/repos/\\$repo/os/\\$arch](https://mirrors.dotsrc.org/artix-linux/repos/$repo/os/$arch)

Server = [https://mirror.one.com/artix/\\$repo/os/\\$arch](https://mirror.one.com/artix/$repo/os/$arch)

France

Server = [https://artix.cccp.io/\\$repo/os/\\$arch](https://artix.cccp.io/$repo/os/$arch)

Server = [https://ftp.crifo.org/artix/repos/\\$repo/os/\\$arch](https://ftp.crifo.org/artix/repos/$repo/os/$arch)

Server = [https://mirror.opensrv.org/artixlinux/\\$repo/os/\\$arch](https://mirror.opensrv.org/artixlinux/$repo/os/$arch)

Hungary

Server = [https://quantum-mirror.hu/mirrors/pub/artix-linux/\\$repo/os/\\$arch](https://quantum-mirror.hu/mirrors/pub/artix-linux/$repo/os/$arch)

Germany

Server = [https://mirror.netcologne.de/artix-linux/\\$repo/os/\\$arch](https://mirror.netcologne.de/artix-linux/$repo/os/$arch)

Server = [http://mirrors.redcorelinux.org/artixlinux/\\$repo/os/\\$arch](http://mirrors.redcorelinux.org/artixlinux/$repo/os/$arch)

Server = [https://mirror.pascalpuffke.de/artix-linux/\\$repo/os/\\$arch](https://mirror.pascalpuffke.de/artix-linux/$repo/os/$arch)

Server = [https://ftp.uni-bayreuth.de/linux/artix-linux/\\$repo/os/\\$arch](https://ftp.uni-bayreuth.de/linux/artix-linux/$repo/os/$arch)

Server = [https://ftp.halifax.rwth-aachen.de/artixlinux/\\$repo/os/\\$arch](https://ftp.halifax.rwth-aachen.de/artixlinux/$repo/os/$arch)

Server = [https://artix.unixpeople.org/repos/\\$repo/os/\\$arch](https://artix.unixpeople.org/repos/$repo/os/$arch)

Server = [https://mirror1.artixlinux.org/repos/\\$repo/os/\\$arch](https://mirror1.artixlinux.org/repos/$repo/os/$arch)

Server = [https://eu-mirror.artixlinux.org/repos/\\$repo/os/\\$arch](https://eu-mirror.artixlinux.org/repos/$repo/os/$arch)

Greece

Server = [https://ftp.cc.uoc.gr/mirrors/linux/artixlinux/\\$repo/os/\\$arch](https://ftp.cc.uoc.gr/mirrors/linux/artixlinux/$repo/os/$arch)

Server = [http://ftp.ntua.gr/pub/linux/artix-linux/\\$repo/os/\\$arch](http://ftp.ntua.gr/pub/linux/artix-linux/$repo/os/$arch)

Monaco

Server = [https://mirrors.qontinuum.space/artixlinux/\\$repo/os/\\$arch](https://mirrors.qontinuum.space/artixlinux/$repo/os/$arch)

Poland

Server = [https://artix.sakamoto.pl/\\$repo/os/\\$arch](https://artix.sakamoto.pl/$repo/os/$arch)

Sweden

Server = [https://ftp.ludd.ltu.se/mirrors/artix/\\$repo/os/\\$arch](https://ftp.ludd.ltu.se/mirrors/artix/$repo/os/$arch)

Server = [https://mirror.linux.pizza/artix-linux/\\$repo/os/\\$arch](https://mirror.linux.pizza/artix-linux/$repo/os/$arch)

Switzerland

Server = [https://artix.kurdy.org/\\$repo/os/\\$arch](https://artix.kurdy.org/$repo/os/$arch)

United Kingdom

Server = [http://artist.artixlinux.org/repos/\\$repo/os/\\$arch](http://artist.artixlinux.org/repos/$repo/os/$arch)

Server = [https://mirror.vinehost.net/artix-linux/\\$repo/os/\\$arch](https://mirror.vinehost.net/artix-linux/$repo/os/$arch)

North America

United States

Server = [https://artix.wheaton.edu/repos/\\$repo/os/\\$arch](https://artix.wheaton.edu/repos/$repo/os/$arch)

Server = [https://mirror.clarkson.edu/artix-linux/repos/\\$repo/os/\\$arch](https://mirror.clarkson.edu/artix-linux/repos/$repo/os/$arch)

Server = [https://mirrors.rit.edu/artixlinux/\\$repo/os/\\$arch](https://mirrors.rit.edu/artixlinux/$repo/os/$arch)

Server = [https://mirrors.ocf.berkeley.edu/artix-linux/\\$repo/os/\\$arch](https://mirrors.ocf.berkeley.edu/artix-linux/$repo/os/$arch)

Server = [http://www.nylxs.com/mirror/repos/\\$repo/os/\\$arch](http://www.nylxs.com/mirror/repos/$repo/os/$arch)

Server = [https://mirrors.nettek.us/artix-linux/\\$repo/os/\\$arch](https://mirrors.nettek.us/artix-linux/$repo/os/$arch)

Server = [https://us-mirror.artixlinux.org/\\$repo/os/\\$arch](https://us-mirror.artixlinux.org/$repo/os/$arch)

Canada

Server = [https://mirror.csclub.uwaterloo.ca/artixlinux/\\$repo/os/\\$arch](https://mirror.csclub.uwaterloo.ca/artixlinux/$repo/os/$arch)

South America

Brazil

Server = [https://gnlug.org/pub/artix-linux/\\$repo/os/\\$arch](https://gnlug.org/pub/artix-linux/$repo/os/$arch)

Chile

Server = [https://mirror1.cl.netactuate.com/artix/repos/\\$repo/os/\\$arch](https://mirror1.cl.netactuate.com/artix/repos/$repo/os/$arch)

Asia

China

Server = [https://mirrors.tuna.tsinghua.edu.cn/artixlinux/\\$repo/os/\\$arch](https://mirrors.tuna.tsinghua.edu.cn/artixlinux/$repo/os/$arch)

Server = [https://mirrors.aliyun.com/artixlinux/\\$repo/os/\\$arch](https://mirrors.aliyun.com/artixlinux/$repo/os/$arch)

Server = [https://mirror.nju.edu.cn/artixlinux/\\$repo/os/\\$arch](https://mirror.nju.edu.cn/artixlinux/$repo/os/$arch)

India

Server = [https://mirror.albony.xyz/artix/\\$repo/os/\\$arch](https://mirror.albony.xyz/artix/$repo/os/$arch)

Korea, Republic of

Server = [https://mirror.funami.tech/artix/\\$repo/os/\\$arch](https://mirror.funami.tech/artix/$repo/os/$arch)

Singapore

Server = [https://mirror.freedif.org/Artix/\\$repo/os/\\$arch](https://mirror.freedif.org/Artix/$repo/os/$arch)

Taiwan

Server = [https://mirrors.cloud.tencent.com/artixlinux/\\$repo/os/\\$arch](https://mirrors.cloud.tencent.com/artixlinux/$repo/os/$arch)

Vietnam

Server = [https://mirrors.42tm.tech/artix-linux/\\$repo/os/\\$arch](https://mirrors.42tm.tech/artix-linux/$repo/os/$arch)

Oceania

Australia

Server = [https://mirror.aarnet.edu.au/pub/artix/\\$repo/os/\\$arch](https://mirror.aarnet.edu.au/pub/artix/$repo/os/$arch)

Invalid or corrupted packages (PGP signature)

If pacman warns you about invalid or corrupted packages, it may be due to obsolete PGP keys or Arch-signed packages in the repos. Make sure the Artix repos are above the Arch ones and:

1. Reinstall keyrings including the latest keys:

```
pacman -Sy archlinux-keyring artix-keyring
```

If you can't install the artix-keyring because of signature errors, perform step 2 and

repeat 1, otherwise proceed to step 3.

2. Remove old and possibly expired, revoked or invalid keys by issuing this command:

```
rm -r /etc/pacman.d/gnupg
```

3. Initialize the pacman keyring:

```
pacman-key --init
```

4. Load the signature keys:

```
pacman-key --populate archlinux artix
```

5. Clear out the software packages downloaded during the aborted installation:

```
pacman -Scc  
pacman -Syyu
```

6. In a pinch, install the package with the `-U` pacman switch:

```
pacman -U /var/cache/pacman/pkg/package-1.3.9-1.x86_64.pkg.tar.xz
```

It is also possible that one of pacman mirrors, perhaps even the first one can't be reached (returning a 404). In this case follow these steps:

1. Change the order of mirror list.

2. Delete sync database:

```
sudo rm -fr /var/lib/pacman/sync
```

3. Update database and update the system:

```
pacman -Syyu
```

Can't play games, run Steam etc!

You must enable `[lib32]` from Artix and `[multilib]` from Arch in `/etc/pacman.conf` and install relevant packages for 32bit executables. Make sure `[lib32]` is above `[multilib]` (as a matter of fact **all** Artix repos must be above Arch's).

```
[lib32]  
Include = /etc/pacman.d/mirrorlist  
[multilib]
```

Warning: this GPT partition label contains no BIOS Boot Partition; embedding won't be possible.

Install parted and execute:

```
parted -s /dev/sdx set 1 bios_grub on
```

KDE/Plasma won't start, saying something bad about dbus!

See next section.

I'm having dbus-related problems and I keep seeing messages about `/etc/machine-id`

This file originated from dbus development and was adopted by systemd as a universally unique machine identifier. Ergo, it is a useless (for the end user) tag but apparently of some use to dbus because *"The important properties of the machine UUID are that 1) it remains unchanged until the next reboot and 2) it is different for any two running instances of the OS kernel. That is, if two processes see the same UUID, they should also see the same shared memory, UNIX domain sockets, local X displays, localhost.localdomain resolution, process IDs, and so forth."* Also, *"The simple configuration file format of /etc/machine-id originates in the /var/lib/dbus/machine-id file introduced by D-Bus. In fact, this latter file might be a symlink to /etc/machine-id."* and *This ID uniquely identifies the host. It should be considered "confidential", and must not be exposed in untrusted environments, in particular on the network.* Privacy-conscious people *might* want to regenerate their machine-ids frequently; a new ID can be generated with

```
dbus-uuidgen >| /etc/machine-id
```

or

```
dbus-uuidgen >| /var/lib/dbus/machine-id
```

You may want to symlink one to another (recommended) or choose to have two different machine IDs (advanced).

Note: Some non-privacy respecting programs might depend on machine-id being accessible (ie. whitelisted when executed from firejail) and not changing on a regular basis. Viber is one such program.

The display is locked and I can't unlock it / requires unlocking twice (MATE / XFCE / LXDE / Cinnamon)

It may be the fault of **light-locker** (the screenlocker of **lightdm**). Uninstall the former (**light-locker**) and let the screenlocker of your chosen DE handle screen locking.

Btrfs can't boot from RAID

In some setups, Btrfs can't boot if the root filesystem is on RAID subvolumes, despite adding btrfs to `/etc/mkinitcpio.conf` hooks array (and re-creating a new initramfs) or using `rootflags=device=/dev/sda2,device=/dev/sdb2` as a workaround in the kernel command line. The solution is to instruct the btrfs hook create the node manually; edit `/usr/lib/initcpio/hooks/btrfs` as follows:

```
run_hook() {
    mknod /dev/btrfs-control c 10 234
    btrfs device scan
}
```

and re-create the initramfs (**mkinitcpio -P**).

Service file not present

The init scripts of daemons (also known as 'services') are packaged separately. With very few exceptions, the naming convention is consistent across inits, e.g. for dnsmasq it's dnsmasq-openrc, dnsmasq-runit, dnsmasq-s6 and dnsmasq-dinit. Installing the service package will also pull and install the daemon package too. Example for OpenRC:

```
pacman -S dnsmasq-openrc
rc-update add dnsmasq default
rc-service dnsmasq start
```

My file manager / application cannot perform elevated privileges tasks

Make sure your desktop session is started with *dbus-launch* in `~/.xinitrc`

When I switch theme in Plasma/MATE/XFCE/AnyDEorWM the colours are broken!

All Artix ISOs, except base, come slightly preconfigured with a dark theme. To force uniformity across Qt and Gtk, some tricks had to be performed, namely:

- Creating a theme that has support for all major toolkits, i.e. Qt, Gtk2 and Gtk3. In our case, the dark variant of Vertex was modified into Artix-dark.
- Choosing a common style, icon and colour theme.
- Instructing (or forcing) the active toolkit to use a theme engine that translates one toolkit into another. The chosen solution was to make Qt follow Gtk and this was achieved through qt5-styleplugins and the `QT_*` environment variables defined in `/etc/environment`.

The overall look of the minimal desktop ISOs is controlled by the following packages, listed alphabetically:

```
artix-backgrounds
artix-dark-theme
artix-desktop-presets
artix-grub-theme
artix-gtk-presets
artix-icons
artix-qt-presets
qt6gtk2
```

The `artix-desktop-presets` package contains the common settings which instruct a desktop environment (DE) to use the dark theme. Additional, more toolkit-specific settings are contained in the `artix-{gtk,qt}-presets` packages. However, not all applications respect the chosen theme and arbitrarily set the background and foreground colours, usually white and black respectively. For those applications that are part of the ISO images, the `artix-{gtk,qt}-presets` packages also contain configuration files that control their appearance. Finally, `artix-community-presets` contain settings for a much larger collection of applications, namely the ones found in the community editions ISO images.

Notice: very few themes have complete support for Qt *and* Gtk2 *and* Gtk3, namely Breeze, Adwaita, Vertex and Artix-dark. Ergo, using another theme will most probably result in inconsistent look across different toolkit applications.

Most of these settings are installed in `/etc/skel`, which means that they are copied over to the home directory of every created user *upon creation*. Installing the packages doesn't mean the settings are applied automatically; the configuration files must be copied from `/etc/skel` to the home directory of each user (be careful not to overwrite existing files without a backup). It also means that removing the package doesn't delete the files from the home directories: **they have to be removed manually**, if so desired.

Using another theme is generally very easy: just select it in the appearance settings page of your preferred desktop environment. The selected theme will (usually) also change the style and colour theme to its own. If not, select it yourself. For Plasma, the settings are found under *Appearance* -> *Global Theme*, *Plasma Style*, *Application Style* and *Colors*.

Plasma and LXQt can also control the appearance of non-native toolkit applications.

- For Plasma: change the GTK2 and GTK3 themes in the *Application Style* tab to your desired theme, e.g. Breeze.
- For LXQt: these settings are found in the *Widget* tab of the *Appearance* module and in the *Openbox Settings* module of the *Configuration Center*. LXQt users will additionally need to disable the `QT_QPA_PLATFORMTHEME` variable wherever it is set, usually `/etc/xdg/lxqt/session.conf` and perhaps `~/.config/lxqt/session.conf`.

Our Gtk-based DEs use the `QT_QPA_PLATFORMTHEME` and `QT_STYLE_OVERRIDE` environment variables to control the appearance of Qt applications. Using an all-toolkit theme like Breeze or Artix-dark is recommended and no additional actions are needed to achieve visually appealing results. Otherwise, those variables may need to be disabled in `/etc/environment`. For some Qt applications that follow `~/.config/kdeglobals`, you may need to remove the lines containing 'artix-dark' (or delete it).

Last, but not least, some applications may partially or completely ignore the user's `~/.gtkrc-2.0` over the global `/usr/share/gtk-2.0/gtkrc`. The tell-tale of such cases is mixed-colour window widgets (i.e. from both *artix-dark* and the selected theme). Should you encounter this, just rename `/usr/share/gtk-2.0/gtkrc` to `/usr/share/gtk-2.0/gtkrc.bak` and retry. A logout may be necessary.

More information, totally applicable in Artix, can be found on the [related Arch Wiki page](#).

What about Qt6 applications?

This is harder. Until cross-toolkit themes with Qt6 support mature, you can install `qt6gtk2`

either from [galaxy] or from the AUR and create aliases for your Qt6 applications in your account's `bashrc` or the system-wide one:

```
alias strawberry='QT_QPA_PLATFORMTHEME=qt5gtk2 strawberry'
```

Installing libelogind breaks dependency systemd-libs / systemd

This happens because a package on your system — usually from Arch Linux or from the AUR — lists `systemd` and/or `systemd-libs` as dependencies.

Since version 246.10-4, Artix's `elogind/libelogind` package no longer provides `systemd`. Therefore, to install/update these packages, you'll need to install `artix-archlinux-support`, which provides a dummy `systemd/systemd-libs`.

```
pacman -S artix-archlinux-support
```

Installing lib32-elogind breaks dependency 'lib32-systemd'

This happens because a package on your system — usually from Arch Linux or from the AUR — lists `lib32-systemd` as a dependency.

Since version 246.10-2, Artix's `lib32-elogind` package no longer provides `libsystemd`. Therefore, to install/update these packages, you'll need to install `lib32-artix-archlinux-support`, which provides a dummy `lib32-systemd`.

```
pacman -S lib32-artix-archlinux-support
```

Steam does not remap non-Xbox controllers to the Xbox layout

Create the file `/etc/modules-load.d/uinput` with the following text inside:

```
uinput
```

mDNS service(s) cannot be found; a service

cannot advertise itself via mDNS

Install the Avahi package:

```
$ sudo pacman -S avahi avahi-{init}
```

And add `avahi-daemon` to the list of startup services.

KDE Plasma does not automatically refresh the application list

Trigger a manual refresh:

```
$ kbuildsycoca5
```

Applications won't open while wireless connection is active

Edit `/etc/NetworkManager/NetworkManager.conf` by adding/changing:

```
[main]
plugins=keyfile
hostname-mode=none
```

I can't run KMail properly, it nags about Akonadi, which nags about MySQL/MariaDB/database

See next section

MariaDB won't start on a fresh installation

It's by design. You have to initialize the system tables yourself and read the post-installation notes.

```
sudo mariadb-install-db --user=mysql --basedir=/usr --datadir=/var/lib/mysql
```

Or, if you're on OpenRC,

```
sudo rc-service mariadb-supervise setup
```

Now you can enable and start the service.

Plasma/Wayland immediately crashes back to SDDM

This is wildly reported around the internet and the cause isn't always the same (Wayland being finicky doesn't help either). A workaround is to login in Plasma/X11, open **System Settings** and go to **Color & Themes / Application Style** where you should select **Breeze**, **Fusion**, **Oxygen** or even **MS Windows 9x**. Then logout and select Plasma/Wayland in SDDM.

Another workaround is to set this in `/etc/environment` :

```
QT_STYLE_OVERRIDE=Breeze
```

It is the year of the Linux desktop and you still don't provide a GUI package manager!

But of course we do. The community editions, which you should be using if you don't like reading documentation, have one. Both *pamac* (Gtk) and *octopi* (Qt) are available from the [\[galaxy\] repository](#).

[Edit](#) - [History](#) - [Print](#) - [Recent Changes](#) - [Search](#)

Page last modified on February 18, 2025, at 08:26 PM